

---

# Computational Geometry

## Chapter 8: Tours and Polygons

---

Sándor P. Fekete

Algorithms Division  
Department of Computer Science  
TU Braunschweig



- **Ch 1: Introduction**
- **Ch 2: Convex hulls**
- **Ch 3: Closest pairs**
- **Ch 4: Voronoi diagrams**
- **Ch 5: Polygon triangulation**
- **Ch 6: Point triangulation**
- **Ch 7: Location problems**
- **Ch 8: Tours and polygons**



- 1. Introduction**
- 2. Longest Tours**
- 3. Stars and Matchings**
- 4. Nonsimple Polygons**
- 5. Optimal Area**
- 6. Turn Cost**

**Superhero!**



**Technische  
Universität  
Braunschweig**



Mehmet Ozkanoglu





THE  
TRAVELING  
SALESMAN

**1. Introduction**

**2. Longest Tours**

**3. Stars and Matchings**

**4. Nonsimple Polygons**

**5. Optimal Area**

**6. Turn Cost**



# Several Centuries before Euclid

# Several Centuries before Euclid



# Several Centuries before Euclid



# Several Centuries before Euclid



# Several Centuries before Euclid



# Several Centuries before Euclid



# Several Centuries before Euclid



# Several Centuries before Euclid





# Several Centuries before Euclid



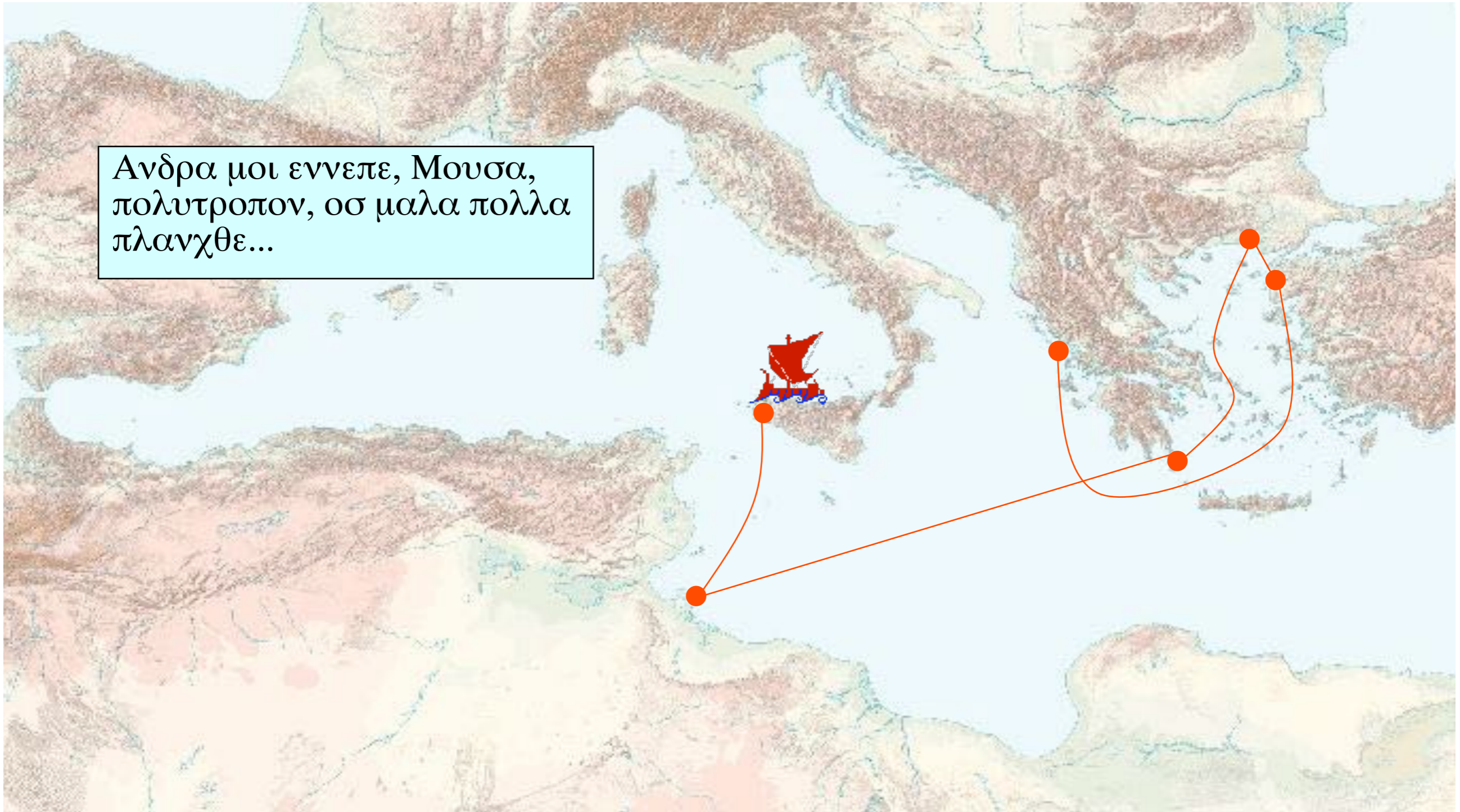
# Several Centuries before Euclid

Ανδρα μοι εννεπε, Μουσα,  
πολυτροπον, οσ μαλα πολλα  
πλανηθε...



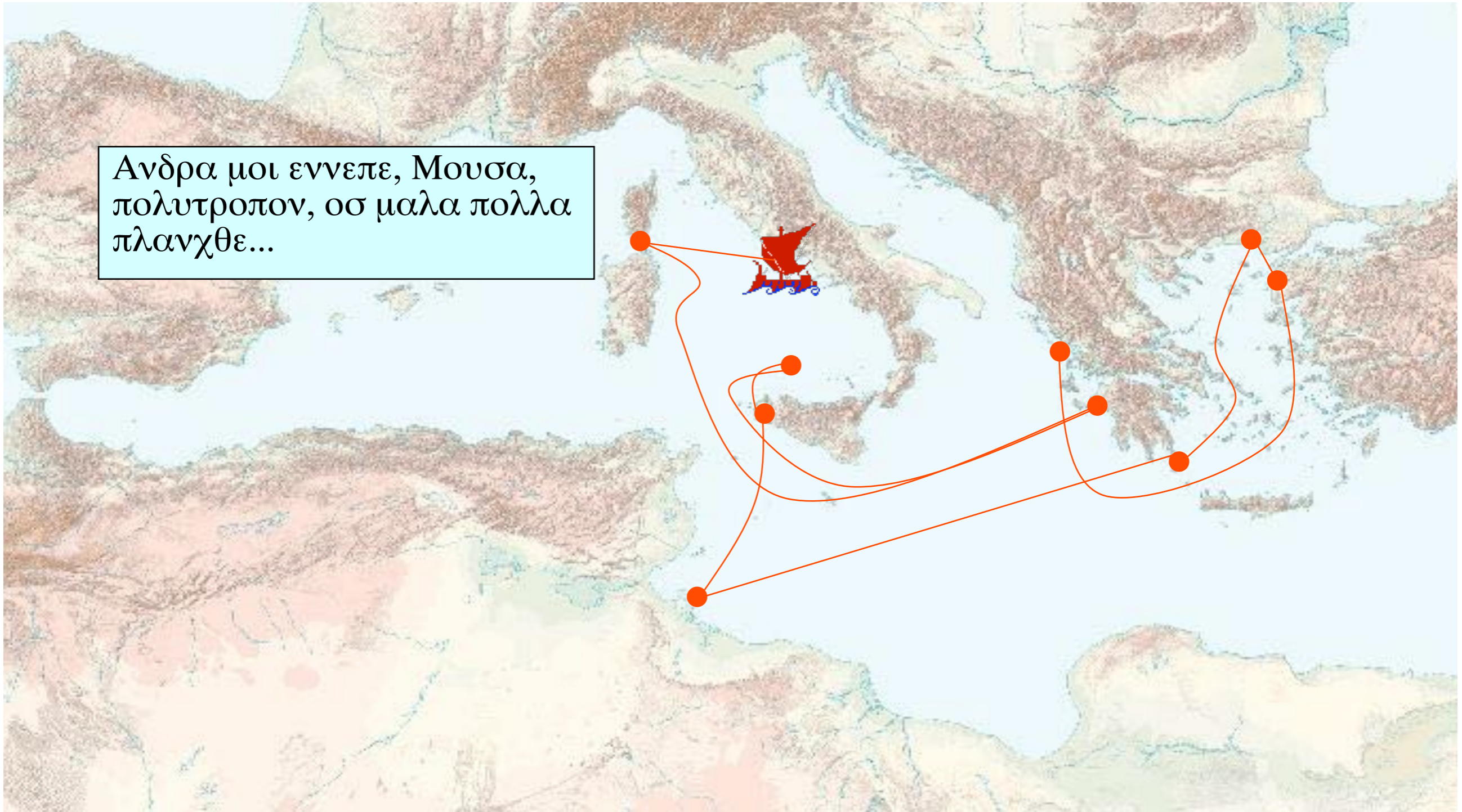
# Several Centuries before Euclid

Ανδρα μοι εννεπε, Μουσα,  
πολυτροπον, οσ μαλα πολλα  
πλανηθε...



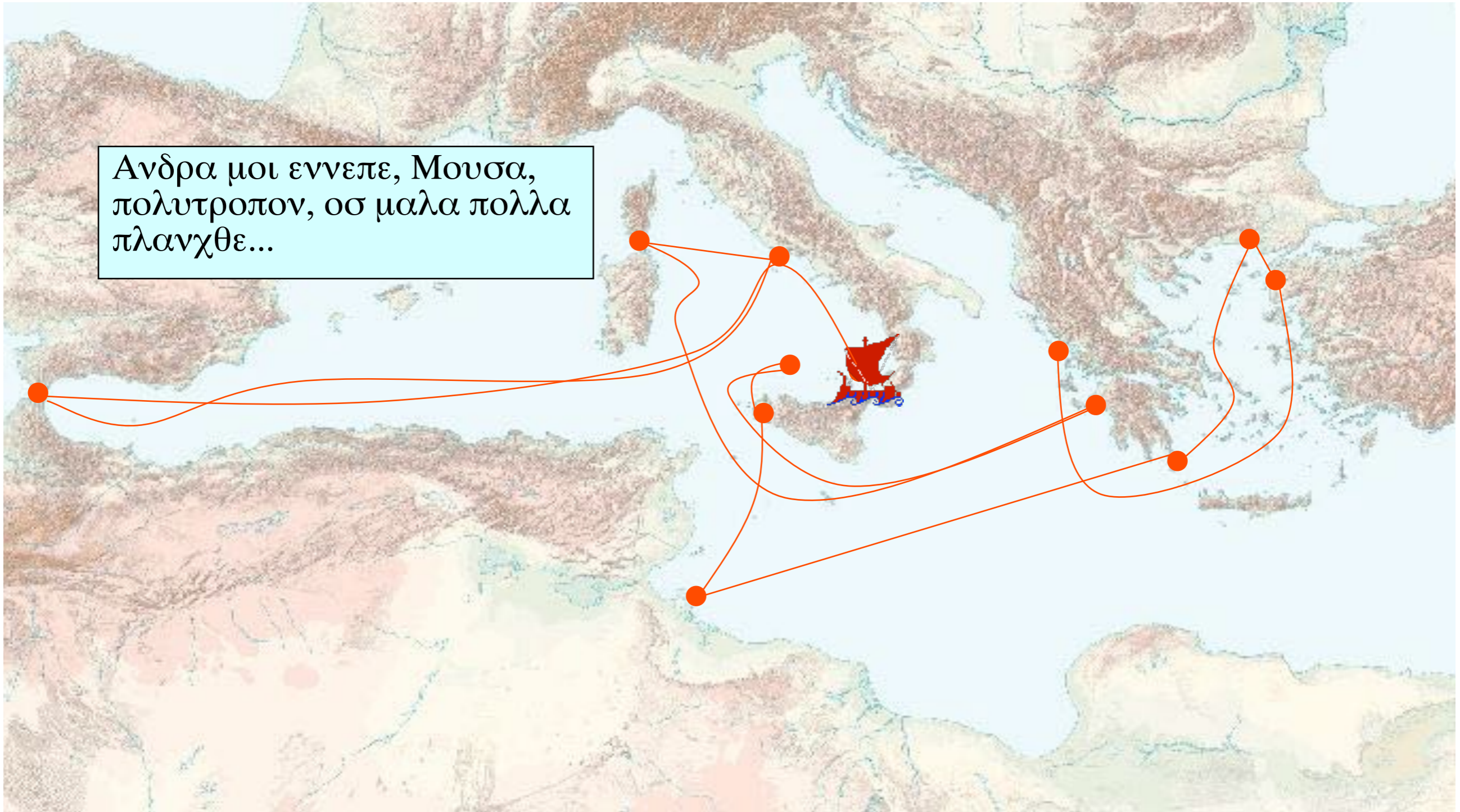
# Several Centuries before Euclid

Ανδρα μοι εννεπε, Μουσα,  
πολυτροπον, οσ μαλα πολλα  
πλανηθε...



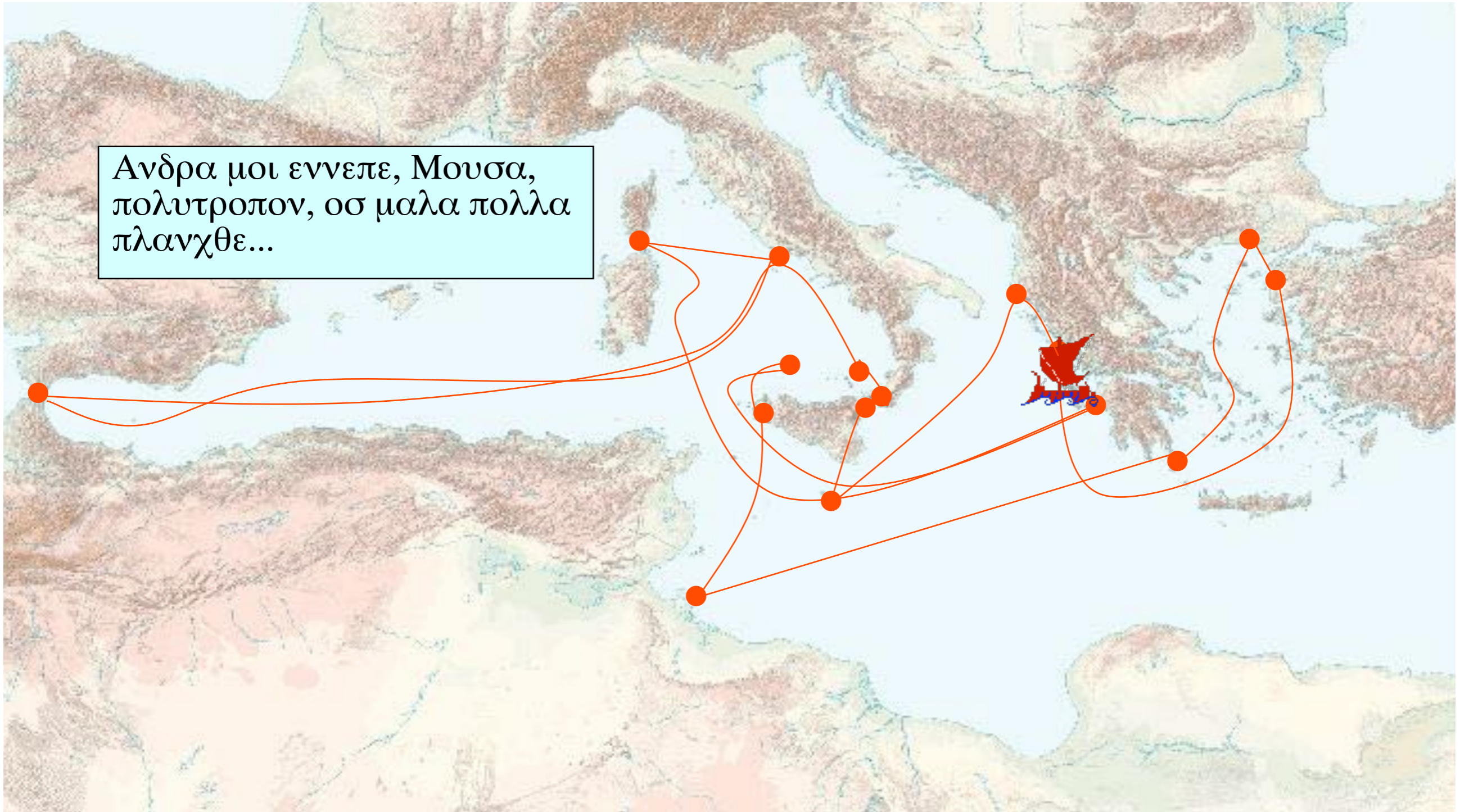
# Several Centuries before Euclid

Ανδρα μοι εννεπε, Μουσα,  
πολυτροπον, οσ μαλα πολλα  
πλανηθε...



# Several Centuries before Euclid

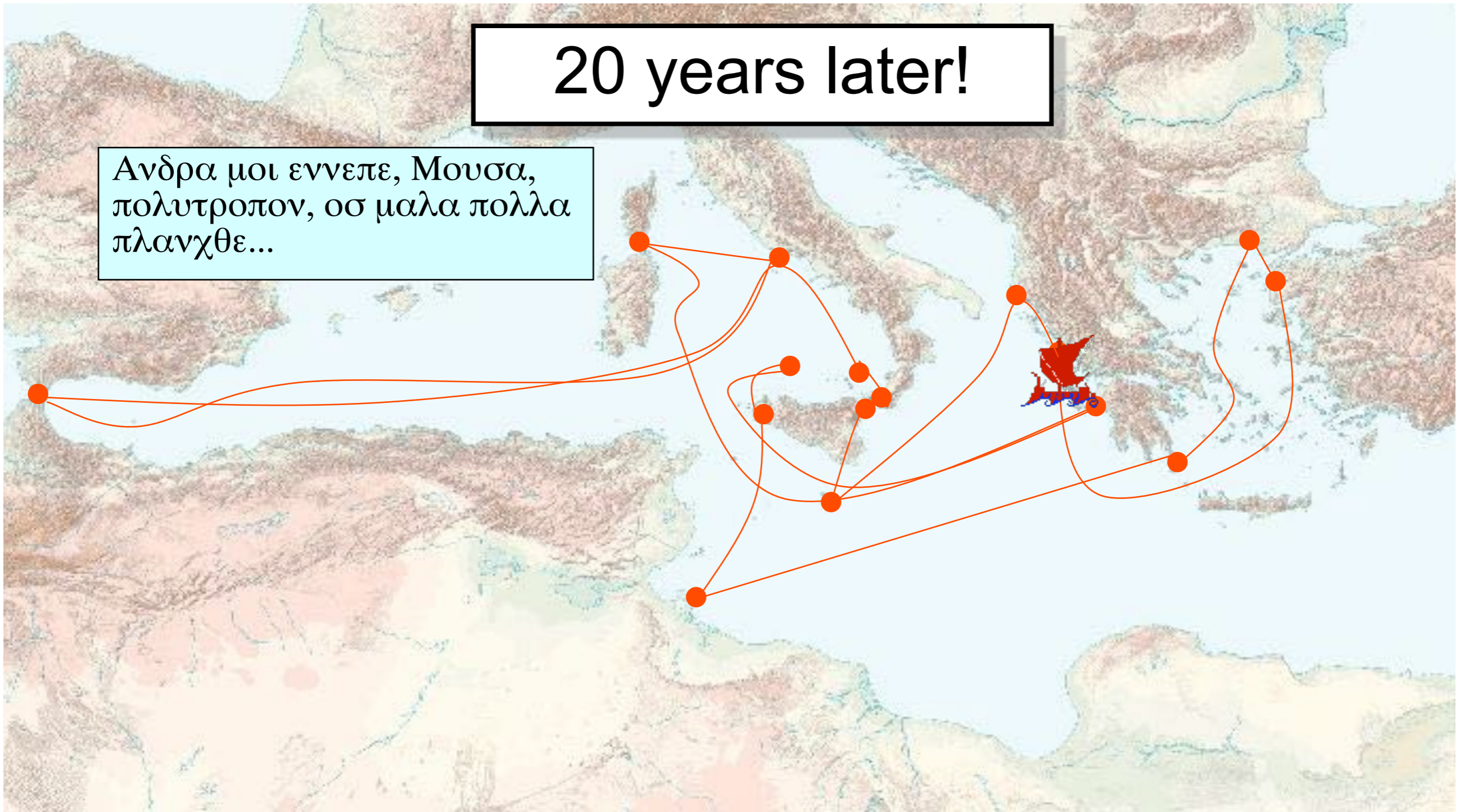
Ανδρα μοι εννεπε, Μουσα,  
πολυτροπον, οσ μαλα πολλα  
πλανχθε...



# Several Centuries before Euclid

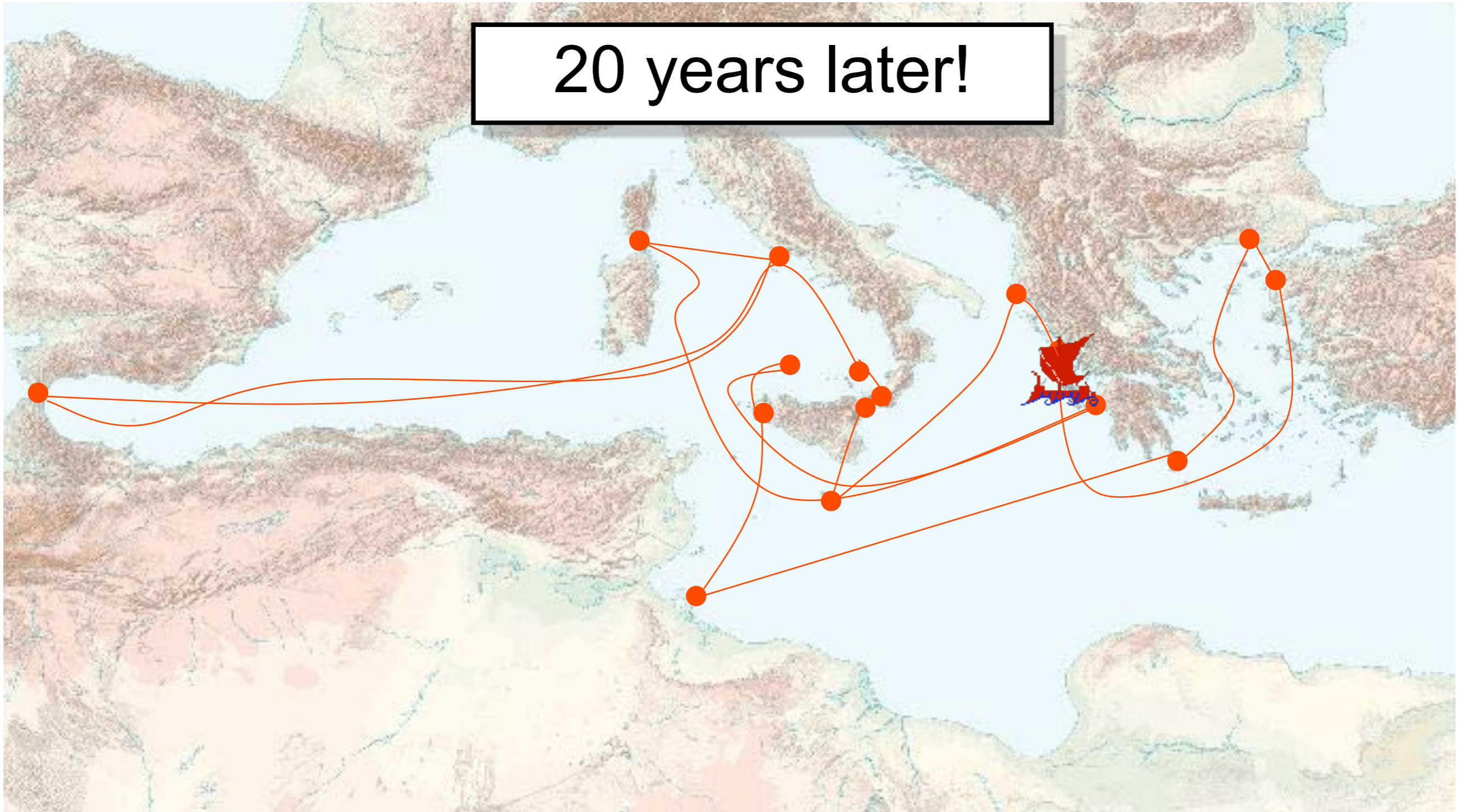
20 years later!

Ανδρα μοι εννεπε, Μουσα,  
πολυτροπον, οσ μαλα πολλα  
πλανχθε...



# Several Centuries before Euclid

20 years later!





# Several Centuries before Euclid

20 years later!

„Couldn't you make it any faster?!“

# The Traveling Salesman Problem

# The Traveling Salesman Problem

**Given:**



# The Traveling Salesman Problem

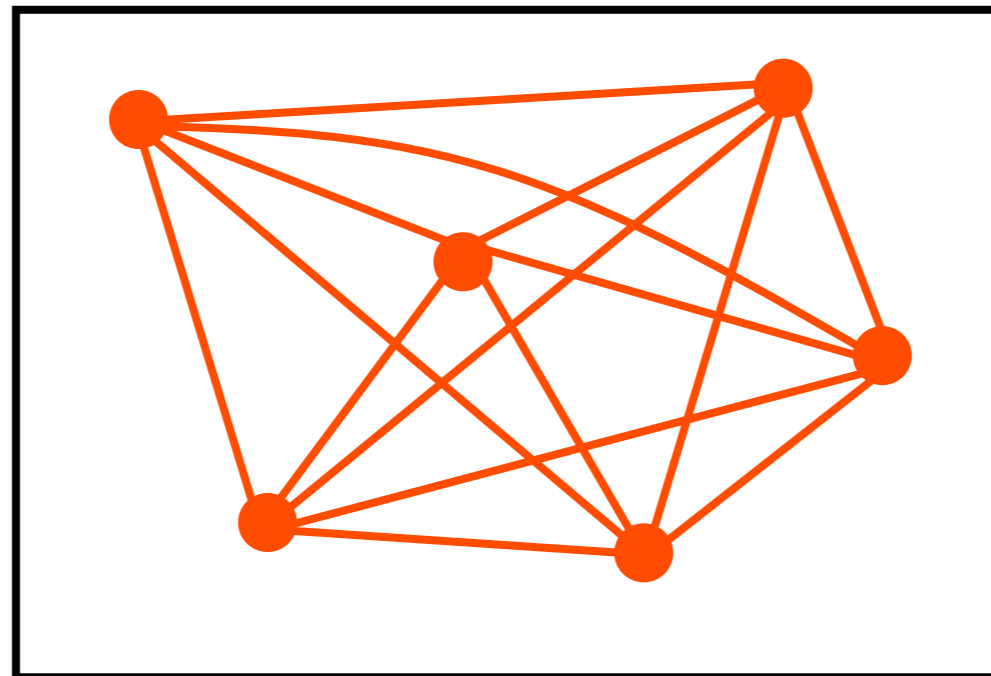
**Given:**

A graph  $G = (V, E)$   
with edge lengths  $w_e$

# The Traveling Salesman Problem

**Given:**

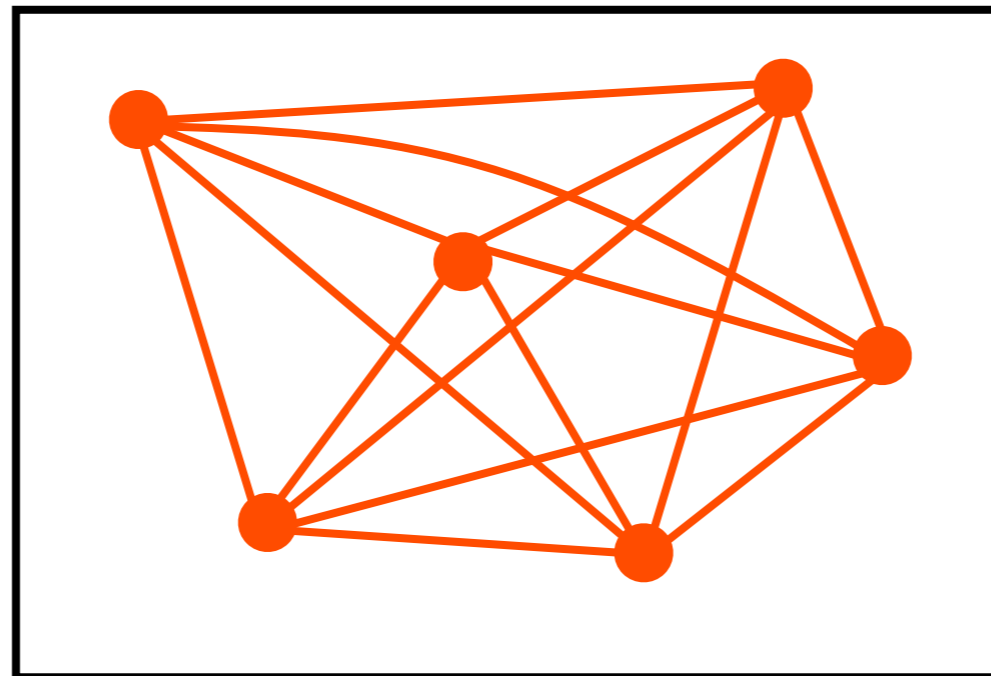
A graph  $G = (V, E)$   
with edge lengths  $w_e$



# The Traveling Salesman Problem

**Given:**

A graph  $G = (V, E)$   
with edge lengths  $w_e$

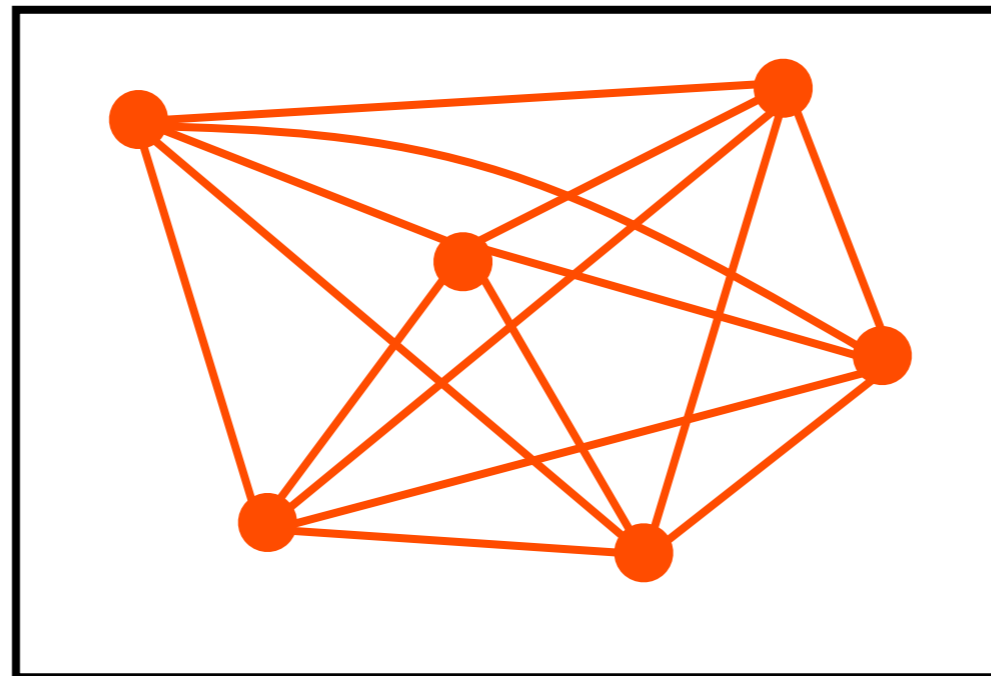


**Wanted:**

# The Traveling Salesman Problem

**Given:**

A graph  $G = (V, E)$   
with edge lengths  $w_e$



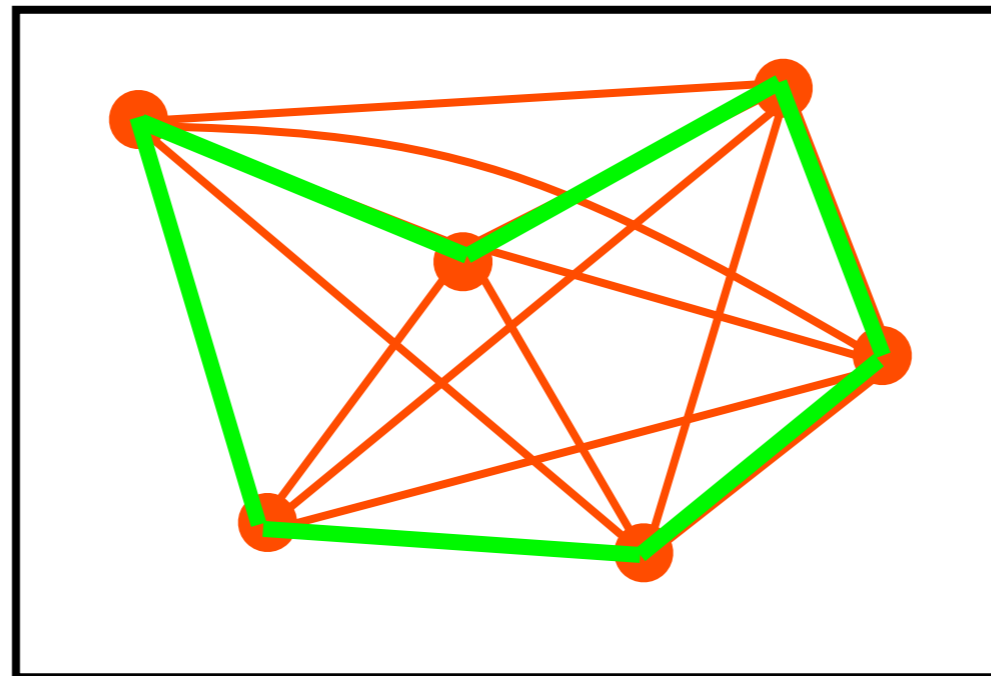
**Wanted:**

A shortest roundtrip through all vertices.

# The Traveling Salesman Problem

**Given:**

A graph  $G = (V, E)$   
with edge lengths  $w_e$



**Wanted:**

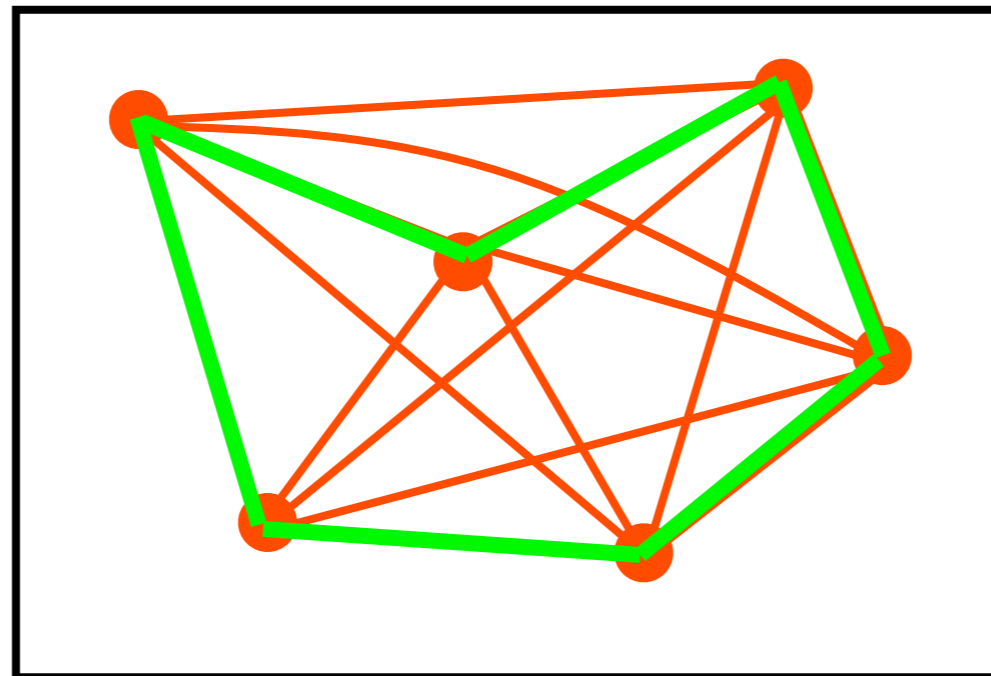
A shortest roundtrip through all vertices.



# The Traveling Salesman Problem

**Given:**

A graph  $G = (V, E)$   
with edge lengths  $w_e$



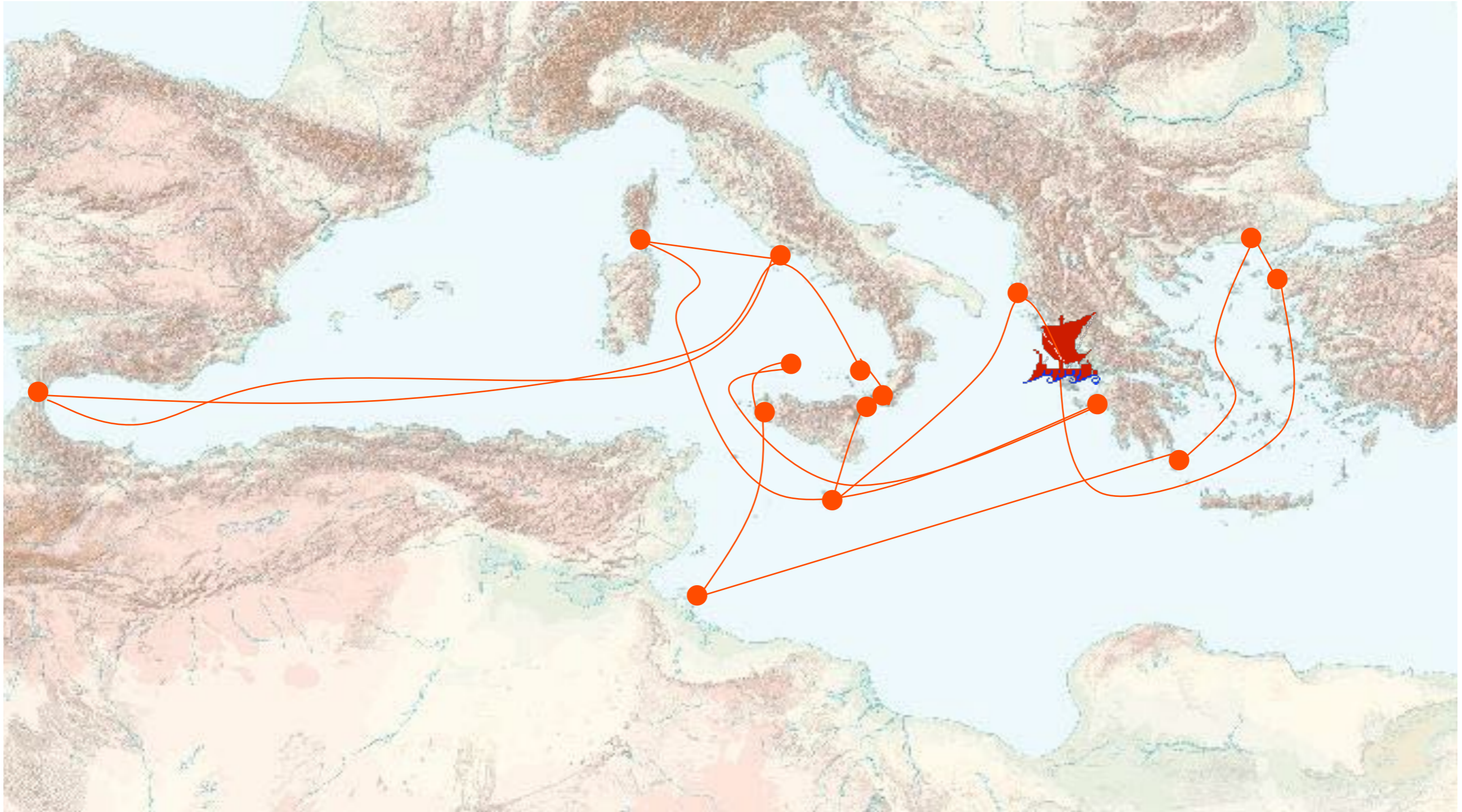
**Wanted:**

A shortest roundtrip through all vertices.

**Theorem:** (Karp 1972) Finding a shortest tour is NP-hard.

# Back to Odysseus

# Back to Odysseus



# Back to Odysseus



„Couldn't you make it any *slower*?“

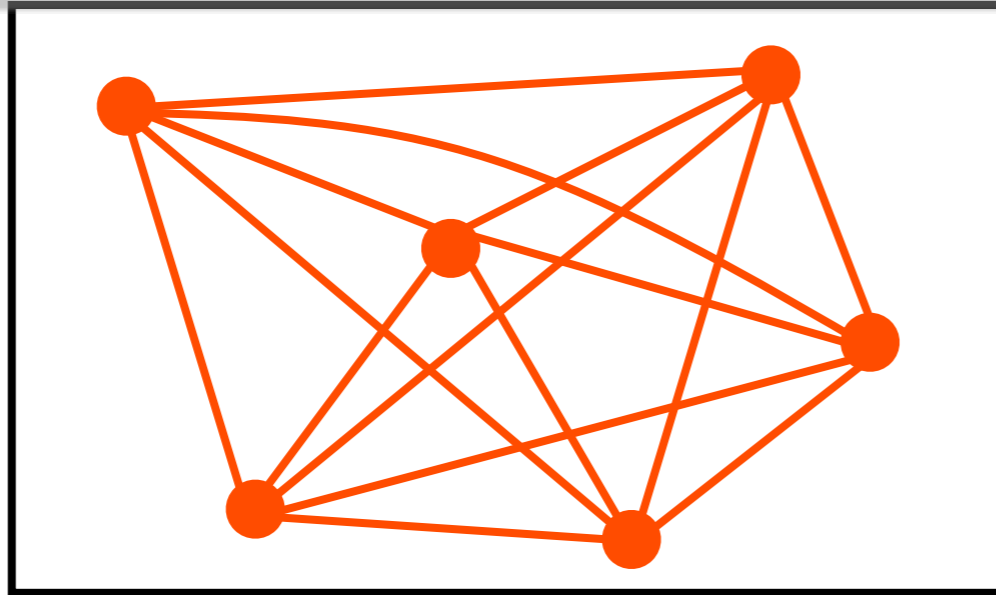
# The Maximum TSP

# The Maximum TSP

**Given:**

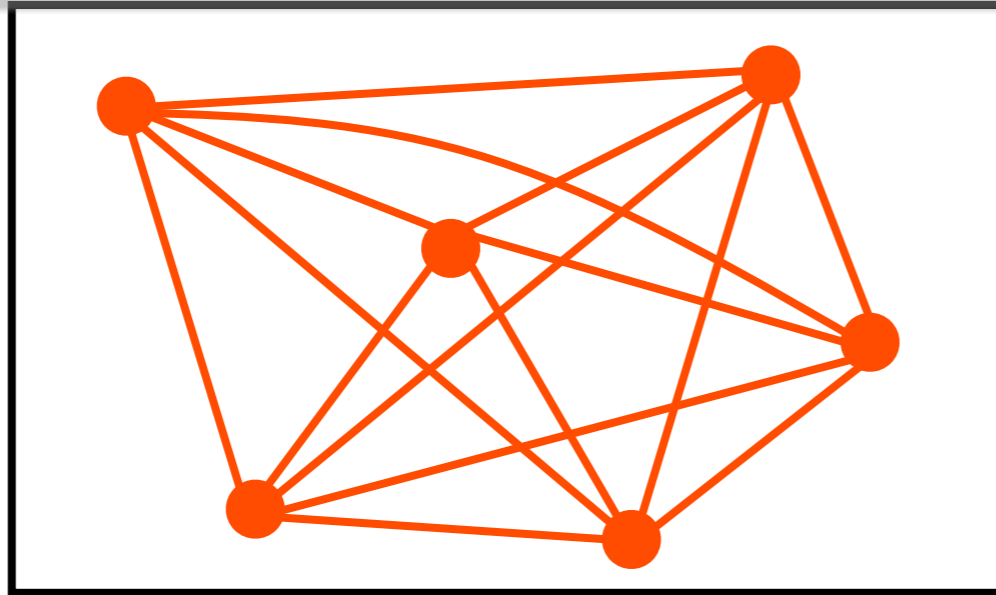
# The Maximum TSP

**Given:** A graph  $G = (V, E)$   
with edge lengths  $w_e$



# The Maximum TSP

**Given:** A graph  $G = (V, E)$   
with edge lengths  $w_e$

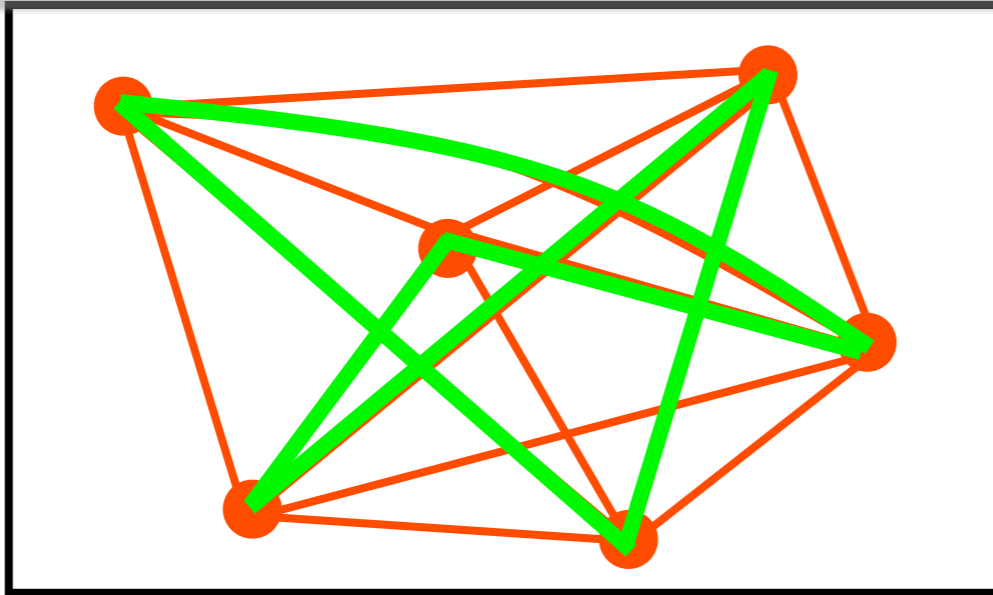


**Wanted:** A longest tour.



# The Maximum TSP

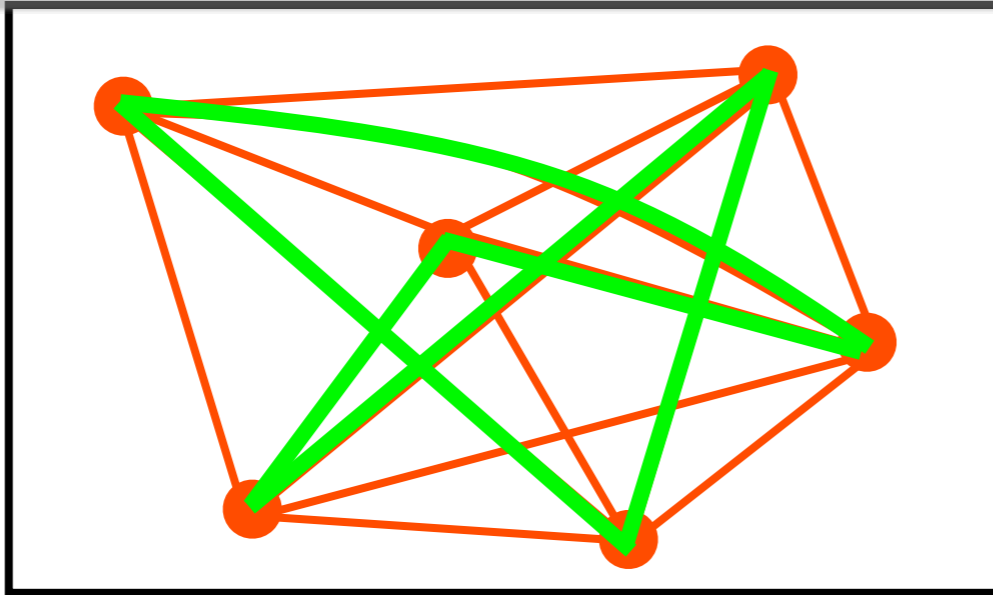
**Given:** A graph  $G = (V, E)$   
with edge lengths  $w_e$



**Wanted:** A longest tour.

# The Maximum TSP

**Given:** A graph  $G = (V, E)$   
with edge lengths  $w_e$



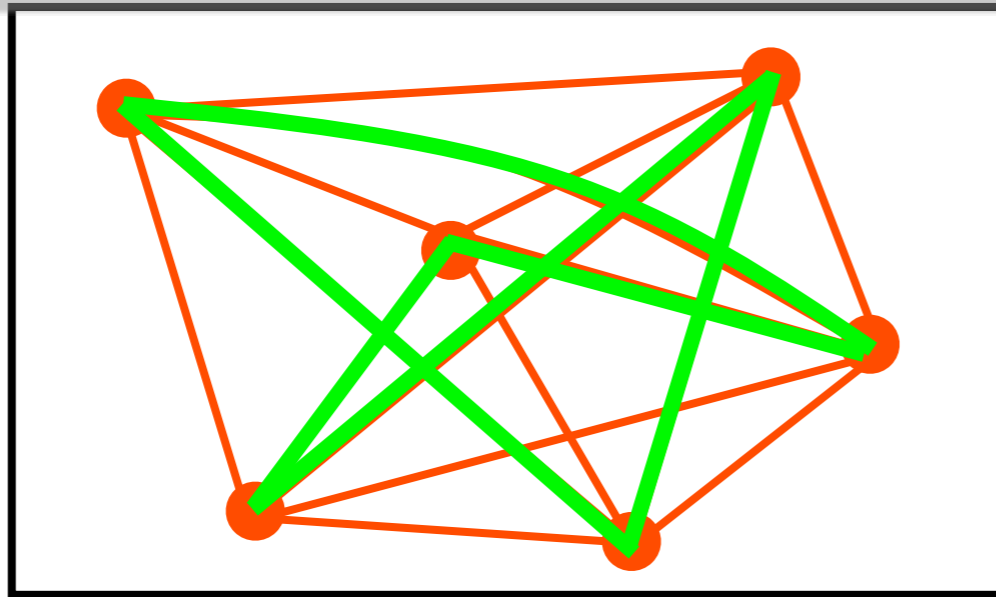
**Wanted:** A longest tour.

**Simple exercise:**

Finding a longest tour is NP-hard.

# The Maximum TSP

**Given:** A graph  $G = (V, E)$   
with edge lengths  $w_e$



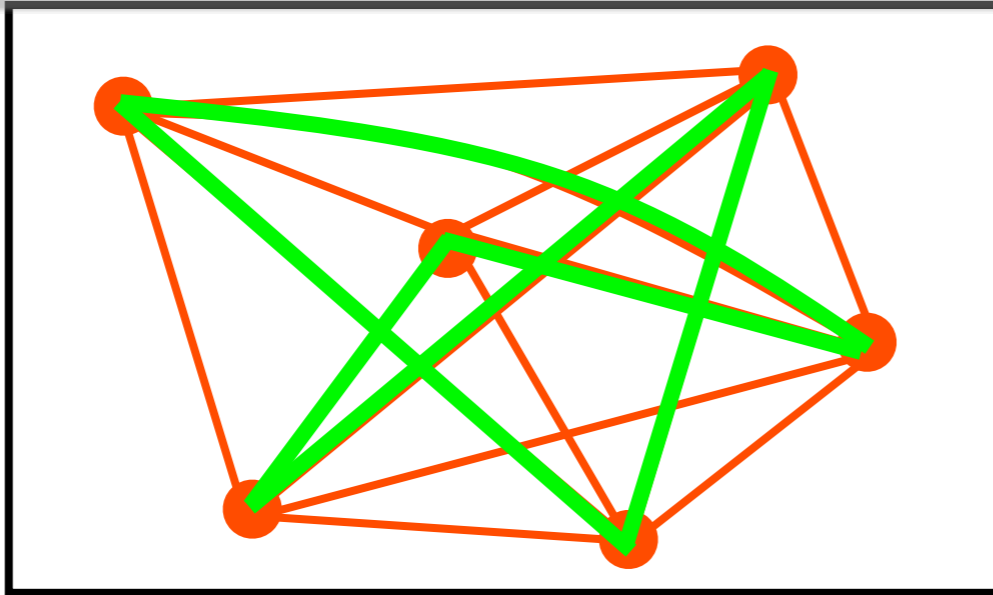
**Wanted:** A longest tour.

**Simple exercise:**

Finding a longest tour is NP-hard.

# The Maximum TSP

**Given:** A graph  $G = (V, E)$   
with edge lengths  $w_e$



**Wanted:** A longest tour.

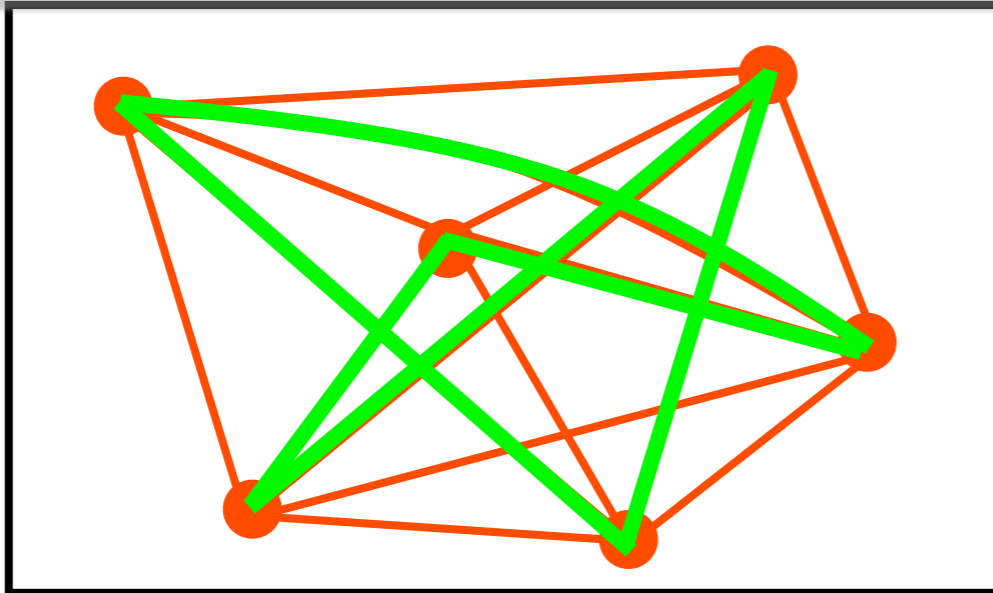
**Simple exercise:**

Finding a longest tour is NP-hard.

**Theorem:** (Serdyukov 1984)

# The Maximum TSP

**Given:** A graph  $G = (V, E)$   
with edge lengths  $w_e$



**Wanted:** A longest tour.

**Simple exercise:**

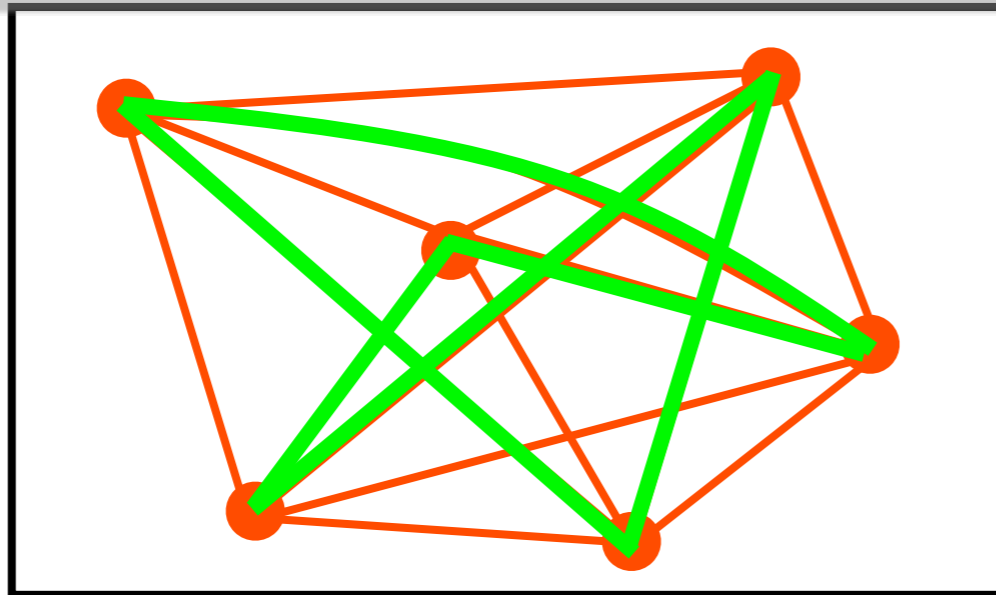
Finding a longest tour is NP-hard.

**Theorem:** (Serdyukov 1984)

For metric distances, the Maximum TSP can be approximated in polynomial time within a factor of  $3/4$ .

# The Maximum TSP

**Given:** A graph  $G = (V, E)$   
with edge lengths  $w_e$



**Wanted:** A longest tour.

**Simple exercise:**

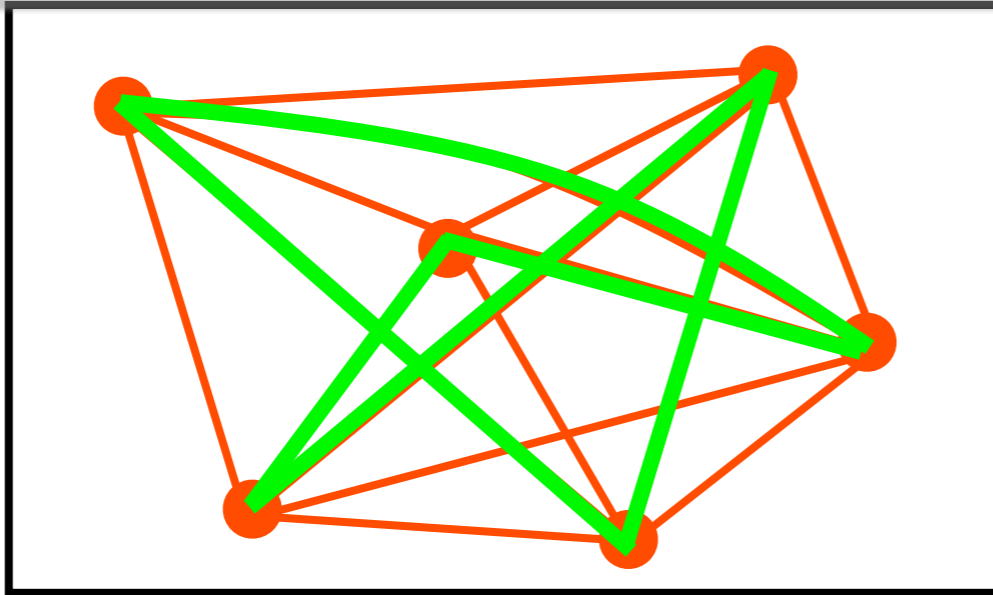
Finding a longest tour is NP-hard.

**Theorem:** (Serdyukov 1984)

For metric distances, the Maximum TSP can be approximated in polynomial time within a factor of  $3/4$ .

# The Maximum TSP

**Given:** A graph  $G = (V, E)$   
with edge lengths  $w_e$



**Wanted:** A longest tour.

**Simple exercise:**

Finding a longest tour is NP-hard.

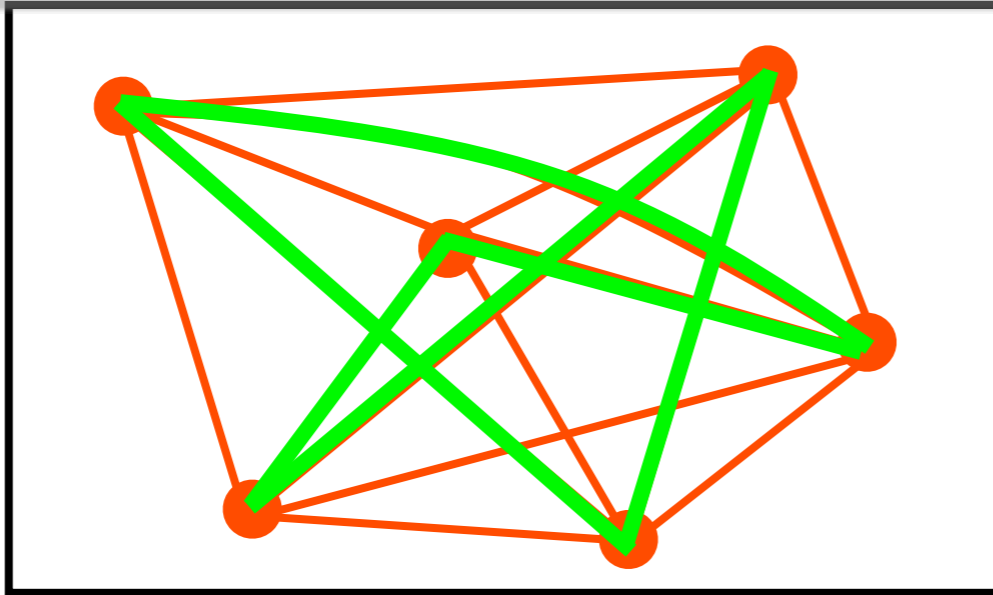
**Theorem:** (Serdyukov 1984)

For metric distances, the Maximum TSP can be approximated in polynomial time within a factor of  $3/4$ .

**Theorem:** (Barvinok 1996)

# The Maximum TSP

**Given:** A graph  $G = (V, E)$   
with edge lengths  $w_e$



**Wanted:** A longest tour.

**Simple exercise:**

Finding a longest tour is NP-hard.

**Theorem:** (Serdyukov 1984)

For metric distances, the Maximum TSP can be approximated in polynomial time within a factor of  $3/4$ .

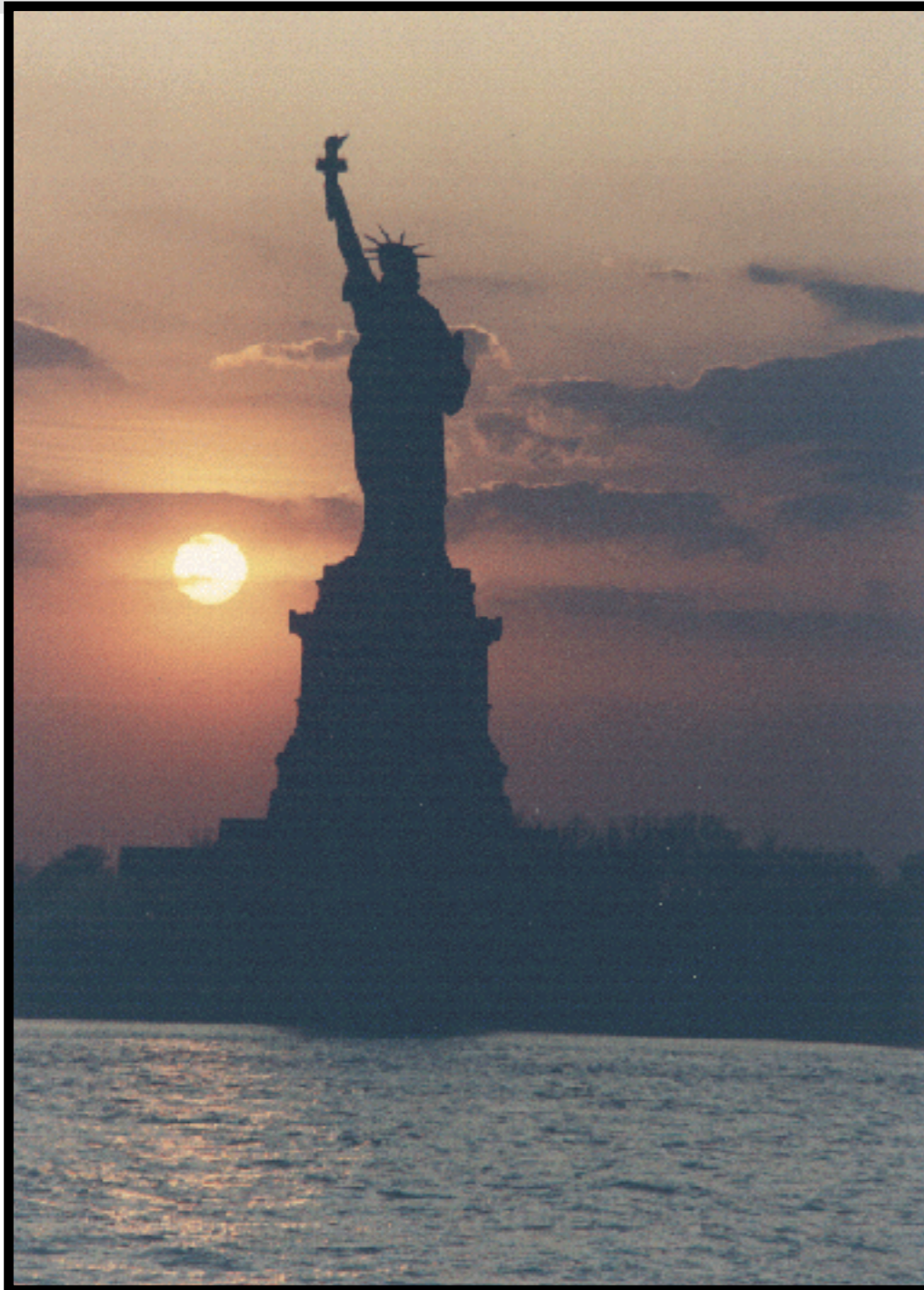
**Theorem:** (Barvinok 1996)

The Maximum TSP for geometric instances can be solved in polynomial time within a factor of  $1/(1+\epsilon)$ .



# Touring the New World

# Touring the New World

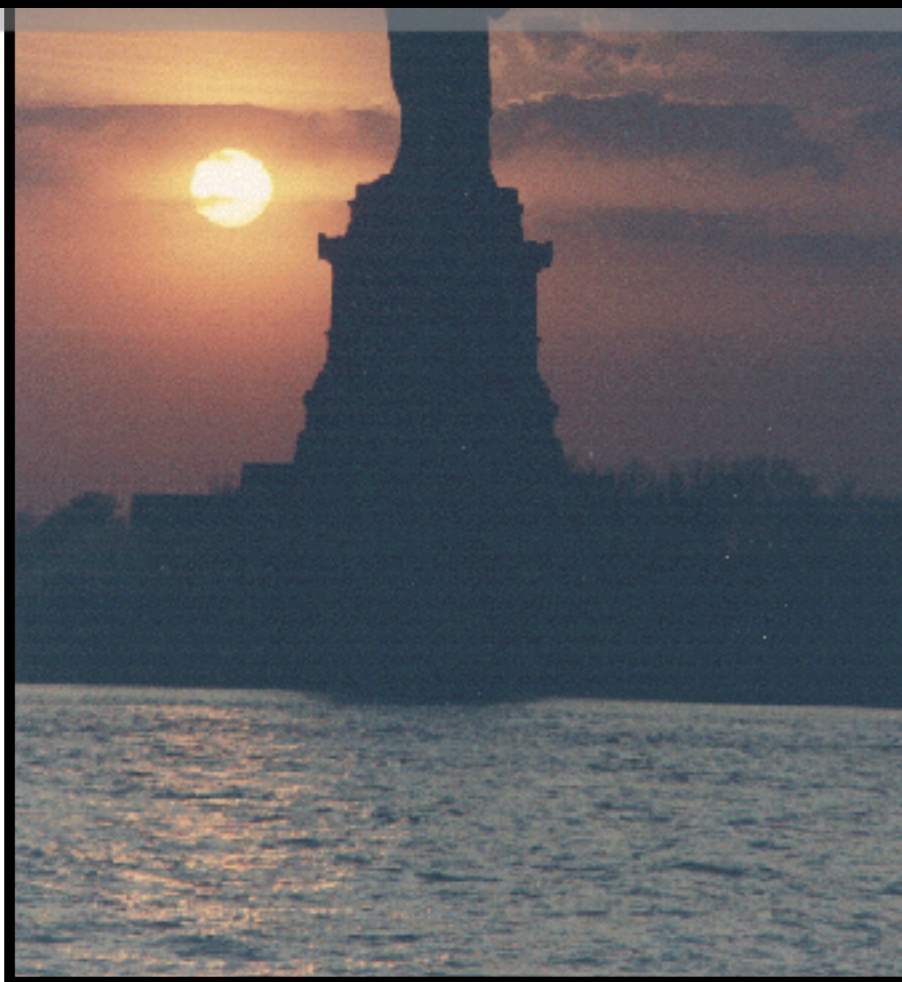


# Touring the New World



# Touring the New World

Distances:

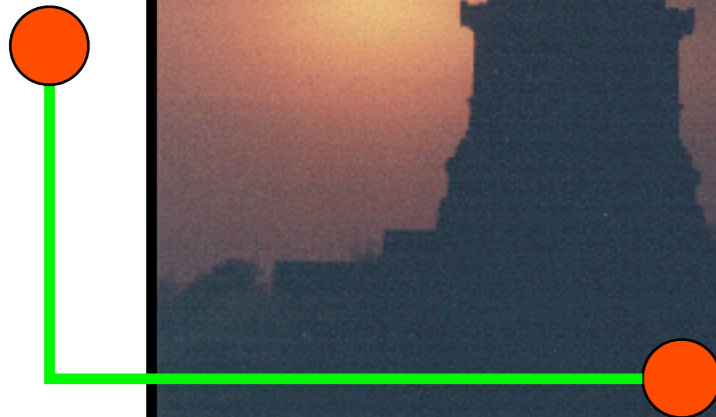
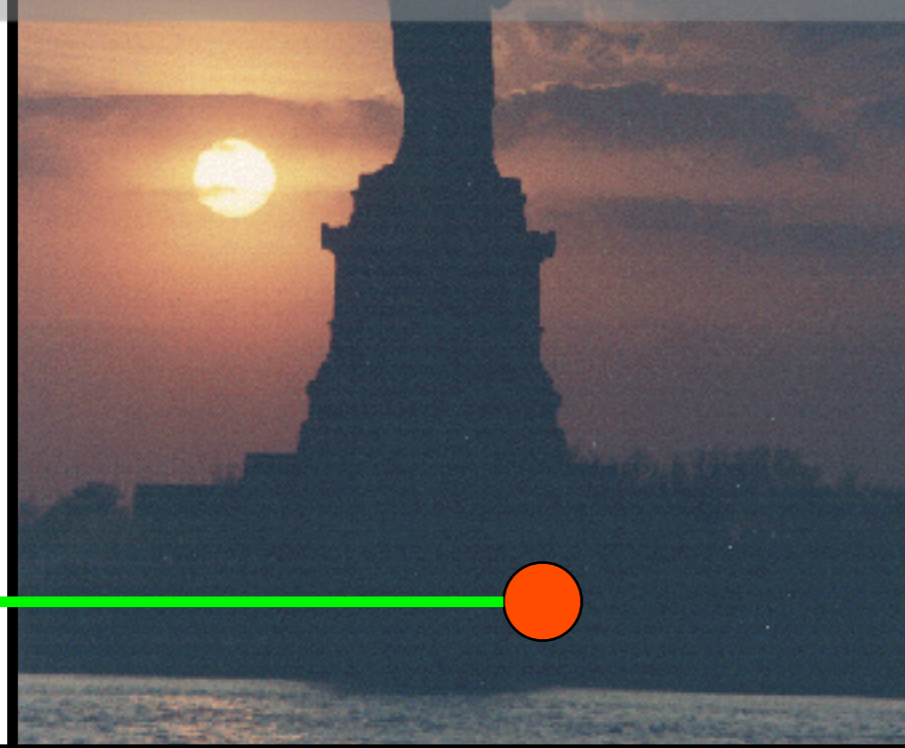


# Touring the New World

Distances:

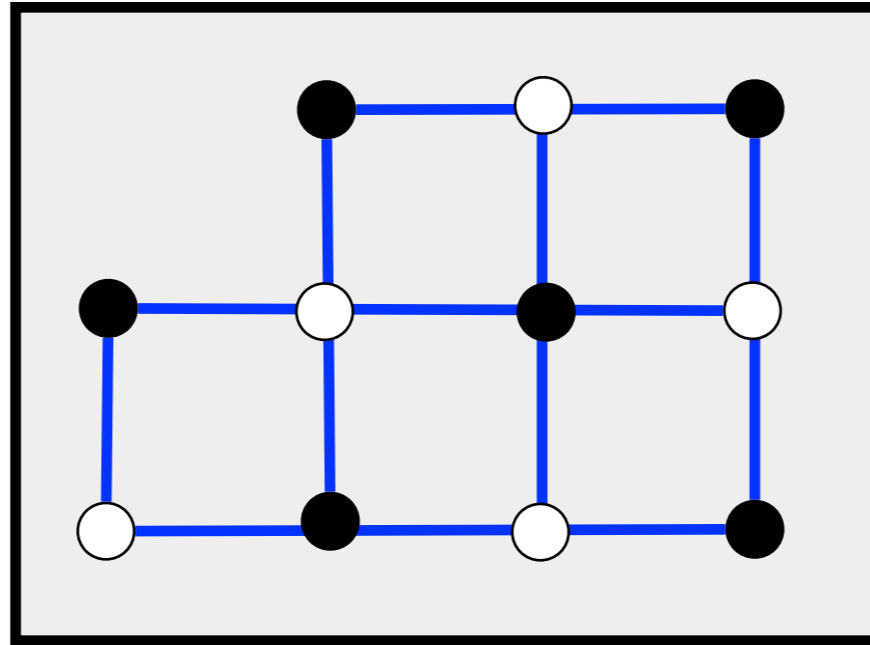
Axis-parallel

„Manhattan metric“

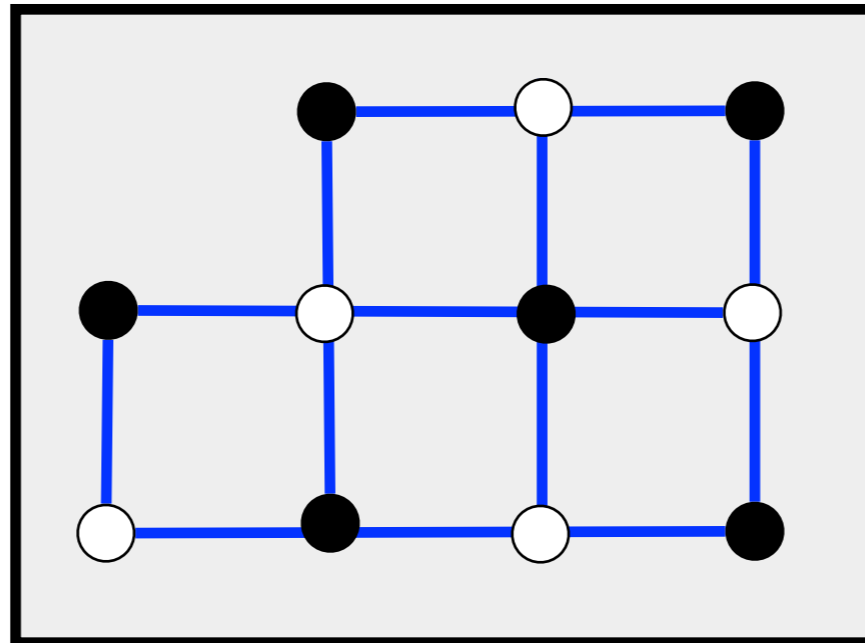


# Shortest Tours in Manhattan

# Shortest Tours in Manhattan



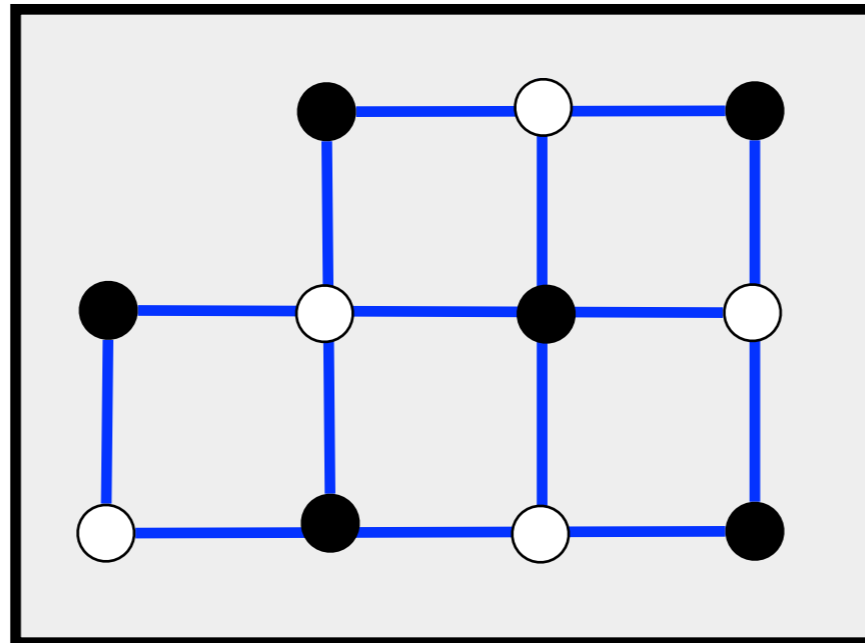
# Shortest Tours in Manhattan



A Grid Graph



# Shortest Tours in Manhattan



A Grid Graph

**Theorem:** (Itai, Papadimitriou, Szwarcfiter 1982)

## HAMILTON PATHS IN GRID GRAPHS\*

ALON ITAI†, CHRISTOS H. PAPADIMITRIOU‡ AND JAYME LUIZ SZWARCFITER§

**Abstract.** A grid graph is a node-induced finite subgraph of the infinite grid. It is rectangular if its set of nodes is the product of two intervals. Given a rectangular grid graph and two of its nodes, we give necessary and sufficient conditions for the graph to have a Hamilton path between these two nodes. In contrast, the Hamilton path (and circuit) problem for general grid graphs is shown to be NP-complete. This provides a new, relatively simple, proof of the result that the Euclidean traveling salesman problem is NP-complete.

**Key words.** Hamilton circuit, Hamilton path, grid graphs, rectangular grid graphs, NP-complete problem, Euclidean traveling salesman problem

**1. Introduction.** Let  $G^\infty$  be the infinite graph whose vertex set consists of all points of the plane with integer coordinates and in which two vertices are connected if and only if the (Euclidean) distance between them is equal to 1. A *grid graph* is a finite, node-induced subgraph of  $G^\infty$ . Thus, a grid graph is completely specified by its vertex set. Let  $v_x$  and  $v_y$  be the *coordinates* of the vertex  $v$ . We say that vertex  $v$  is *even* if  $v_x + v_y \equiv 0 \pmod{2}$ ; otherwise,  $v$  is *odd*. It is immediate that all grid graphs are bipartite, with the edges connecting even and odd vertices.

Let  $R(m, n)$  be the grid graph whose vertex set is  $V(R(m, n)) = \{v : 1 \leq v_x \leq m \text{ and } 1 \leq v_y \leq n\}$ . A *rectangular graph* is a grid graph which, for some  $m$  and  $n$ , is isomorphic to  $R(m, n)$ . Thus  $m$  and  $n$ , the *dimensions*, specify a rectangular graph up to isomorphism.

Let  $s$  and  $t$  be distinct vertices of a graph  $G$ . We say that the *Hamilton path problem*  $(G, s, t)$  has a solution if there exists a Hamilton path from  $s$  to  $t$  in  $G$ . In this paper we examine the Hamilton path problem for grid graphs; rectangular grid graphs were examined first in [LM]. In § 2 we show that the Hamilton path and Hamilton circuit problems for general grid graphs are NP-complete. Consider now a bipartite graph  $B = (V^0 \cup V^1, E)$ . If  $|V^0| - |V^1| + 1$ , then all Hamilton paths of  $B$  must start and end at vertices of  $V^0$ . If  $(R(m, n), s, t)$ , with  $m \times n$  odd, has a solution, then the number of even vertices is greater by one than that of the odd vertices. Hence, a necessary condition for the solvability of  $(R(m, n), s, t)$  is that both  $s$  and  $t$  be even. In § 3 it is shown that this condition is also *sufficient* for nontrivial (i.e.,  $m, n > 1$ ) odd rectangular graphs. If  $m \times n$  is even, then a solution is possible only if  $s$  and  $t$  have different parity. However, this condition is not sufficient. There are three families of configurations for which even though  $s$  and  $t$  have different parity  $(R(m, n), s, t)$  has no solution. In § 3 we give the precise necessary and sufficient conditions for a Hamilton path problem to have a solution. Partial results in this direction were first proved in [LM].

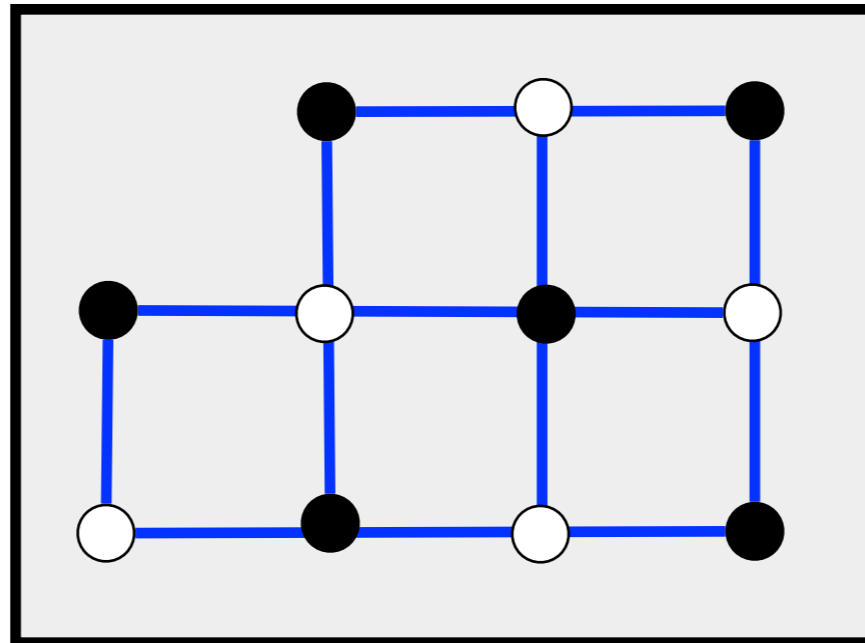
\* Received by the editors September 22, 1980, and in final form August 25, 1981.

† Department of Computer Science, Technion, Haifa, Israel. Part of this work was conducted while this author was visiting the Electrical Engineering and Computer Science Department, University of California at Berkeley, and the Laboratory for Computer Science, Massachusetts Institute of Technology.

‡ Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, and National Technical University of Athens, Greece. The work of this author was supported by the National Science Foundation under grant MCS 76-01193.

§ Universidade Federal do Rio de Janeiro, Brasil. Present address: Computer Science Division, University of California, Berkeley, California 94720. The work of this author was supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brasil, processo 574/78.

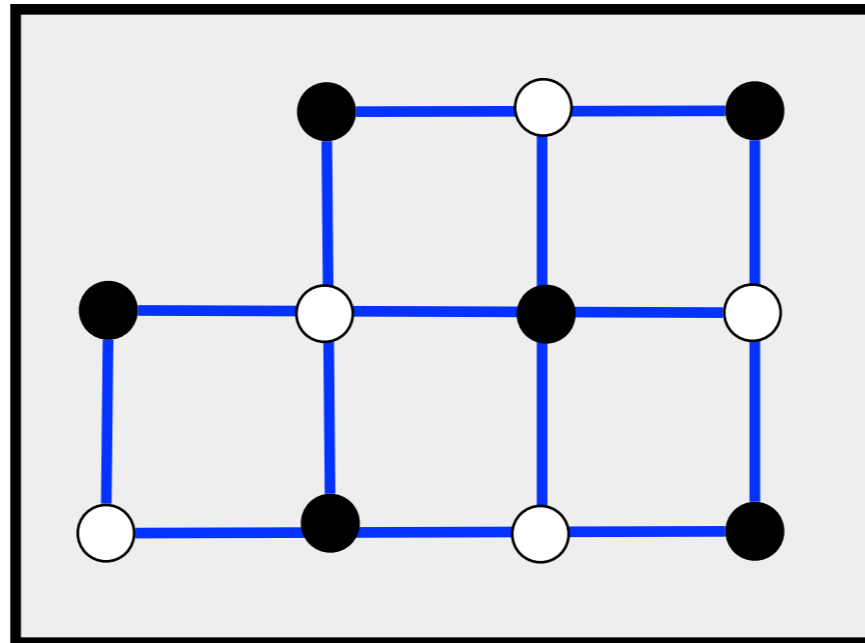
# Shortest Tours in Manhattan



A Grid Graph

**Theorem:** (Itai, Papadimitriou, Szwarcfiter 1982)

# Shortest Tours in Manhattan



A Grid Graph

**Theorem:** (Itai, Papadimitriou, Szwarcfiter 1982)

Finding a *shortest* tour in a grid graph is NP-hard.

# Longest Tours in Manhattan

## Simplicity and Hardness of the Maximum Traveling Salesman Problem under Geometric Distances\*

Sándor P. Fekete†

### Abstract

Recently, Barvinok, Johnson, Woeginger, and Woodroofe have shown that the *Maximum TSP*, i. e., the problem of finding a traveling salesman tour of maximum length, can be solved in polynomial time, provided that distances are computed according to a polyhedral norm in  $\mathbb{R}^d$ , for some fixed  $d$ . The most natural case of this class of problems arises for rectilinear distances in the plane  $\mathbb{R}^2$ , where the unit ball is a square. With the help of some additional improvements by Tamir, the method by Barvinok et al. yields an  $O(n^2 \log n)$  algorithm for this case by making elegant use of geometry, graph theory, and optimization, including some rather powerful tools.

In this paper, we present a simple algorithm with  $O(n)$  running time for computing the length of a longest tour for a set of points in the plane with rectilinear distances. The algorithm does not use any indirect addressing, so its running time remains valid even in comparison based models in which sorting requires  $\Omega(n \log n)$  time, which implies the same lower bound on verifying a Hamiltonian cycle. In addition, our approach gives a simple characterization of *all* optimal solutions. These results give a good idea what makes the (polyhedral) max TSP so much easier than its minimization counterpart.

Resolving the complexity status of the max TSP for Euclidean distances in spaces of fixed dimension has been stated by Barvinok et al. as a main open problem. In this paper, the results on simplicity are complemented by a proof that the Maximum TSP under Euclidean distances in  $\mathbb{R}^d$  for any fixed  $d \geq 3$  is NP-hard, shedding new light on the well-studied difficulties of Euclidean distances. In addition, our result implies NP-hardness of the Maximum TSP under polyhedral norms if the number  $k$  of facets of the unit ball is

not fixed. As a corollary, we get NP-hardness of the Maximum Scatter TSP for geometric instances, where the objective is to find a tour that maximizes the shortest edge. This resolves a conjecture by Arkin, Chiang, Mitchell, Skiena, and Yang in the affirmative.

### 1 Introduction

The Traveling Salesman Problem (TSP) is one of the classical problems of combinatorial optimization: Given a set  $\{v_1, v_2, \dots, v_n\}$  of vertices together with the distance  $d(v_i, v_j)$  between every pair of distinct vertices  $v_i, v_j$ , the goal is to find a permutation  $\pi$  of the vertices (a "tour") that minimizes (Minimum TSP) or maximizes (Maximum TSP) the total tour length

$$d(v_{\pi(n)}, v_{\pi(1)}) + \sum_{i=1}^{n-1} d(v_{\pi(i)}, v_{\pi(i+1)}).$$

Geometric instances of the TSP have always been of particular interest: vertices  $v_i$  correspond to points  $p_i = (x_1, \dots, x_d)$  in space  $\mathbb{R}^d$ , and distances  $d(v_i, v_j)$  are given by some geometric norm  $\|p_i - p_j\|$ . The most common norms considered include the Euclidean norm  $L_2$  and the Manhattan norm  $L_1$ , which are both special cases of the  $L_p$  norms. The  $L_1$  norm is also an example of a *polyhedral norm*, where the set of points at distance 1 from the origin is given by a centrally symmetric polyhedron with  $k$  facets.

Two key questions regarding the complexity of the Minimum TSP on geometric instances have been answered. Itai, Papadimitriou, and Swarcfiter [11] showed that the Minimum TSP is NP-hard for any fixed dimension  $d \geq 2$  and any  $L_p$  or polyhedral norm. On the other hand, Arora [2, 3] and Mitchell [13] showed that all these geometric instances allow a polynomial-time approximation scheme (PTAS), i. e., a sequence of algorithms  $A_s$  that compute a solution within a factor of  $1 + \frac{1}{s}$  of the optimum, in time that is polynomial for any fixed  $s$ . This highlights the special role of geometry, since it is well known that for general Minimum TSP instances, no PTAS can exist. (See Trevisan [17], and Papadimitriou and Yannakakis [14] for such results in more restricted situations.)

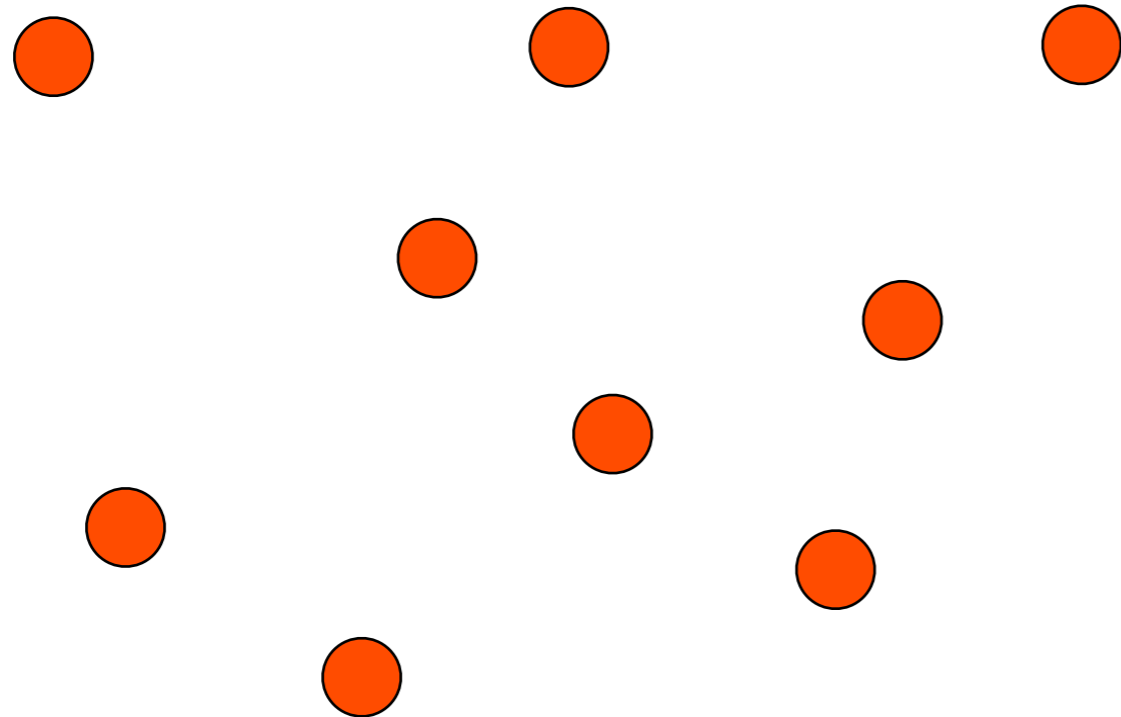
\*Work on this paper was partially supported by the Hermann-Minkowski-Minerva Center for Geometry at Tel Aviv University, while the author was visiting the Center in March 1998, and by DFG travel grant FE 407/3-1 for a visit to the USA in June 1998.

†Center for Applied Computer Science, Universität zu Köln, 50923 Köln, GERMANY, sandor@zpr.uni-koeln.de



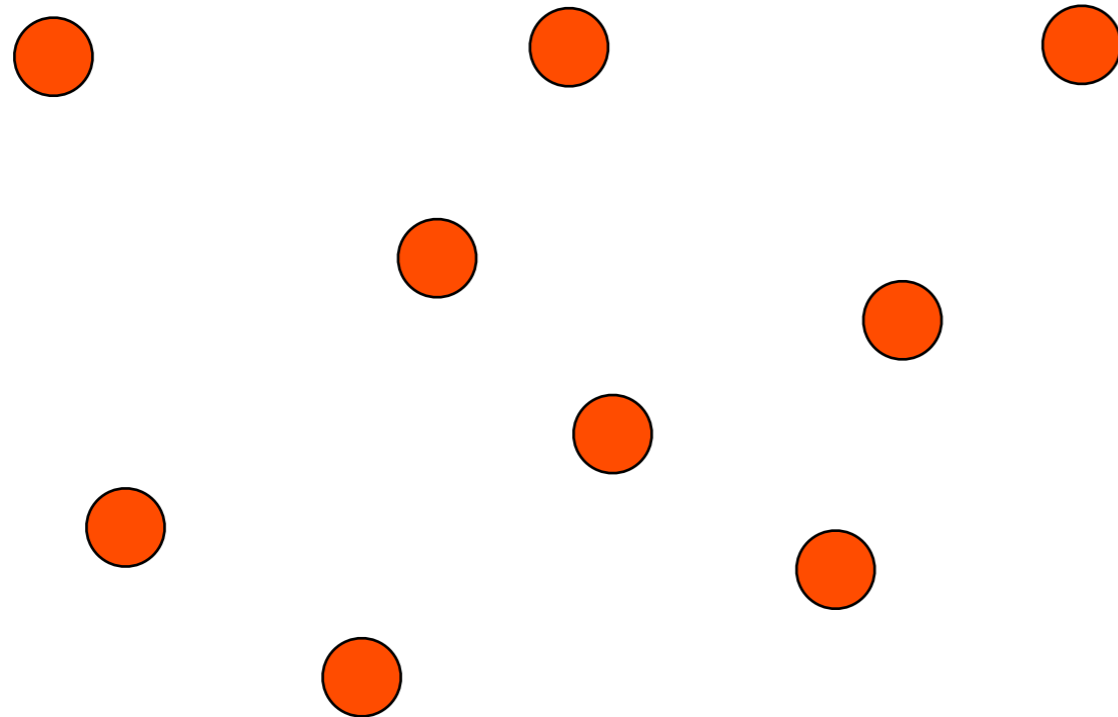
# Longest Tours in Manhattan

# Longest Tours in Manhattan



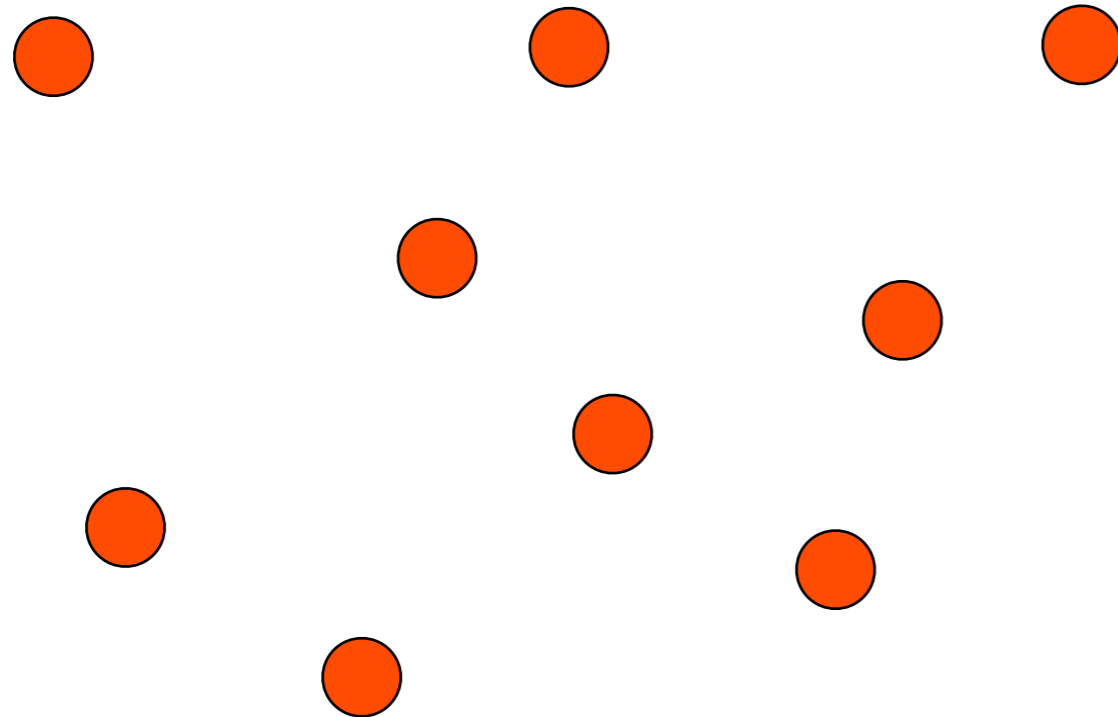


# Longest Tours in Manhattan



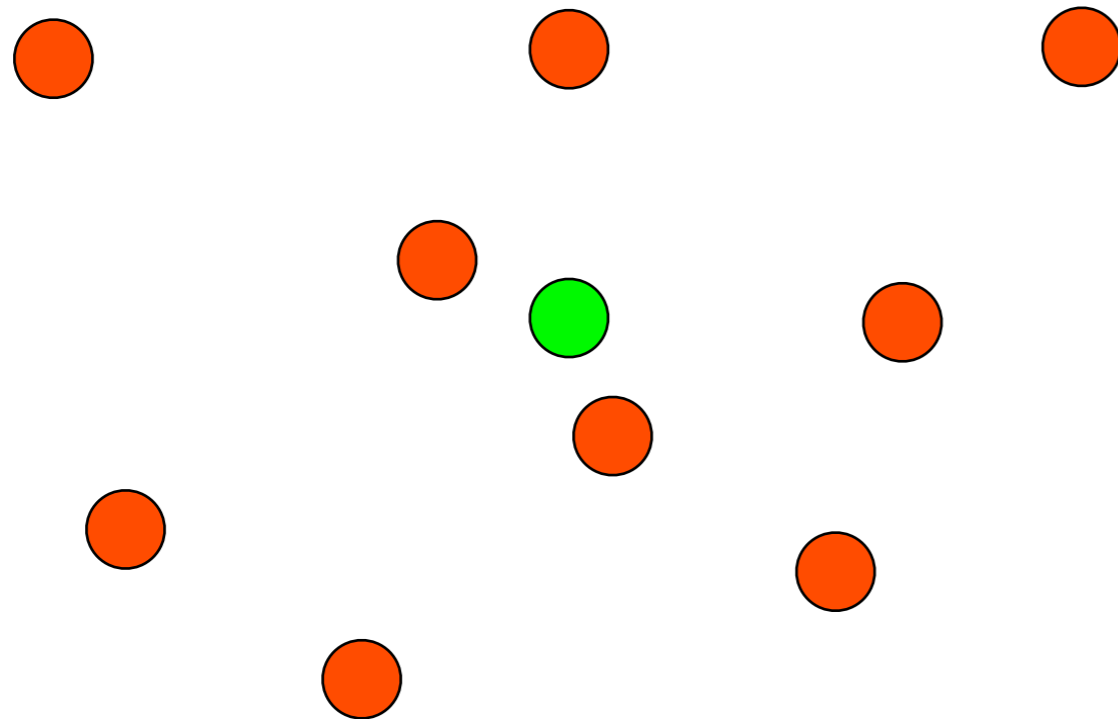
# Longest Tours in Manhattan

- Find a point that minimizes the sum of Manhattan distances:



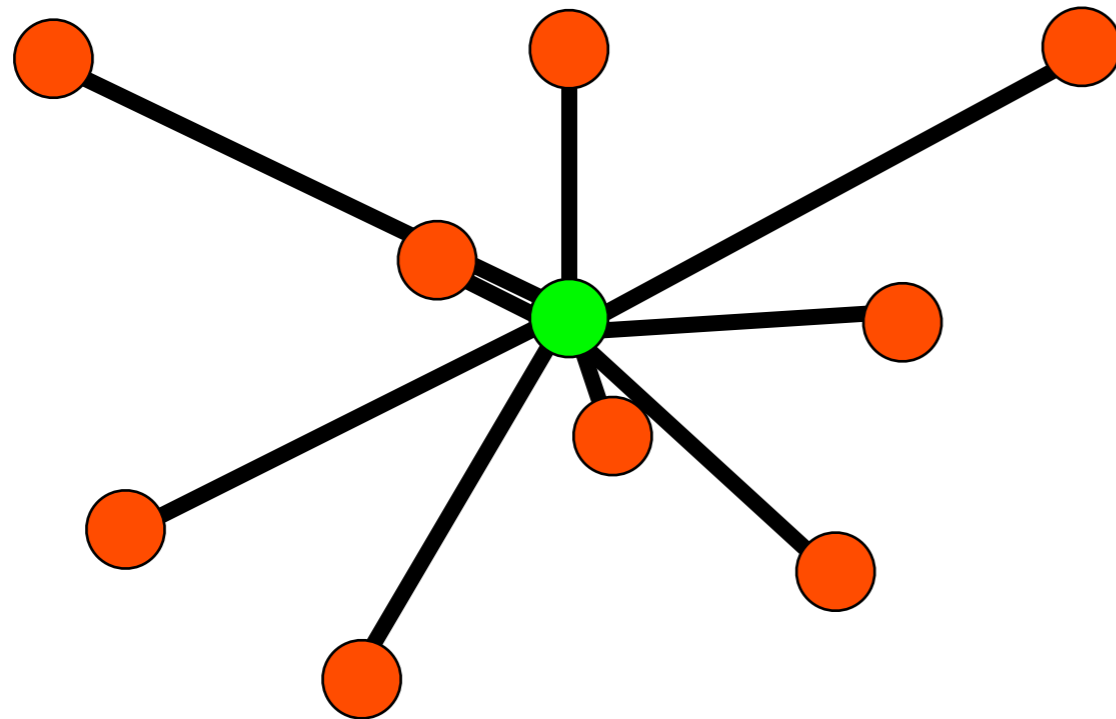
# Longest Tours in Manhattan

- Find a point that minimizes the sum of Manhattan distances:

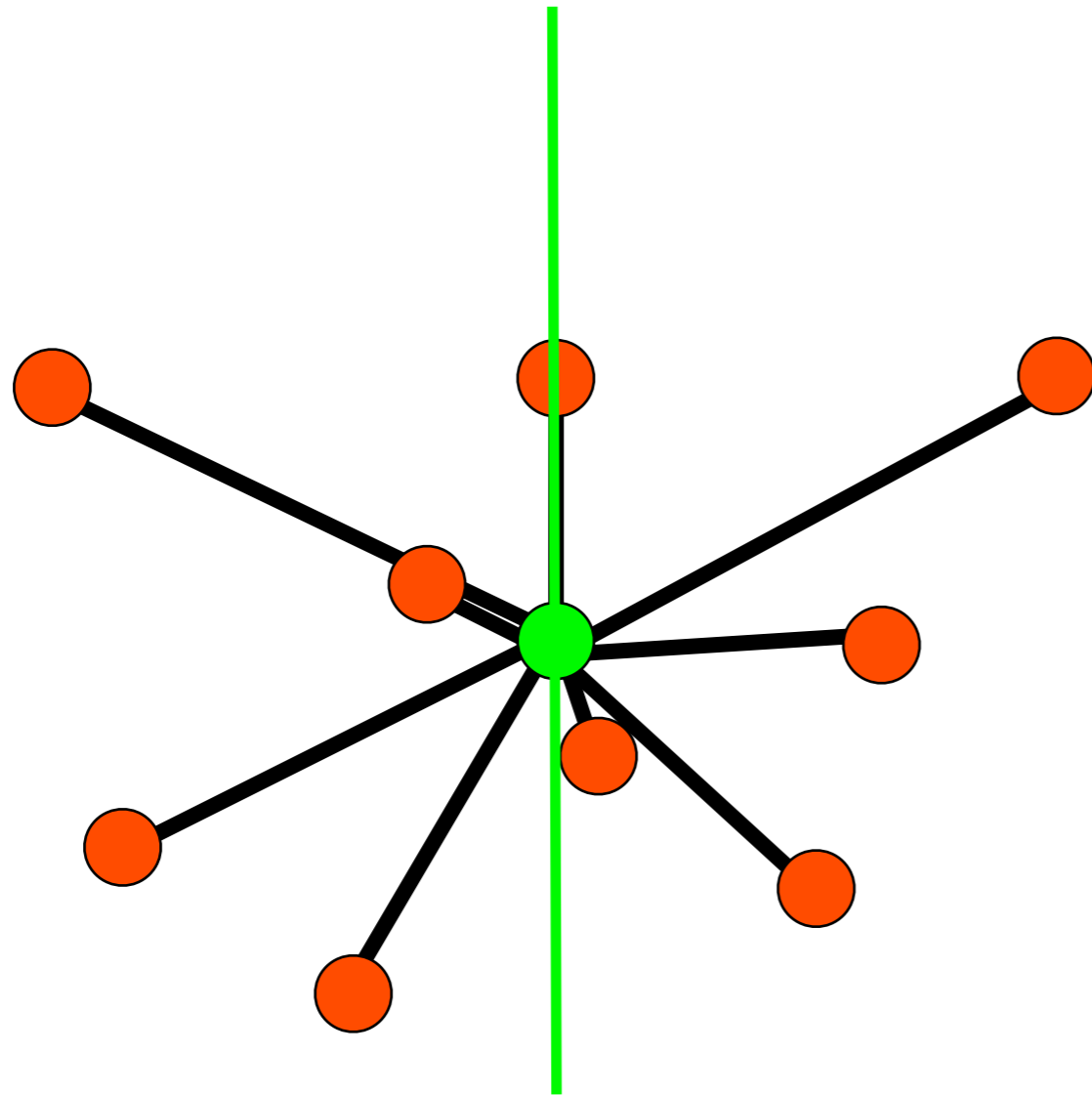


# Longest Tours in Manhattan

- Find a point that minimizes the sum of Manhattan distances:

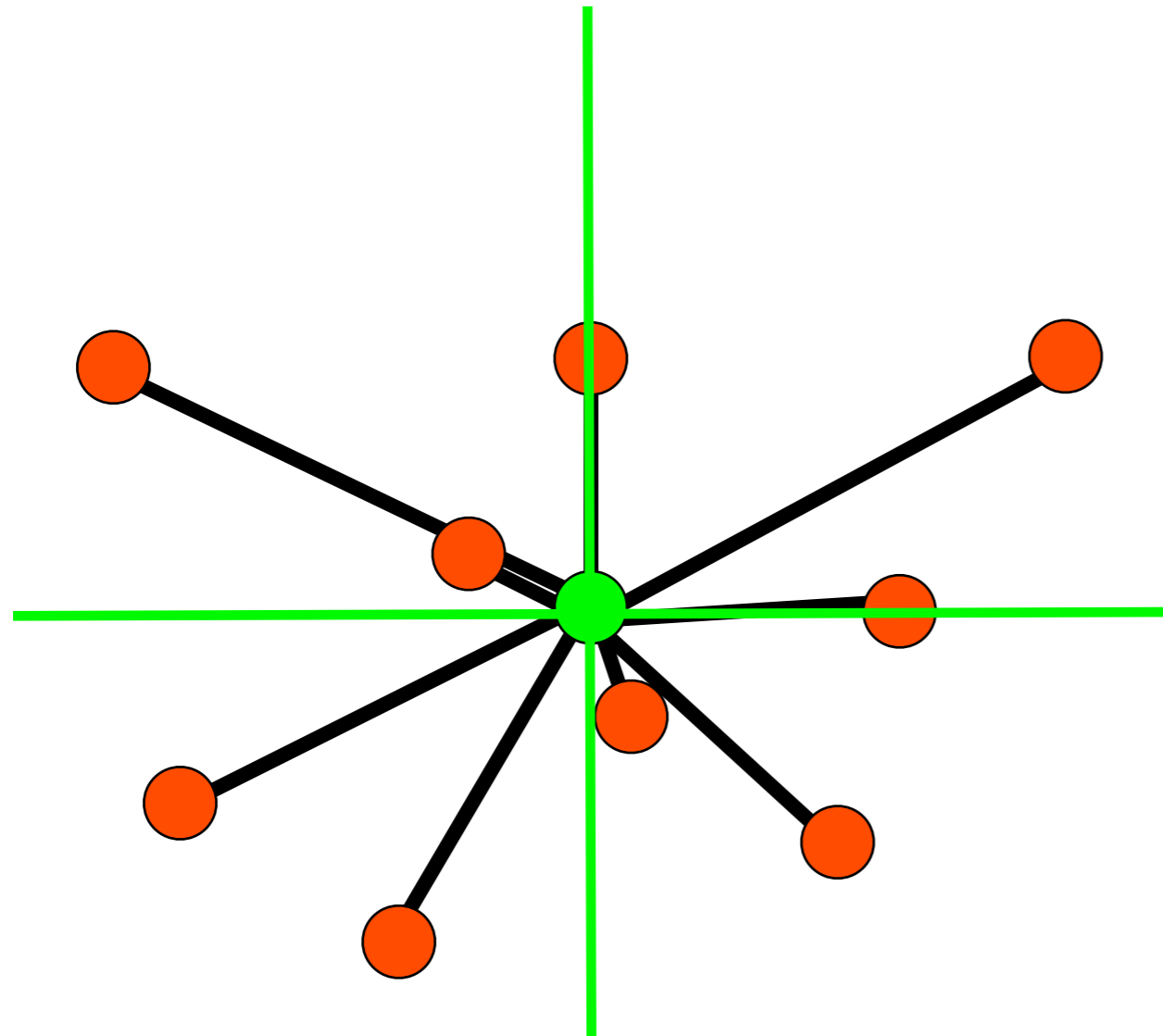


# Longest Tours in Manhattan



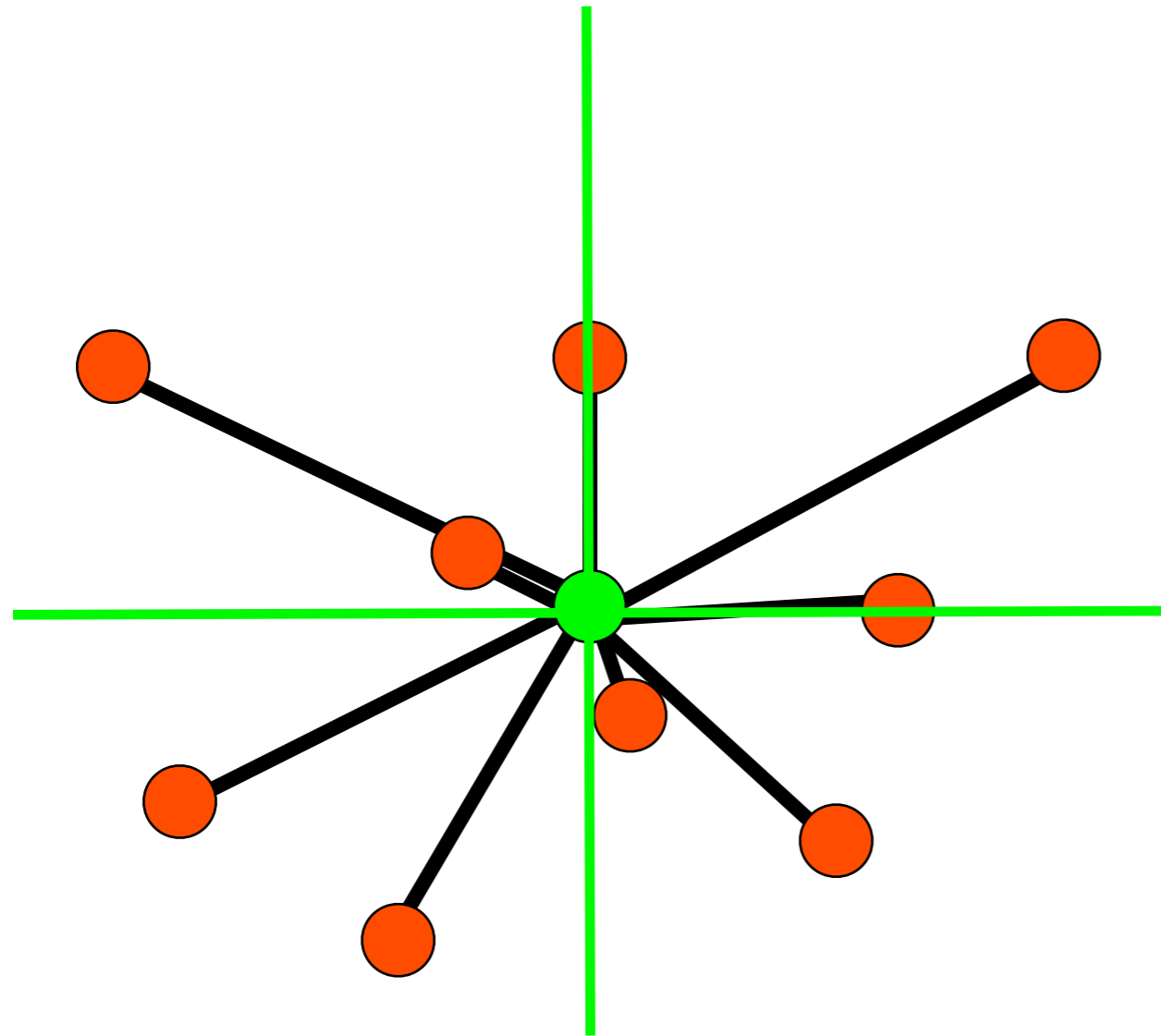
- Find a point that minimizes the sum of Manhattan distances:

# Longest Tours in Manhattan



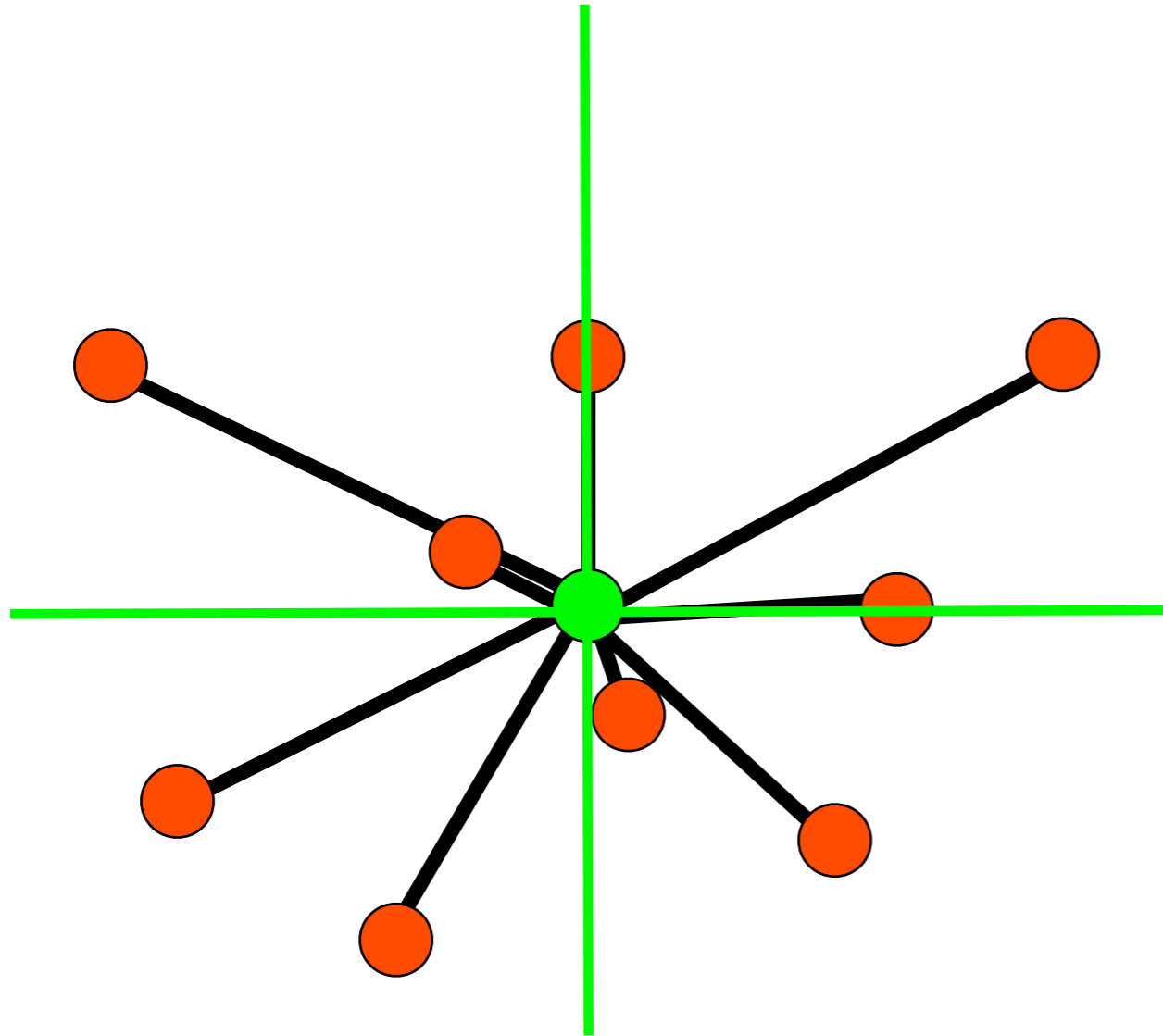
- Find a point that minimizes the sum of Manhattan distances:

# Longest Tours in Manhattan



- Find a point that minimizes the sum of Manhattan distances: median in x and y!

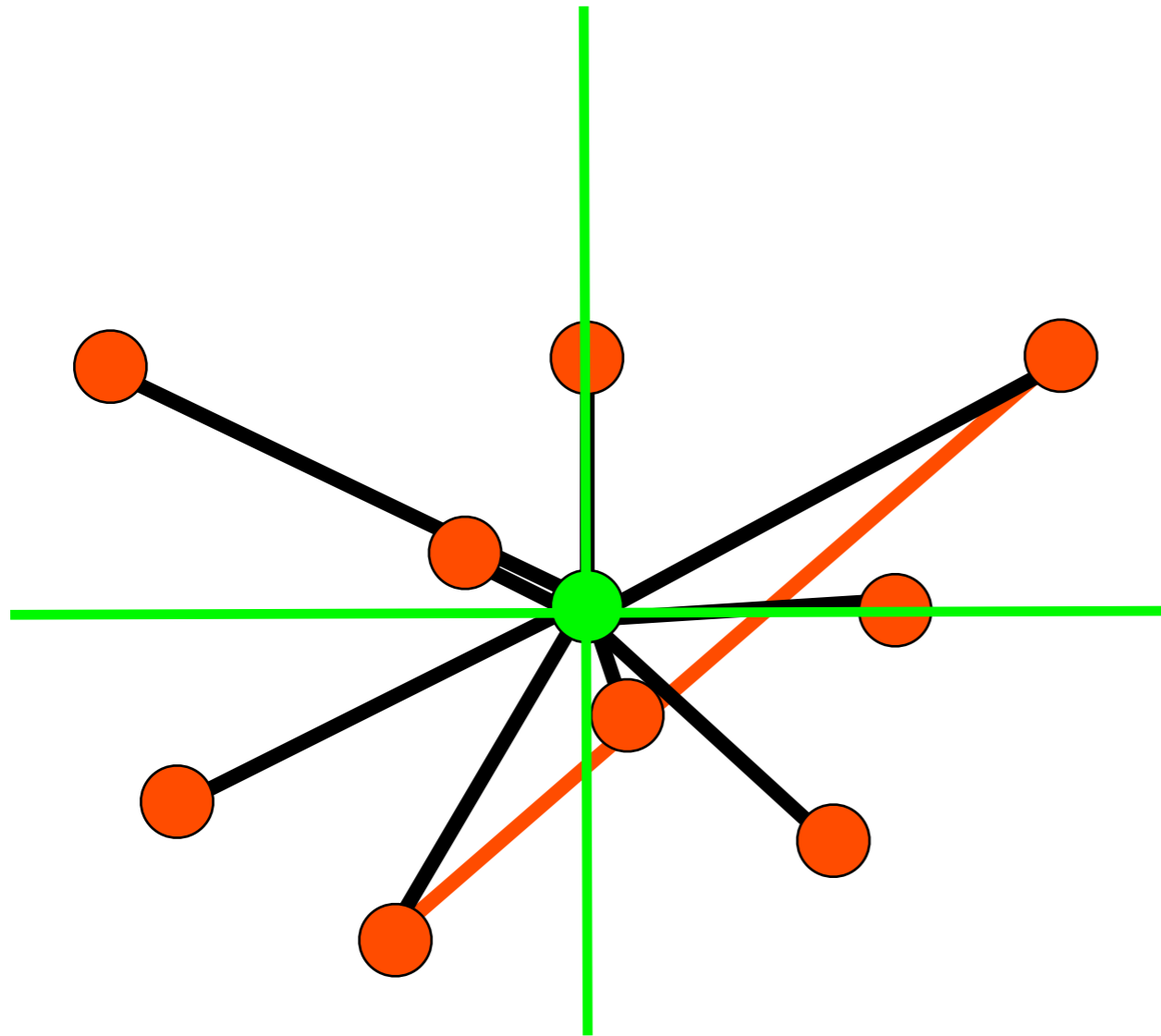
# Longest Tours in Manhattan



- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.

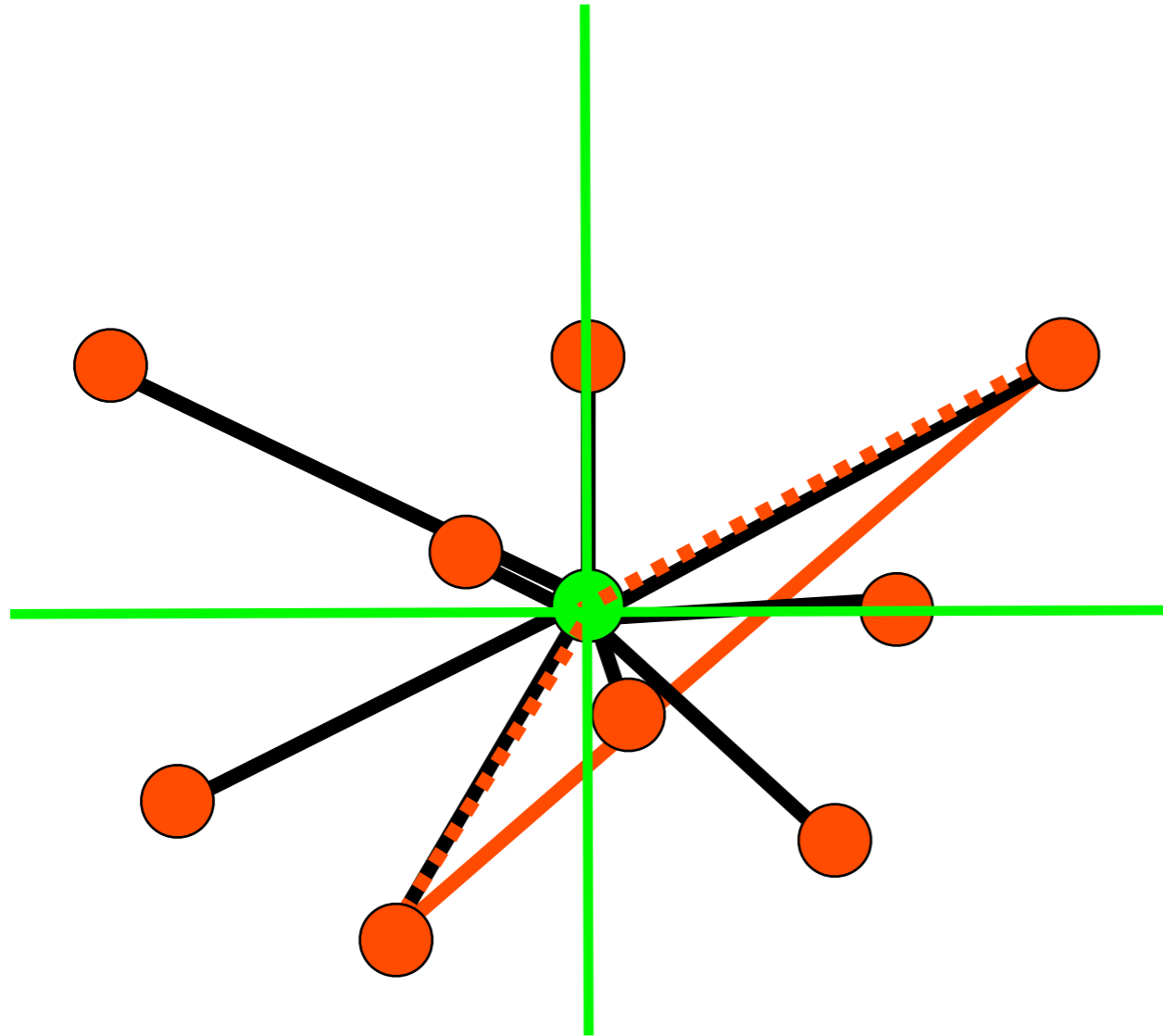


# Longest Tours in Manhattan



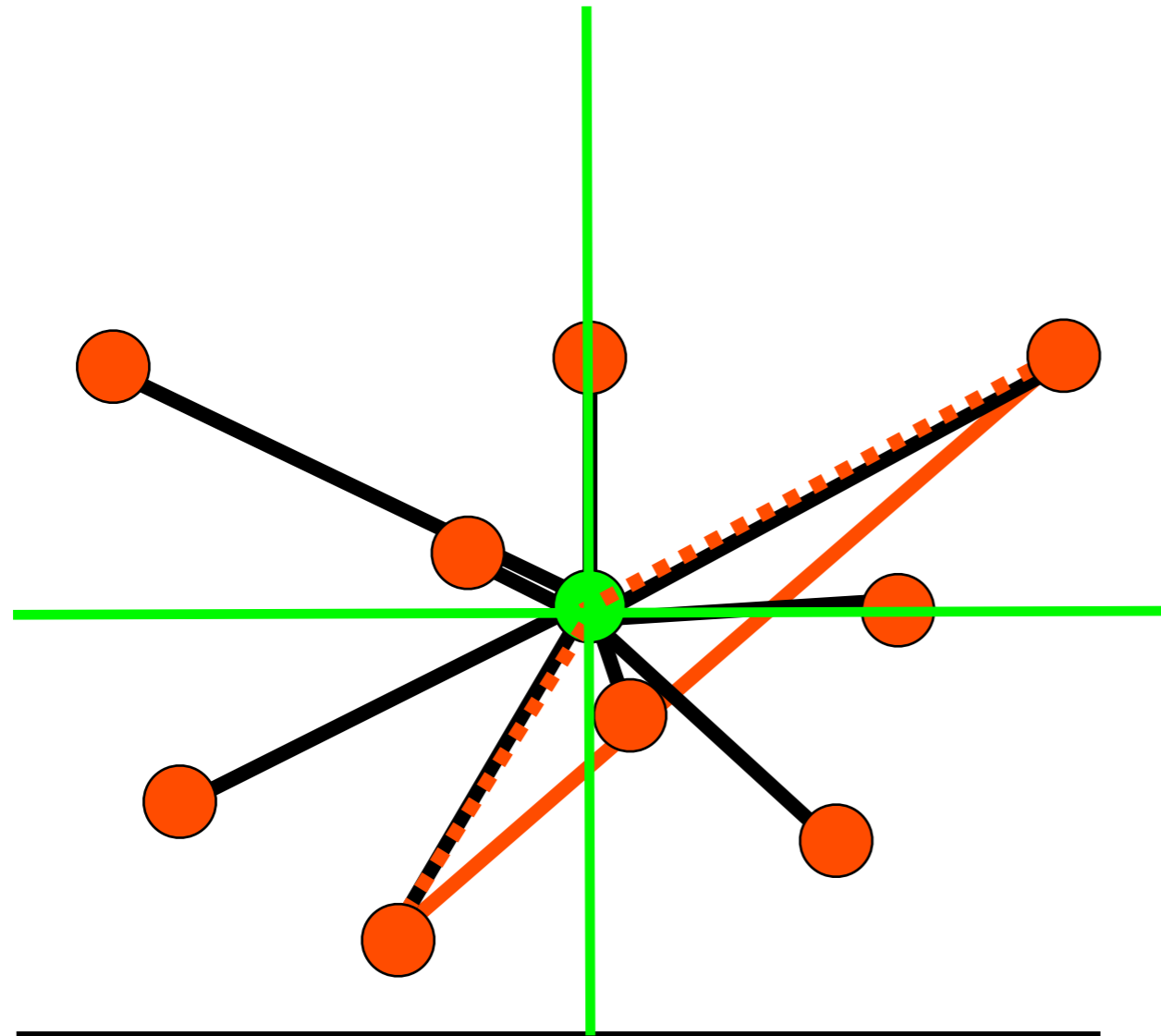
- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.

# Longest Tours in Manhattan



- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.

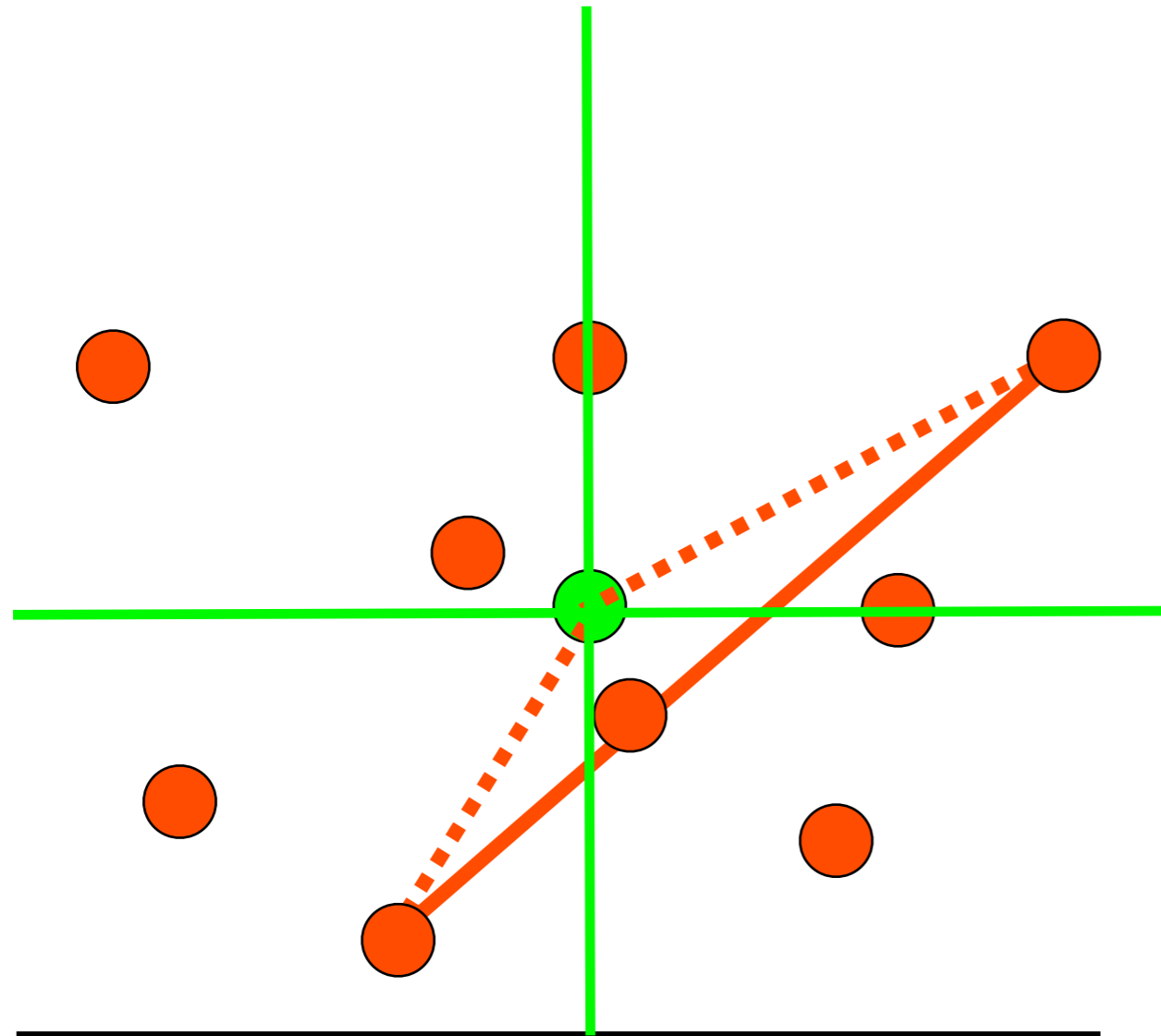
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.

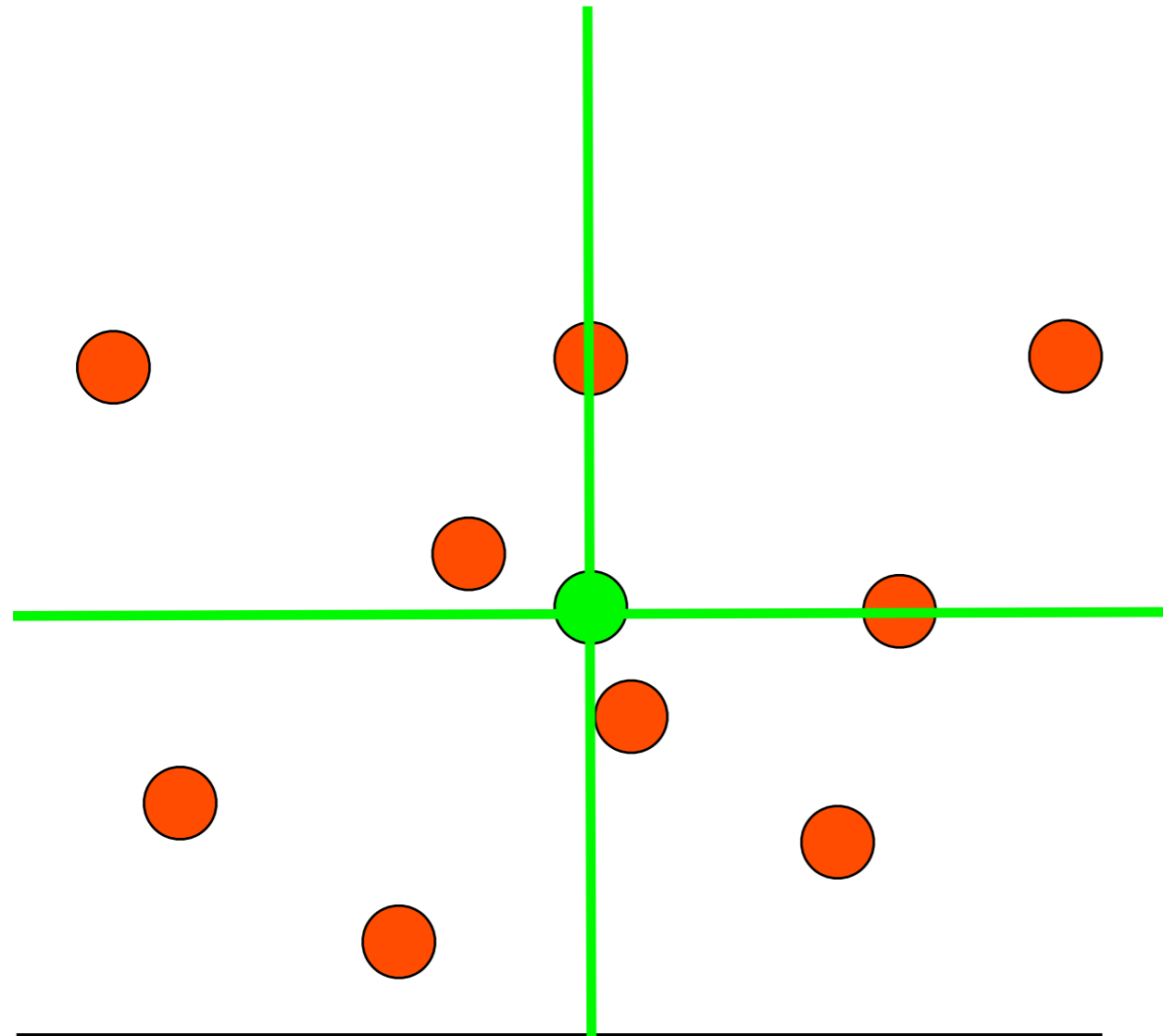
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.

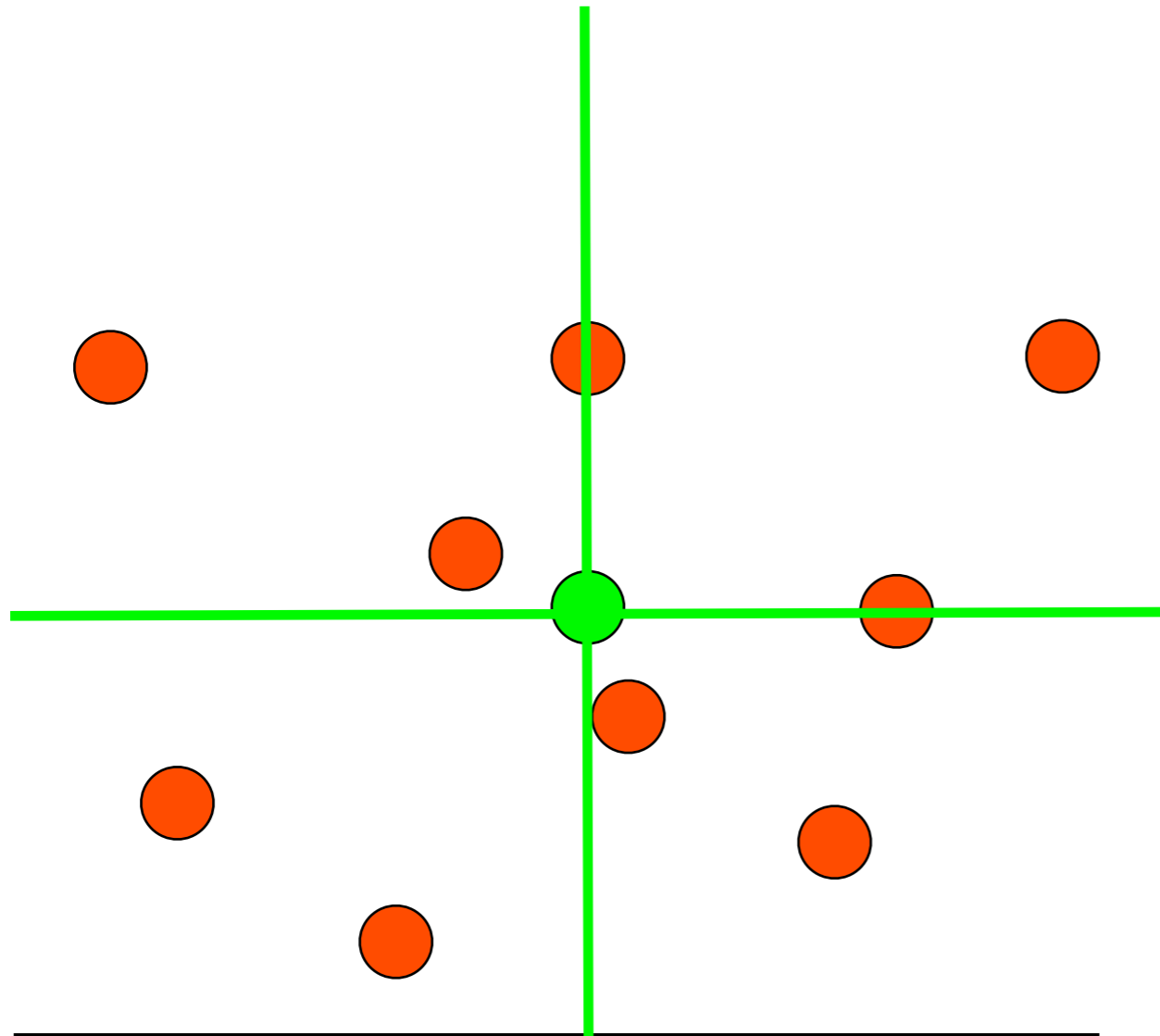
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.

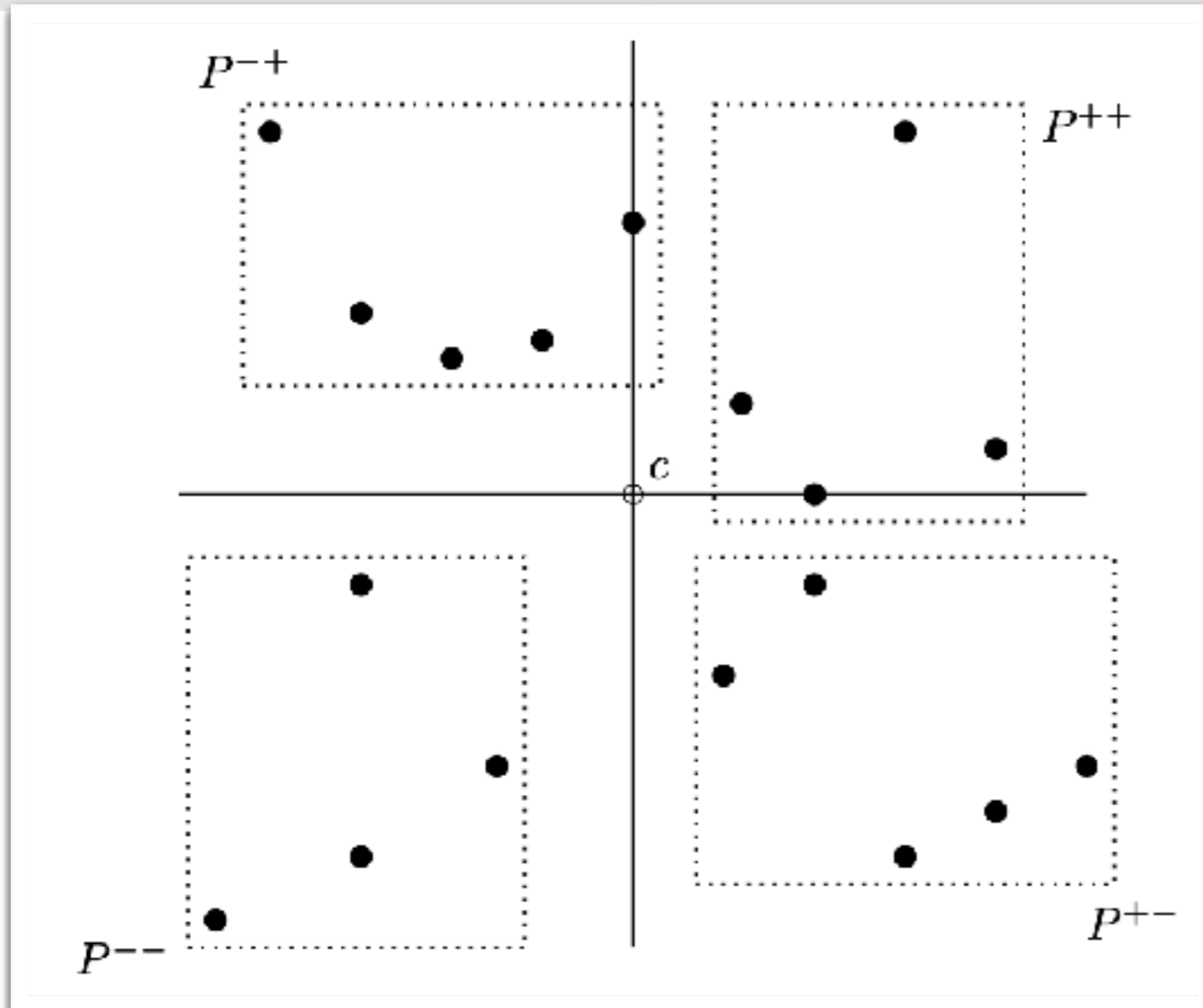
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.

# Longest Tours in Manhattan

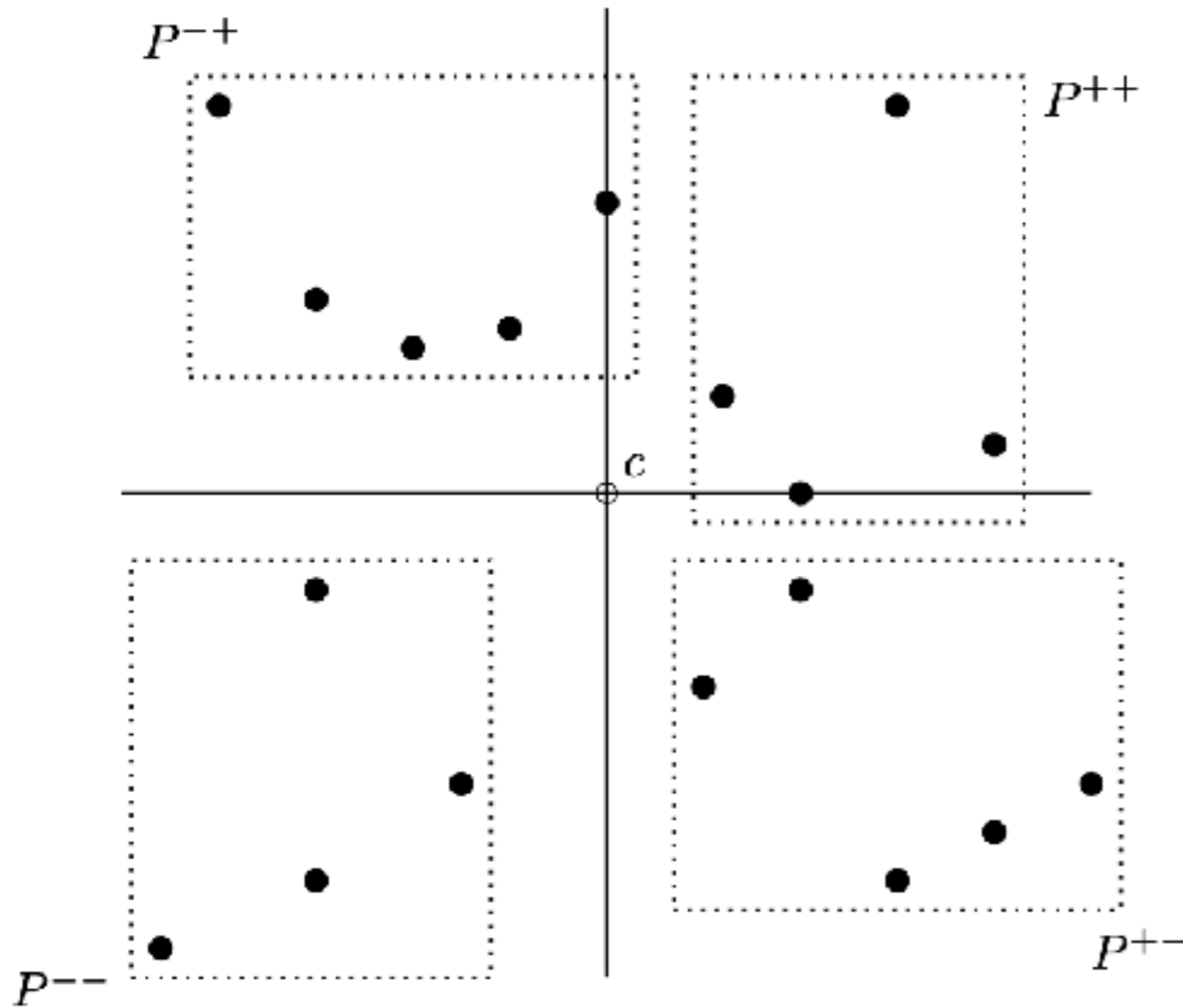


$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.

# Longest Tours in Manhattan

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.



$$n^{--} + n^{-+} = n^{+-} + n^{++}$$

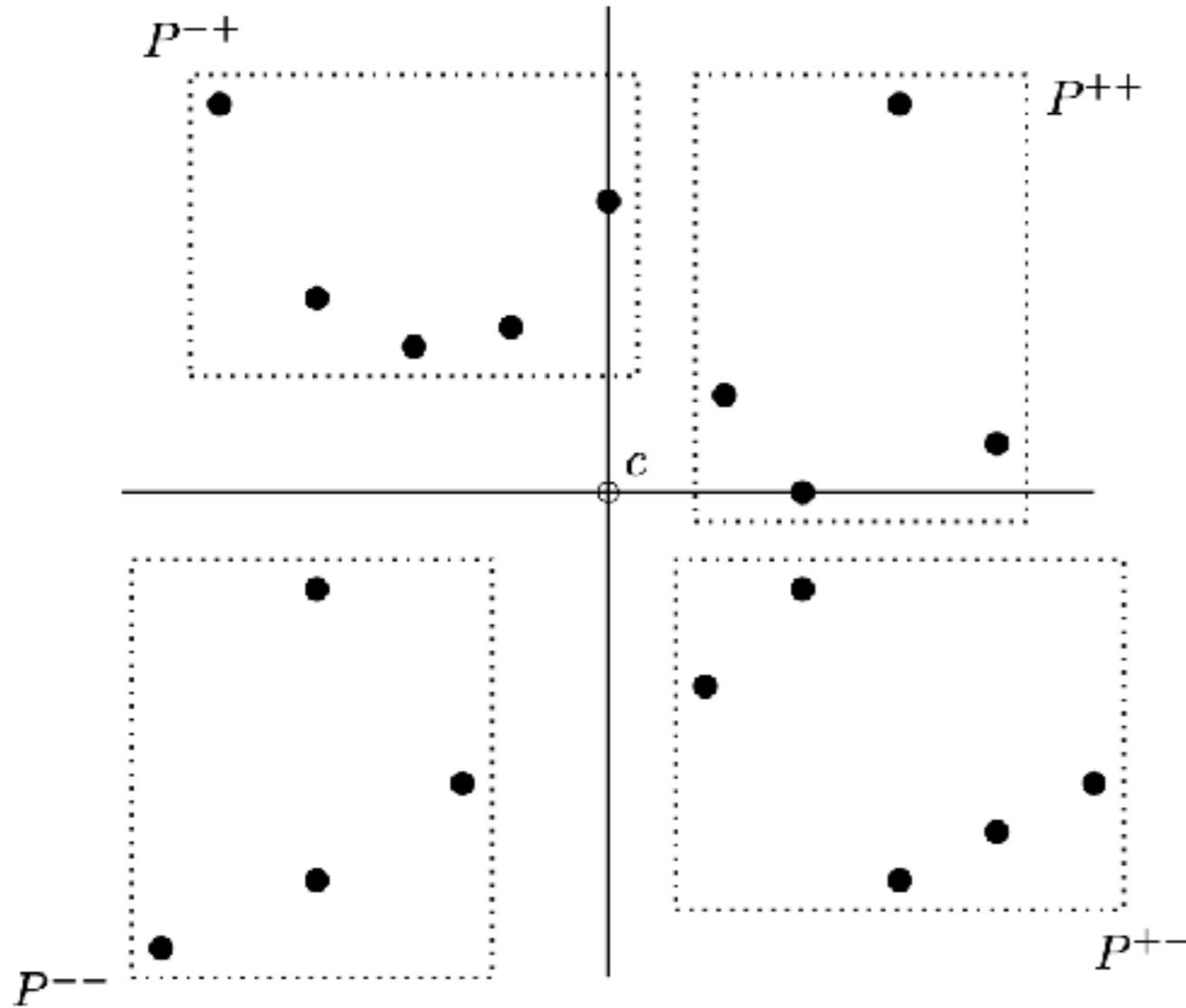
$$n^{--} + n^{+-} = n^{-+} + n^{++}$$

$$\leq 2 \min Star$$



# Longest Tours in Manhattan

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.



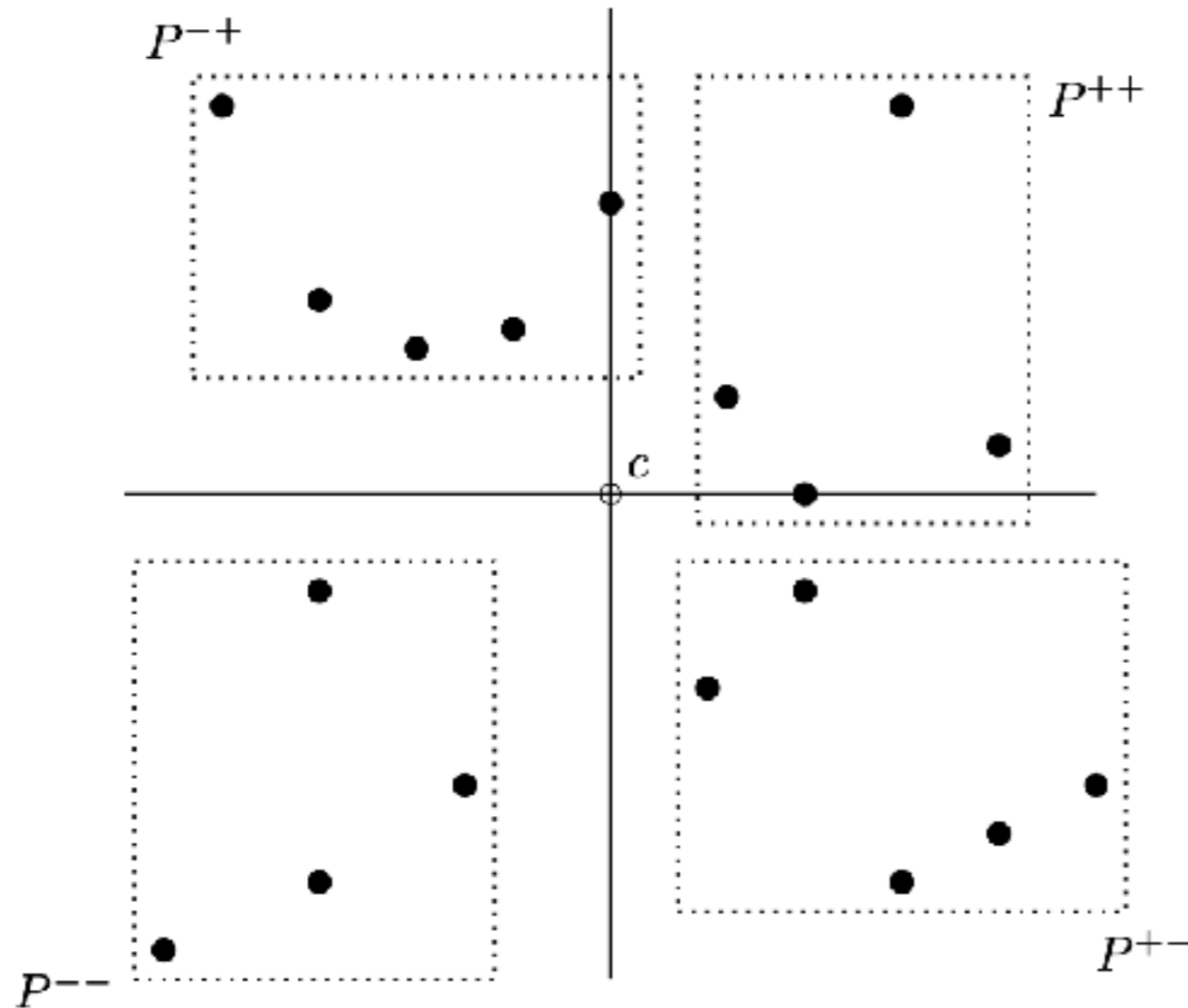
$$n^{--} + n^{-+} = n^{+-} + n^{++}$$

$$n^{--} + n^{+-} = n^{-+} + n^{++}$$

$$n^{-+} - n^{+-} = n^{+-} - n^{-+}$$

# Longest Tours in Manhattan

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.



$$n^{--} + n^{-+} = n^{+-} + n^{++}$$

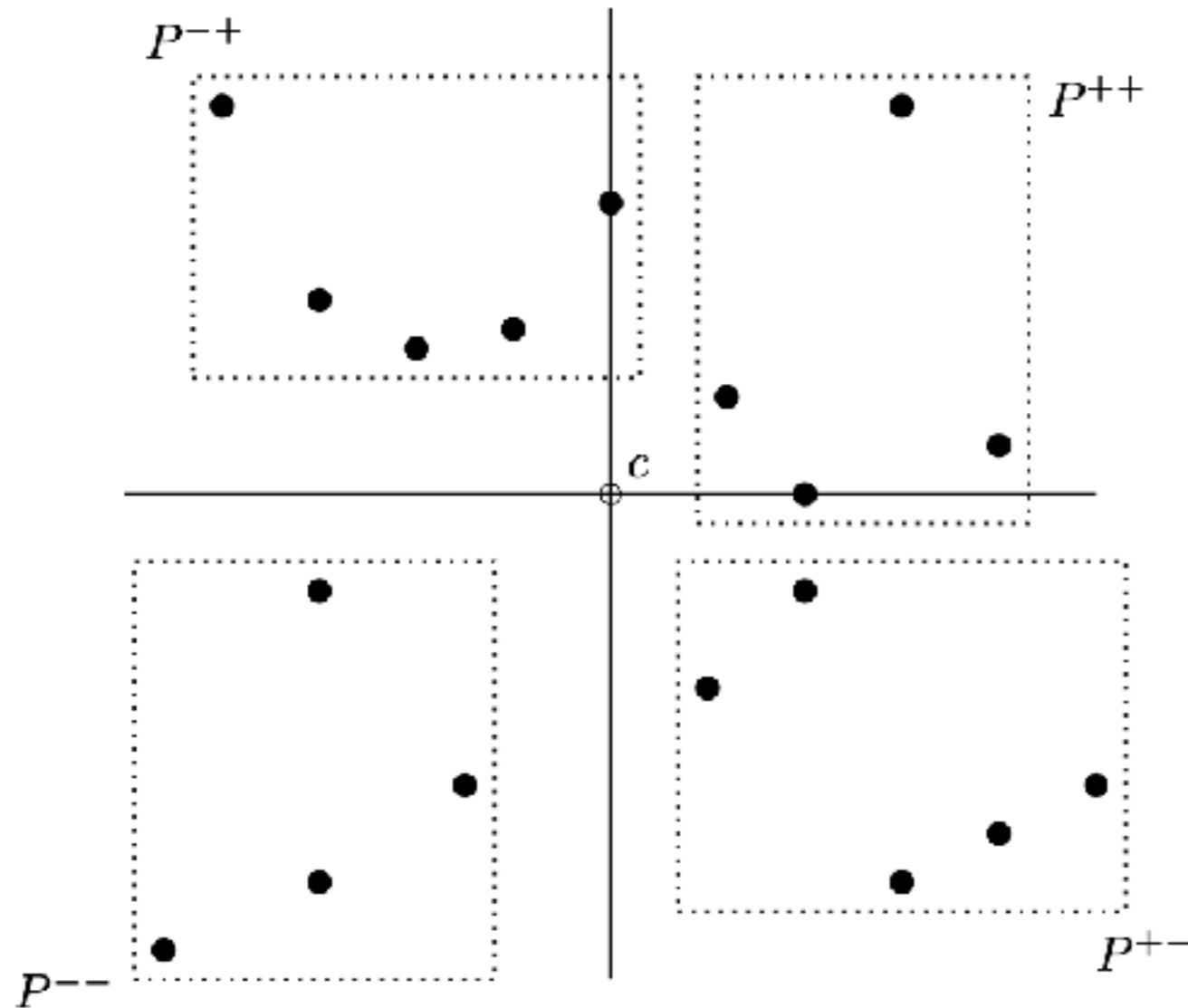
$$n^{--} + n^{+-} = n^{-+} + n^{++}$$

$$n^{-+} - n^{+-} = n^{+-} - n^{-+}$$

$$n^{-+} = n^{+-}$$

# Longest Tours in Manhattan

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.



$$n^{--} + n^{-+} = n^{+-} + n^{++}$$

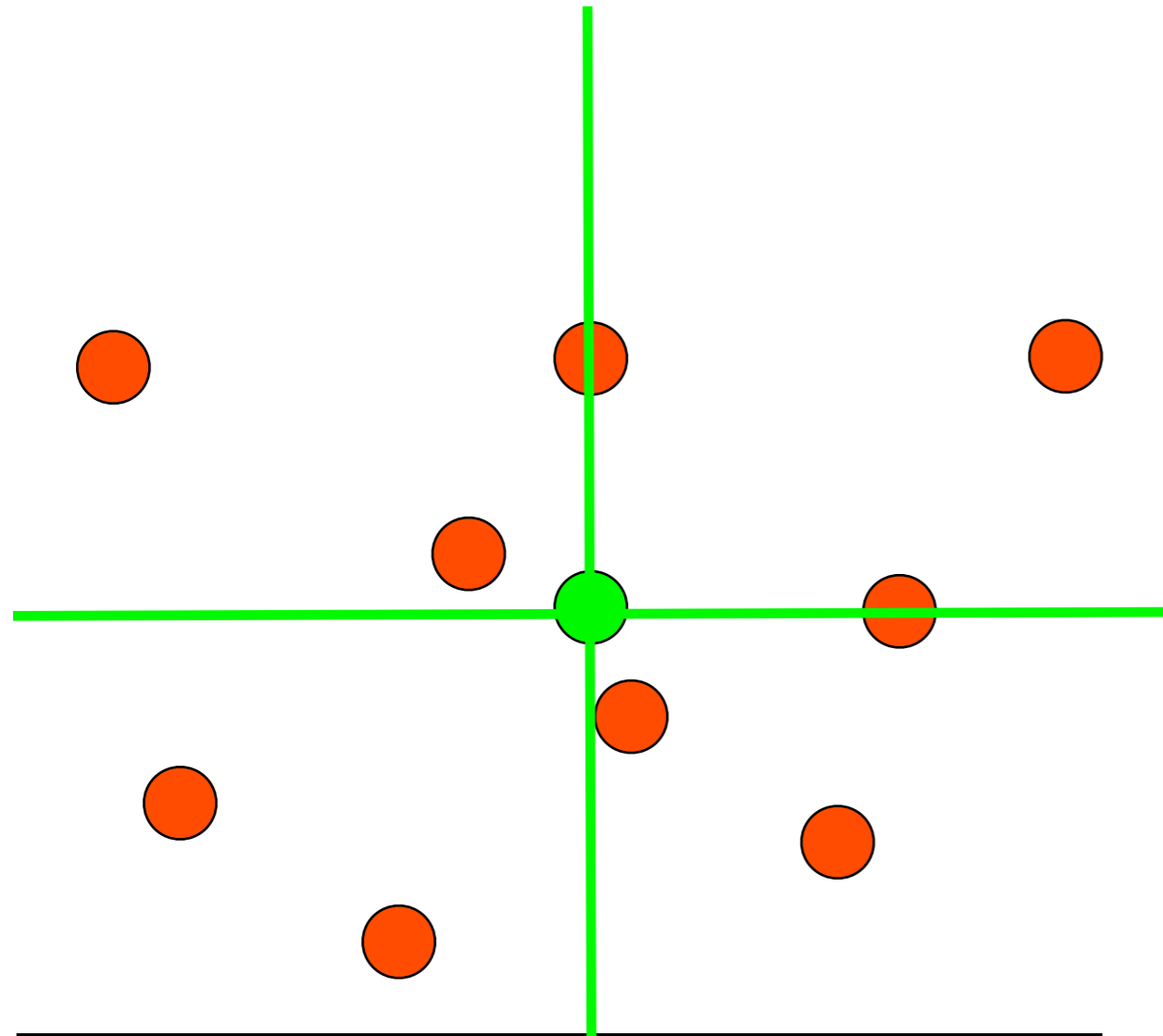
$$n^{--} + n^{+-} = n^{-+} + n^{++}$$

$$n^{-+} - n^{+-} = n^{+-} - n^{-+}$$

$$n^{-+} = n^{+-}$$

$$n^{--} = n^{++}$$

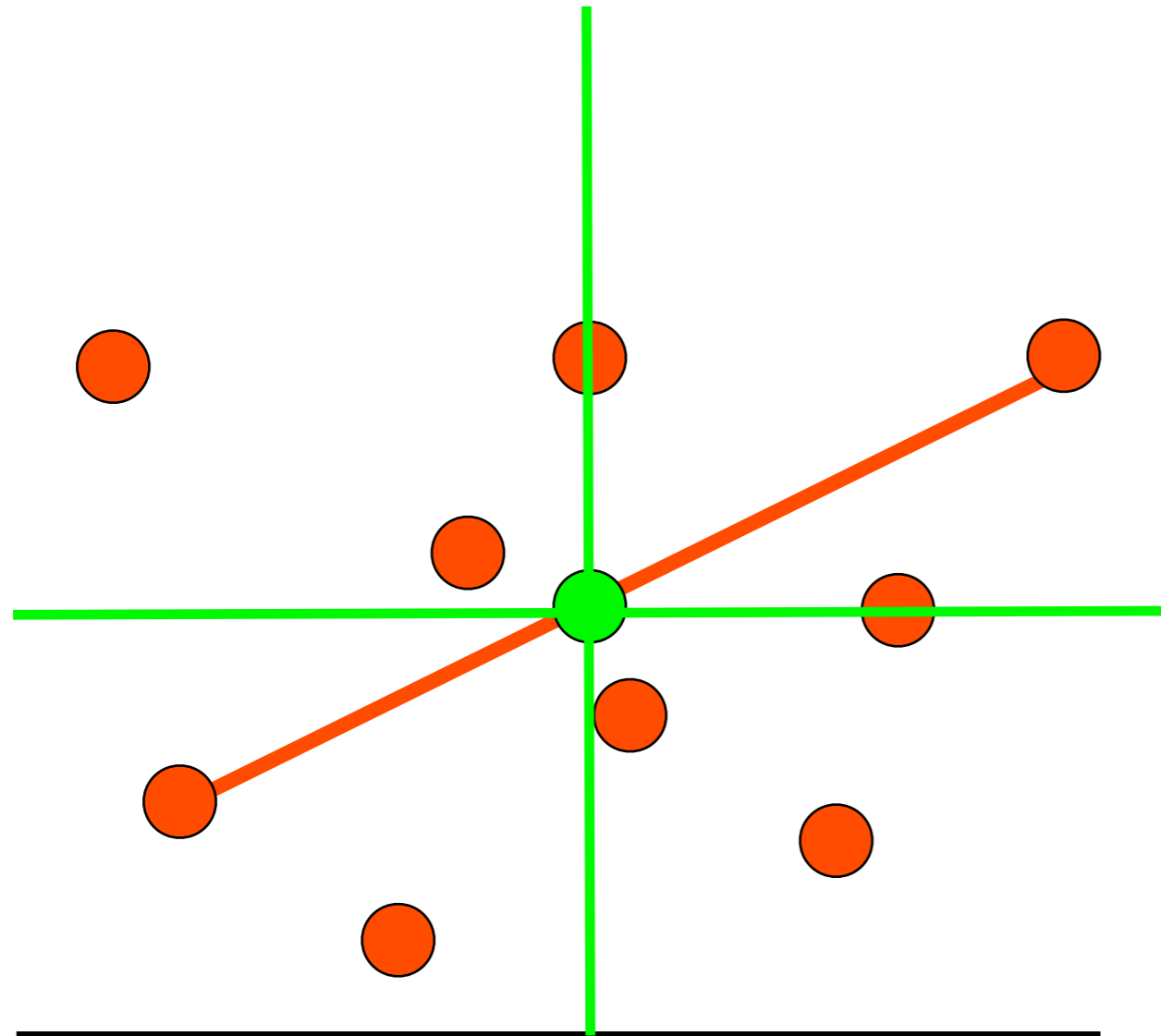
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.

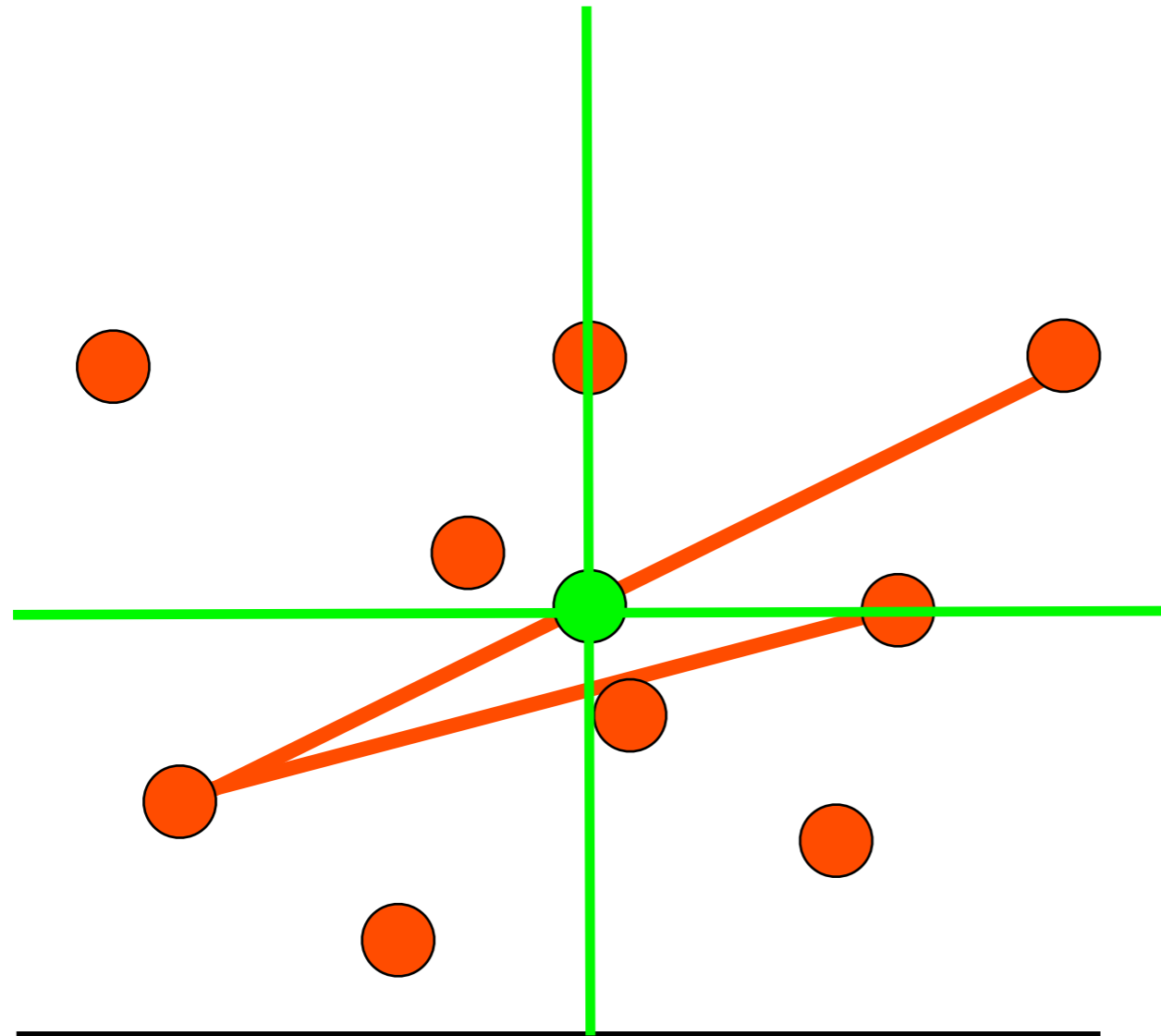
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.

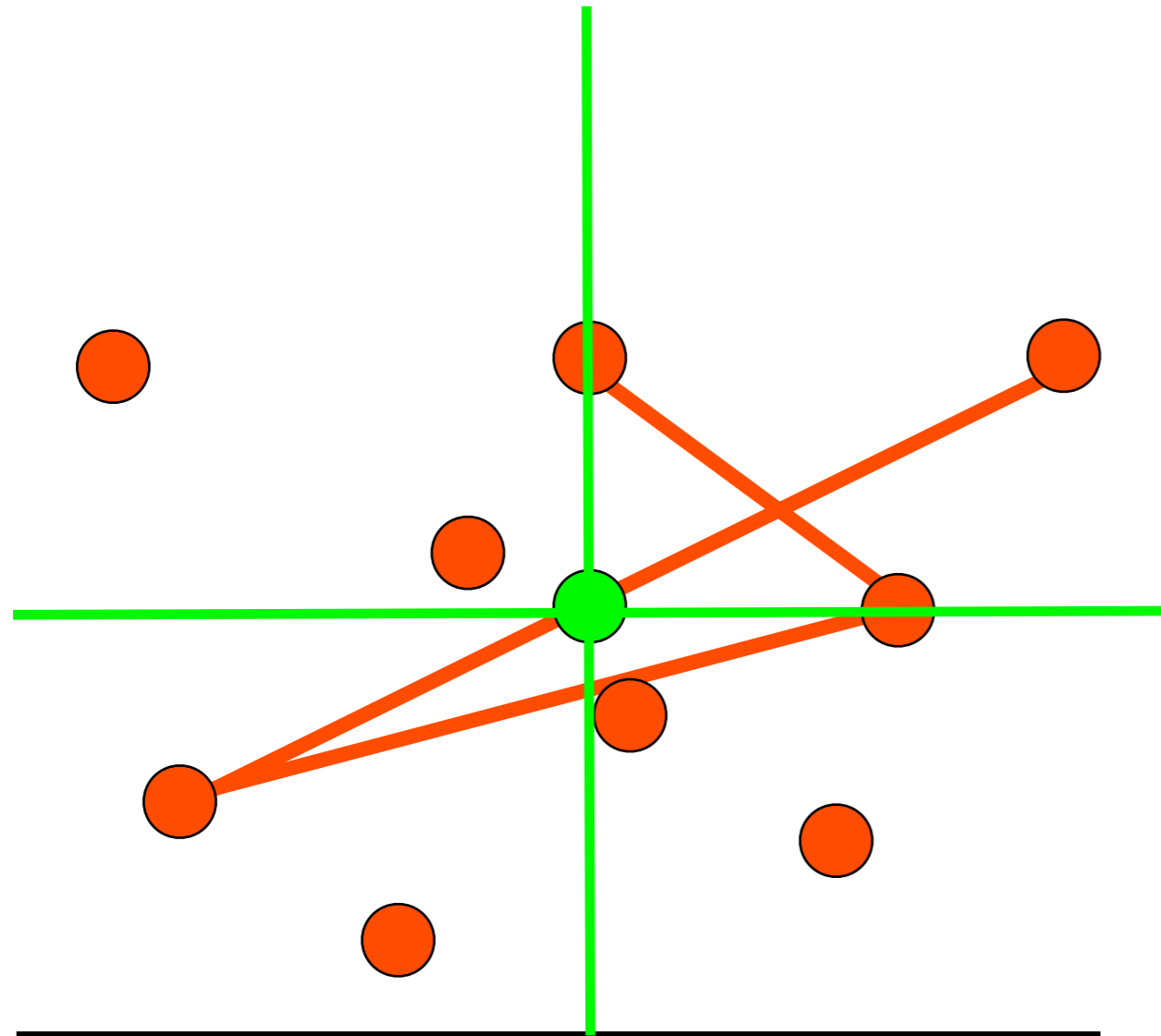
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.

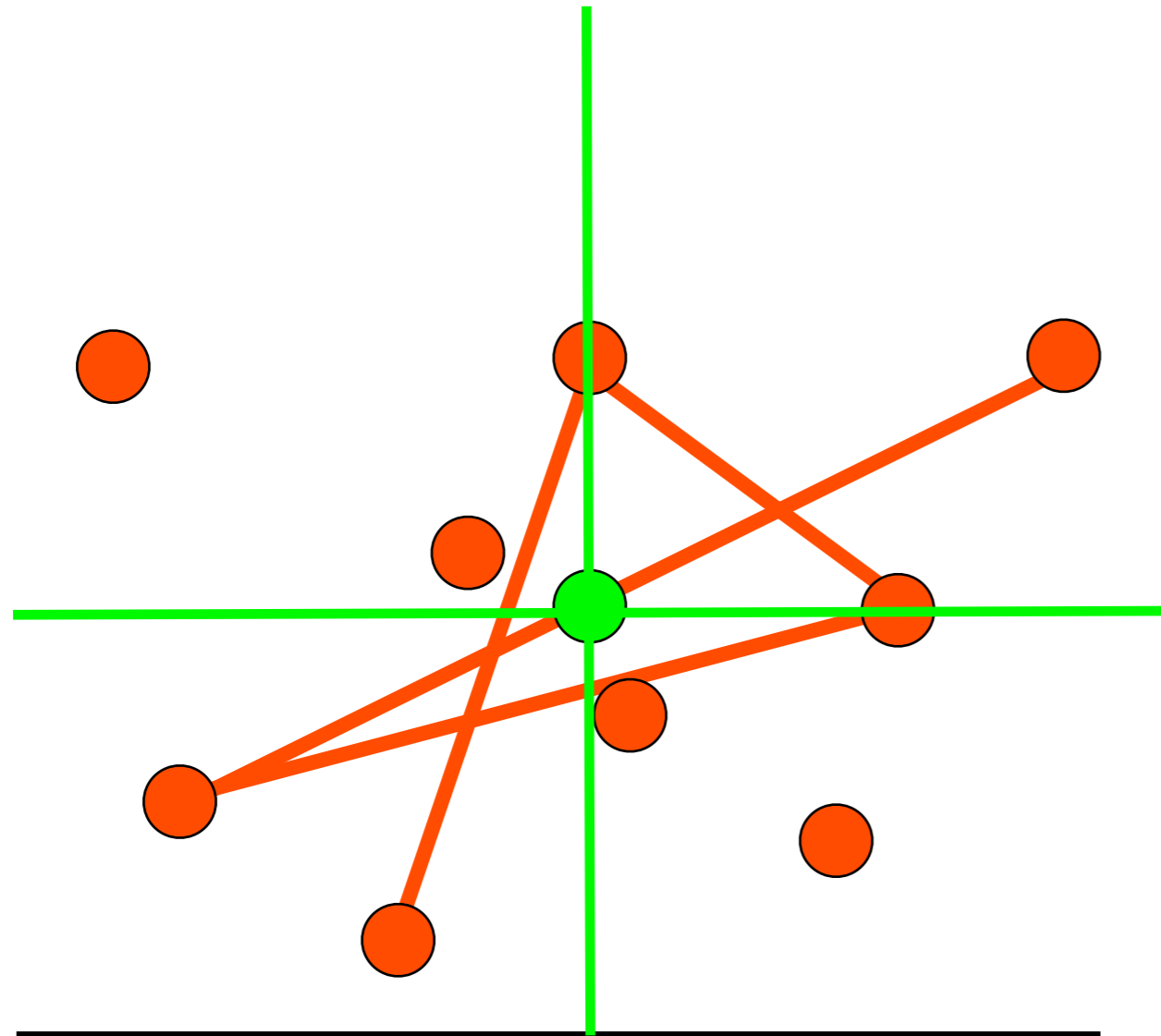
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.

# Longest Tours in Manhattan

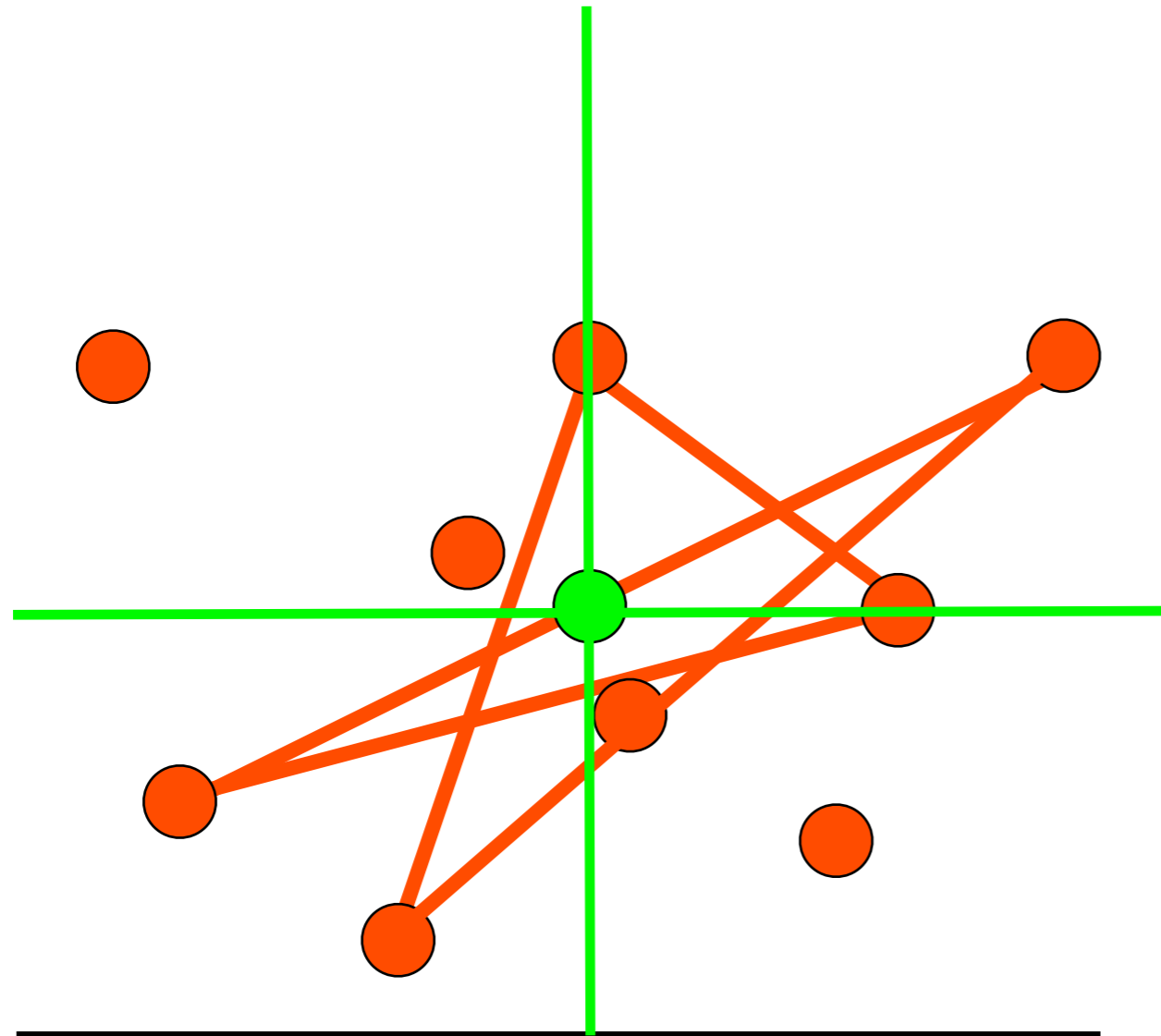


$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances:  
median in x and y!
- Find long connections.
- Find long subtours.



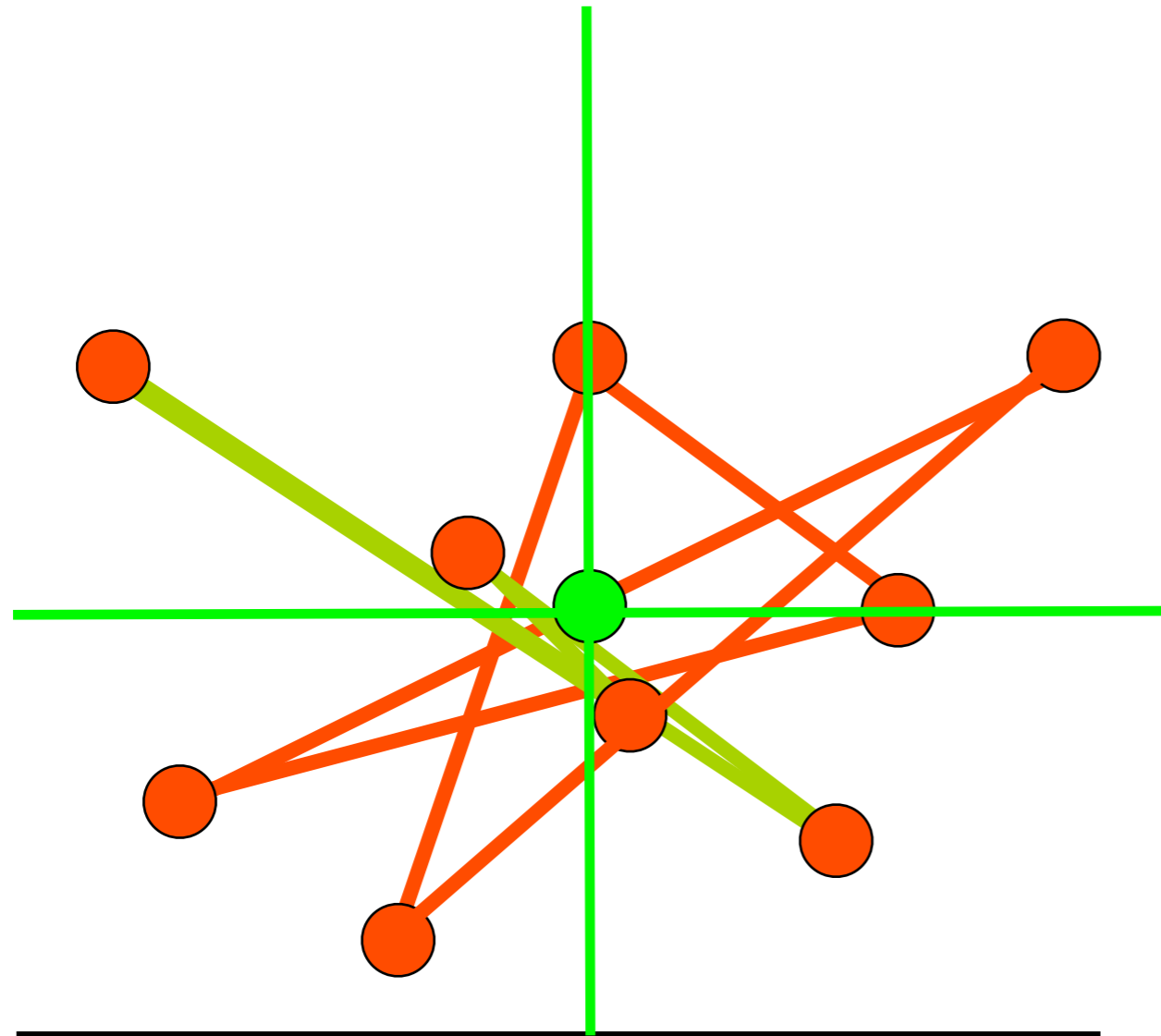
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances:  
median in x and y!
- Find long connections.
- Find long subtours.

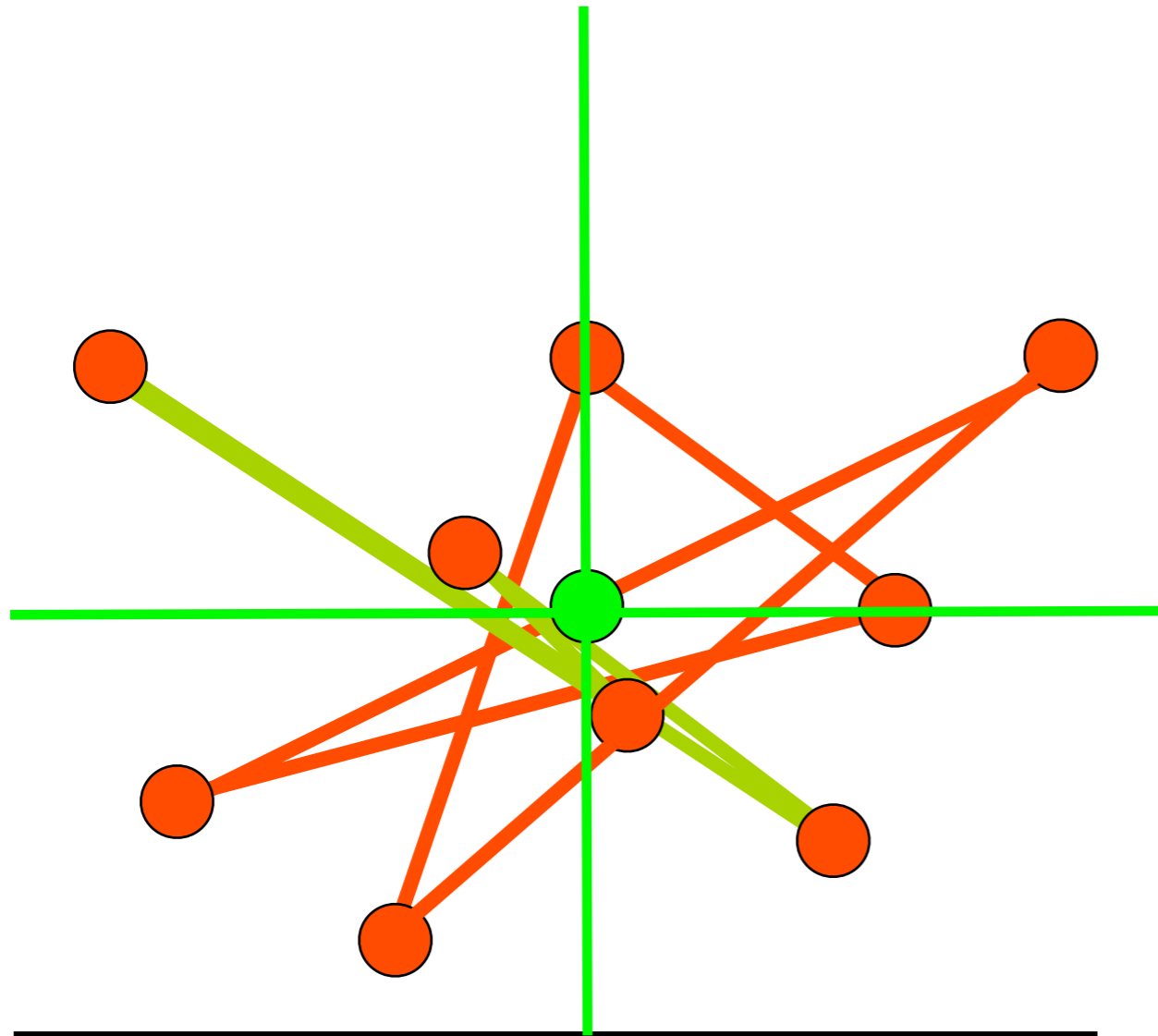
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.

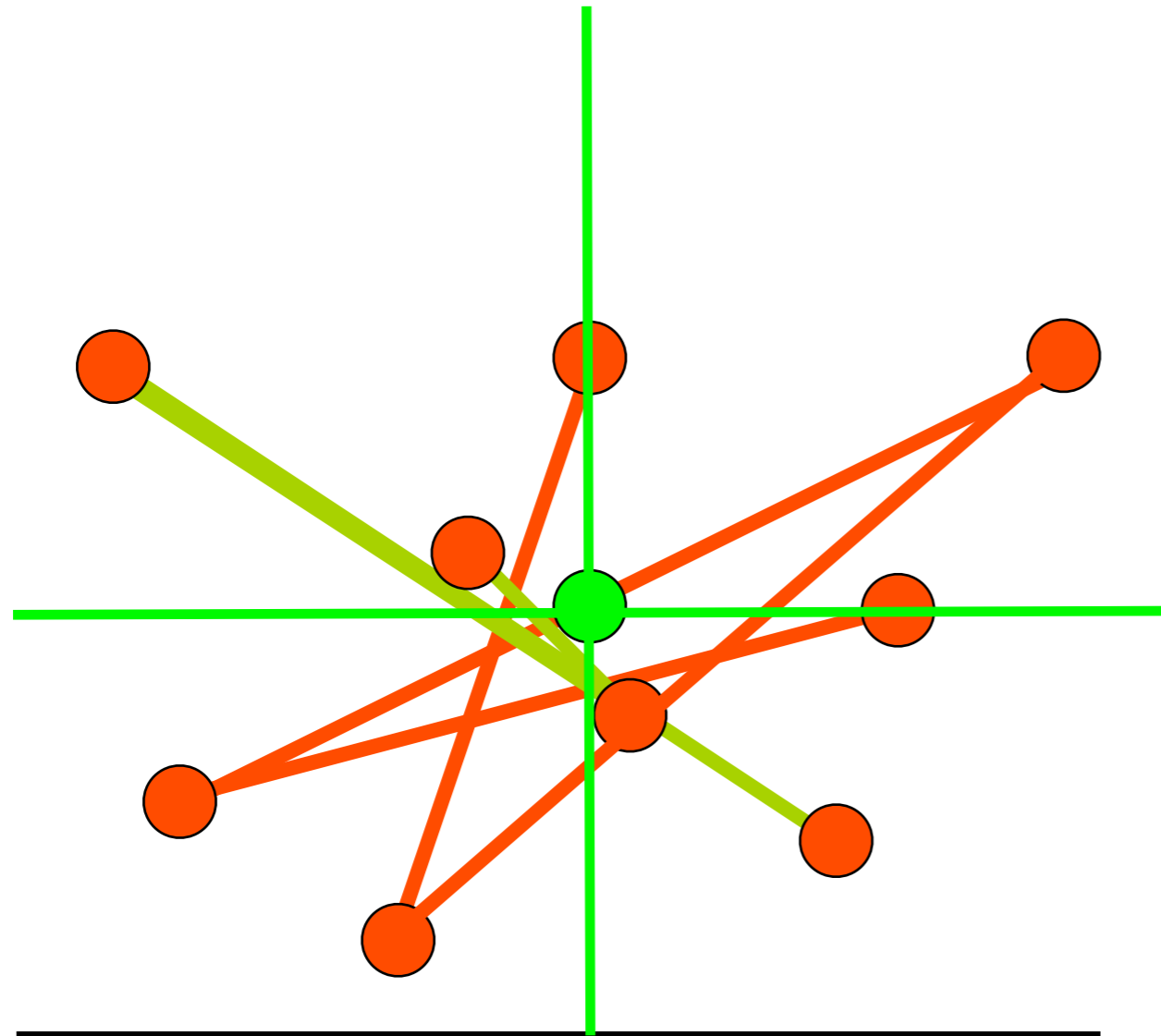
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.
- Find longest tour.

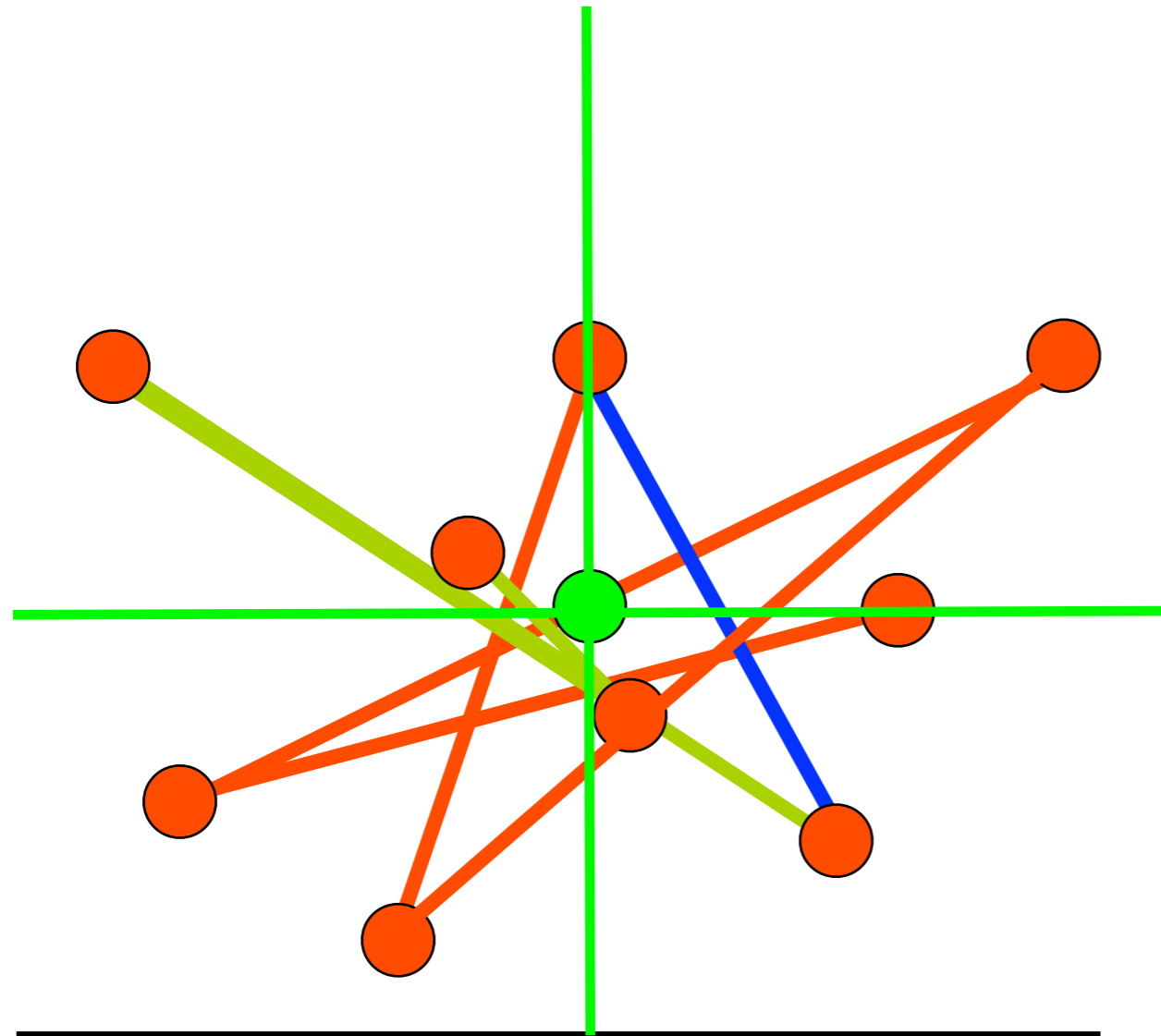
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.
- Find longest tour.

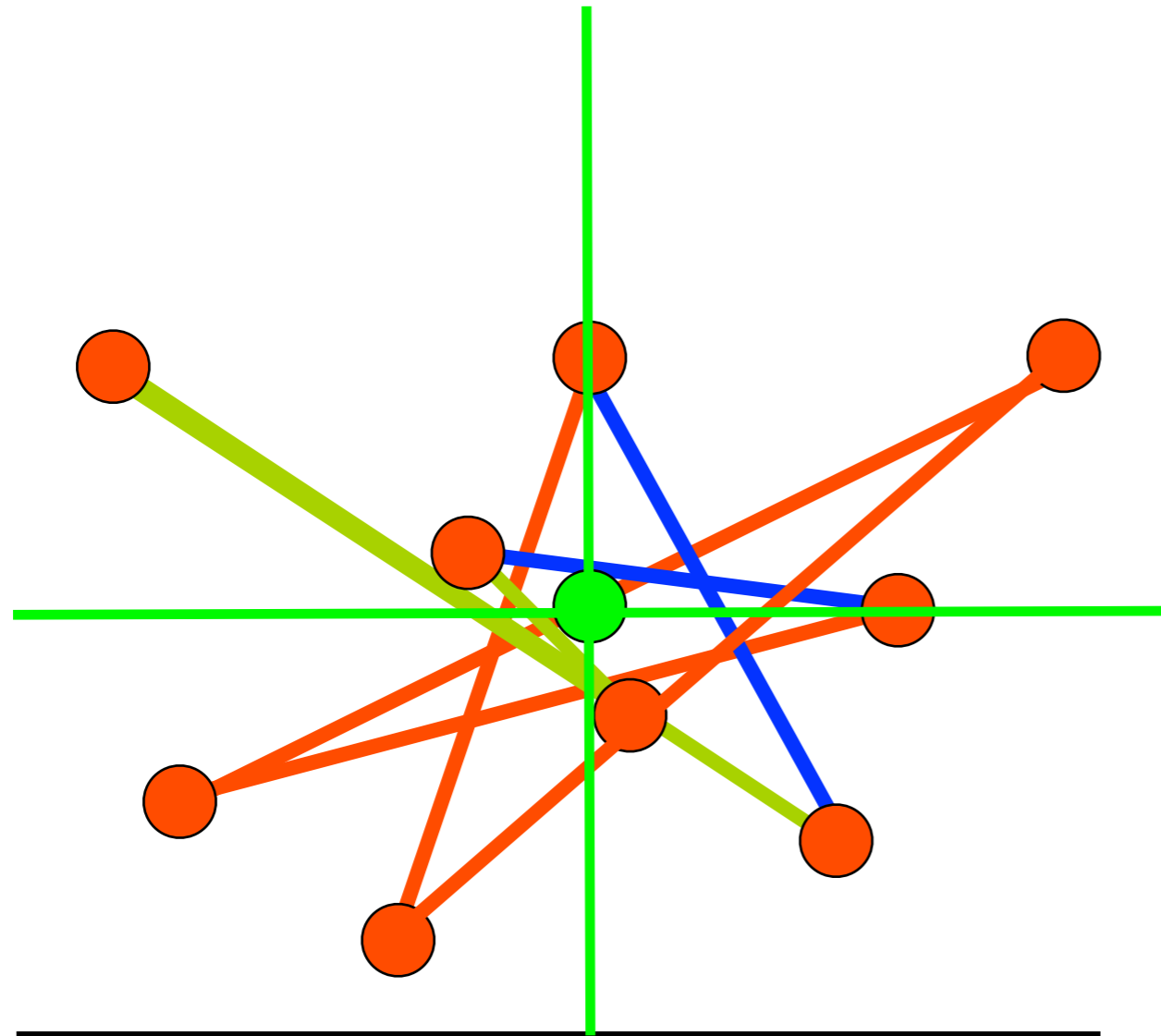
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.
- Find longest tour.

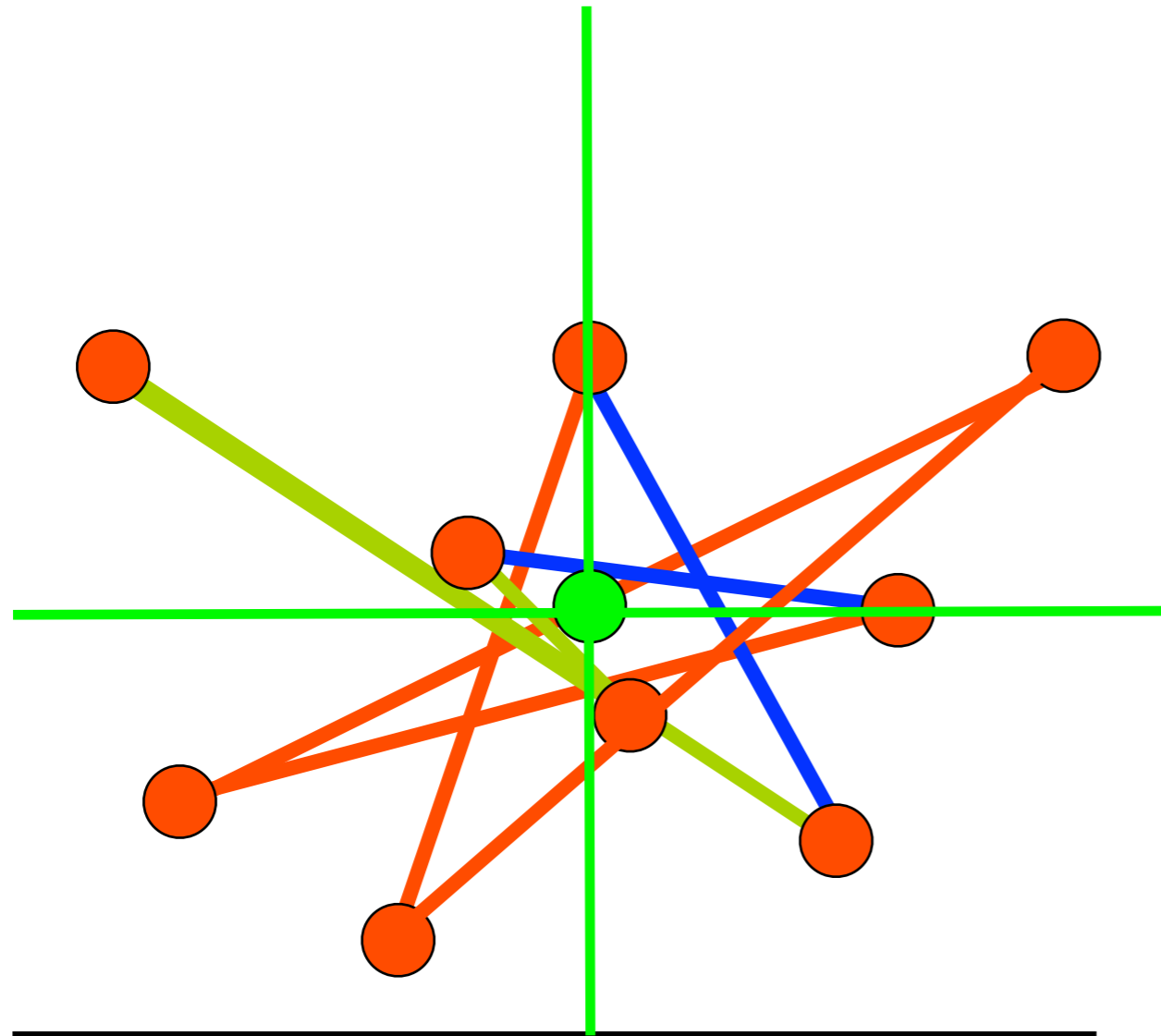
# Longest Tours in Manhattan



$$\max Tour \leq 2 \min Star$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.
- Find longest tour.

# Longest Tours in Manhattan

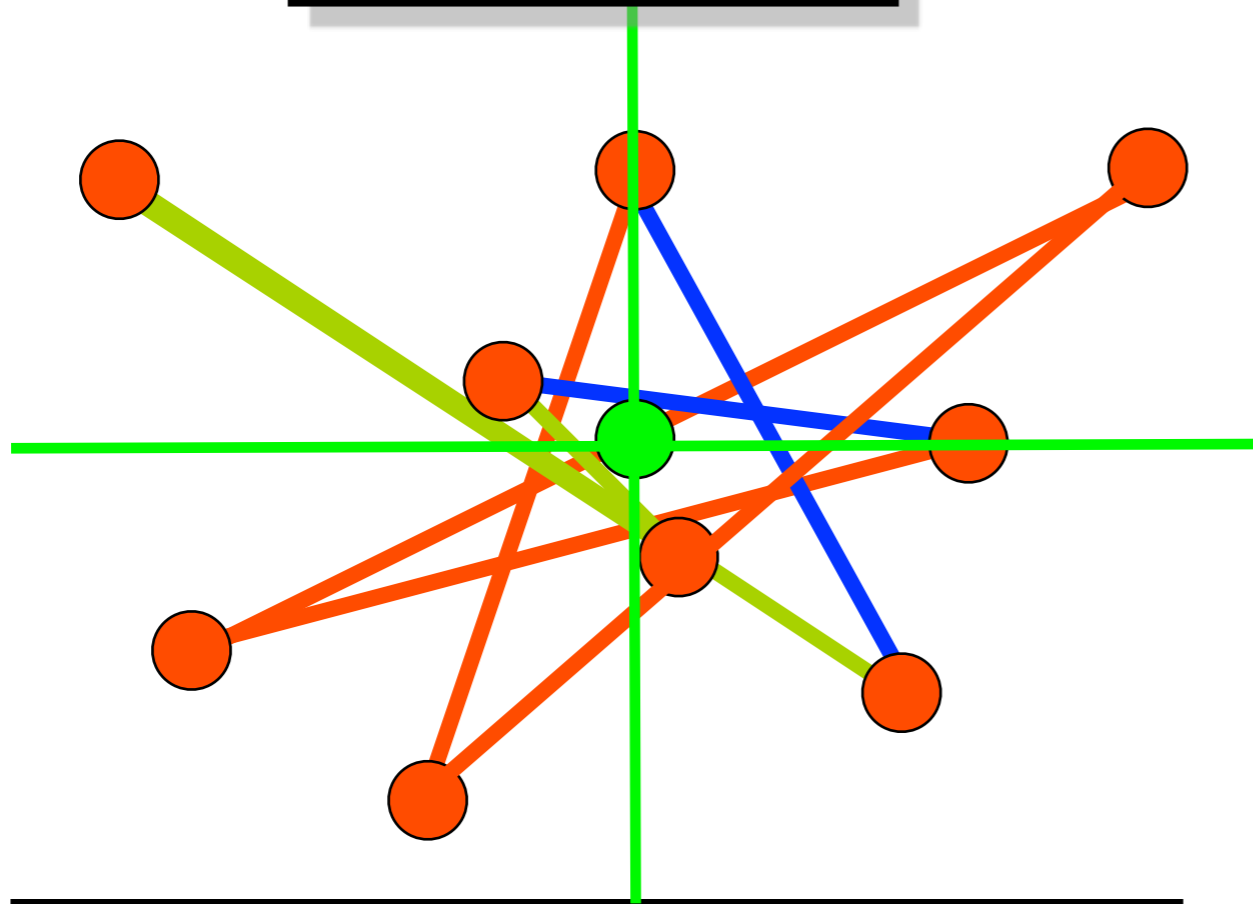


$$\max \text{Tour} = 2 \min \text{Star}$$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.
- Find longest tour.

# Longest Tours in Manhattan

Optimal!!



$\max \text{Tour} = 2 \min \text{Star}$

- Find a point that minimizes the sum of Manhattan distances: median in x and y!
- Find long connections.
- Find long subtours.
- Find longest tour.



# Longest Tours in Manhattan

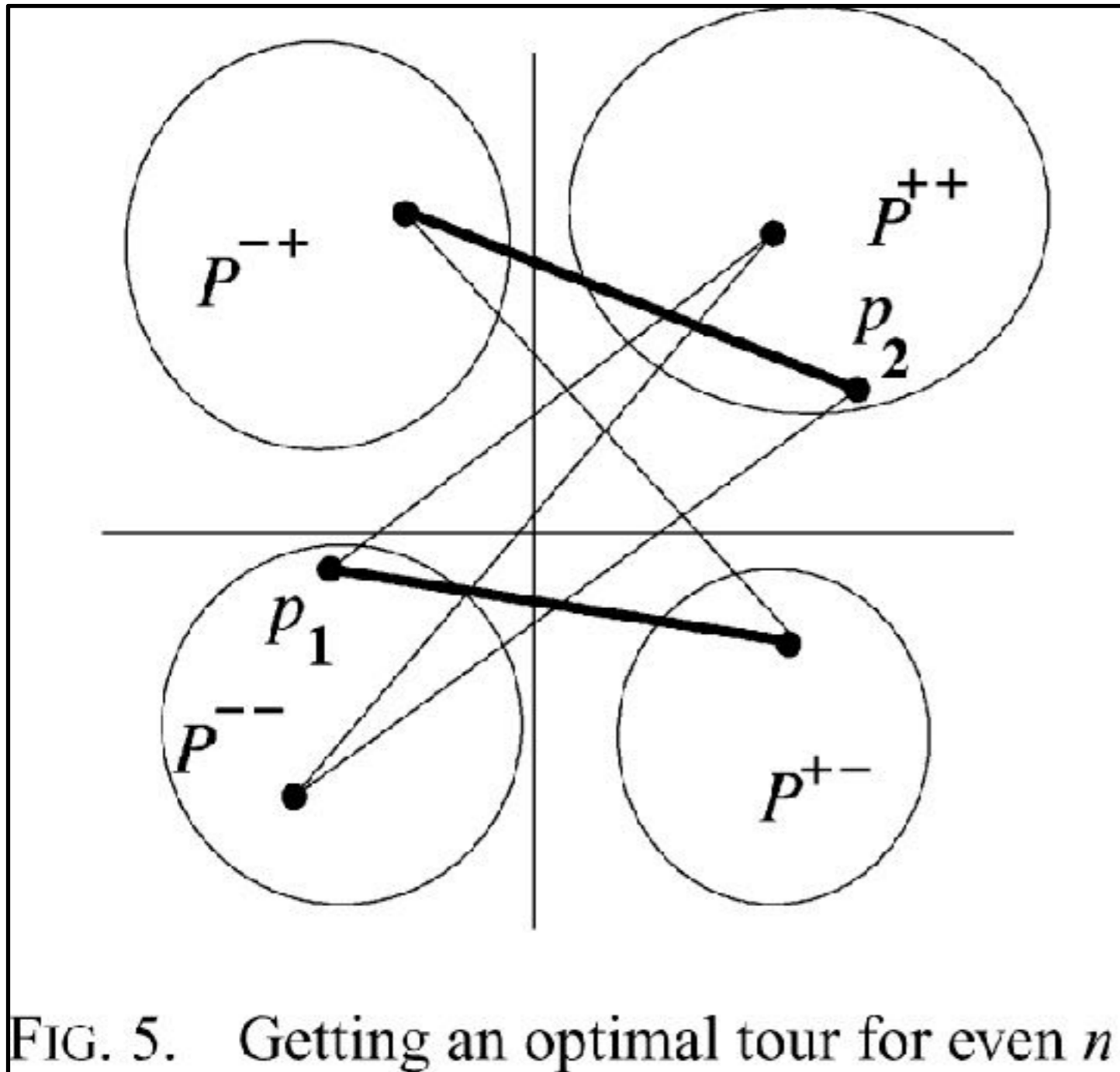


FIG. 5. Getting an optimal tour for even  $n$ .

- Find a point that minimizes the sum of Manhattan distances: median in  $x$  and  $y$ !
- Find long connections.
- Find long subtours.
- Find longest tour.

# Longest Tours in Manhattan

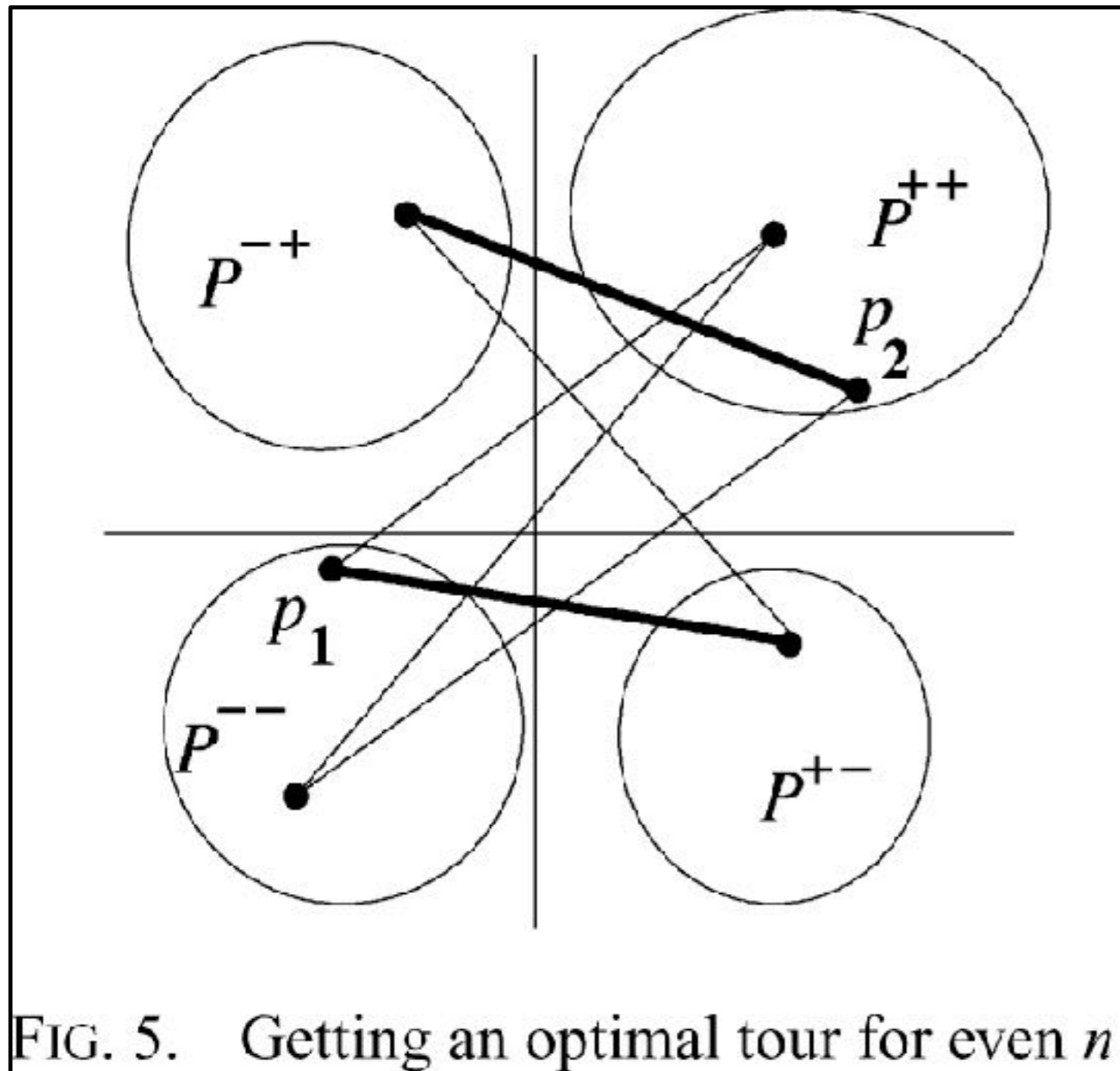
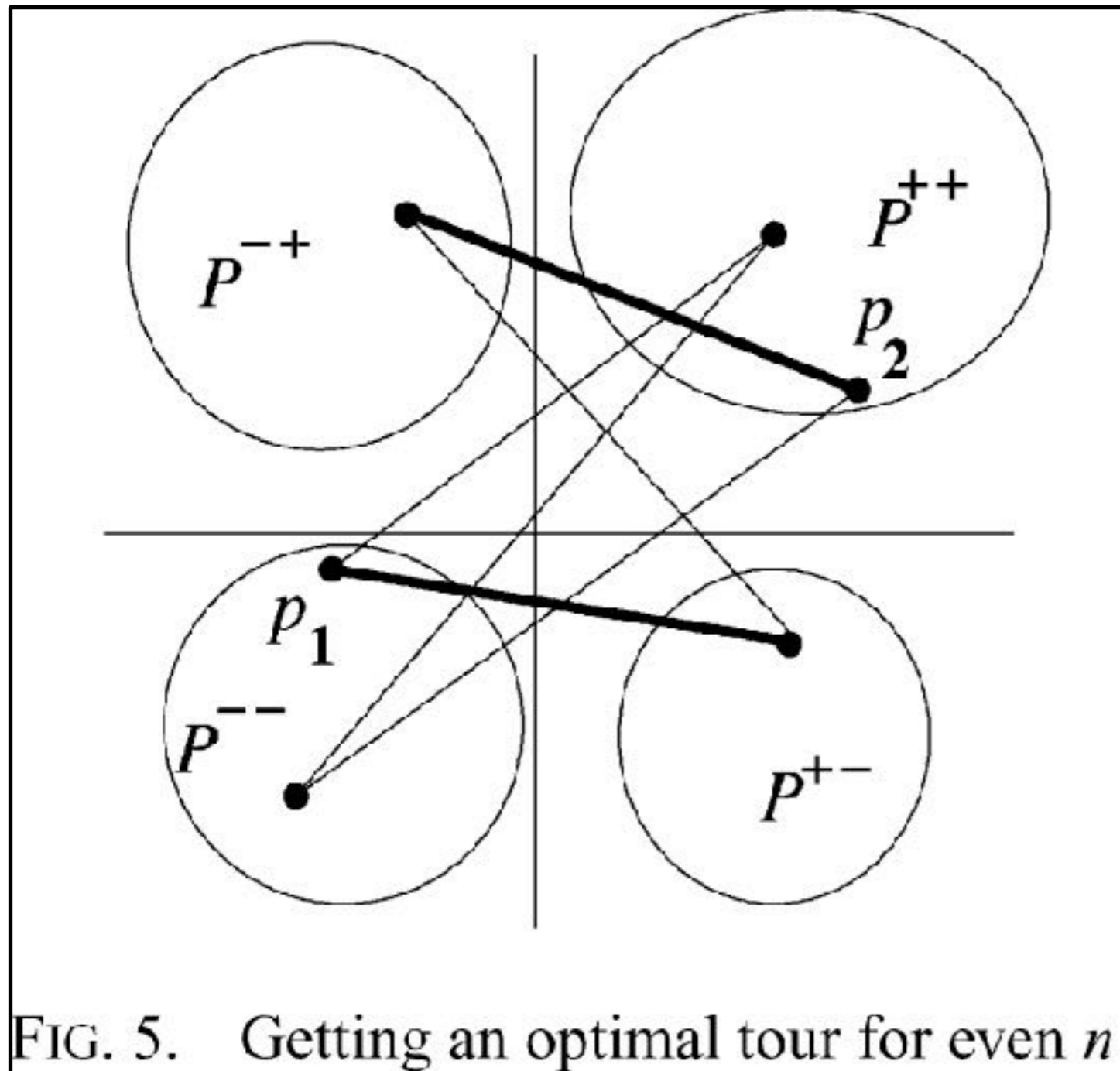


FIG. 5. Getting an optimal tour for even  $n$ .

- Find a point that minimizes the sum of Manhattan distances: median in  $x$  and  $y$ !
- Find long connections.
- Find long subtours.
- Find longest tour.

**Theorem:**

# Longest Tours in Manhattan

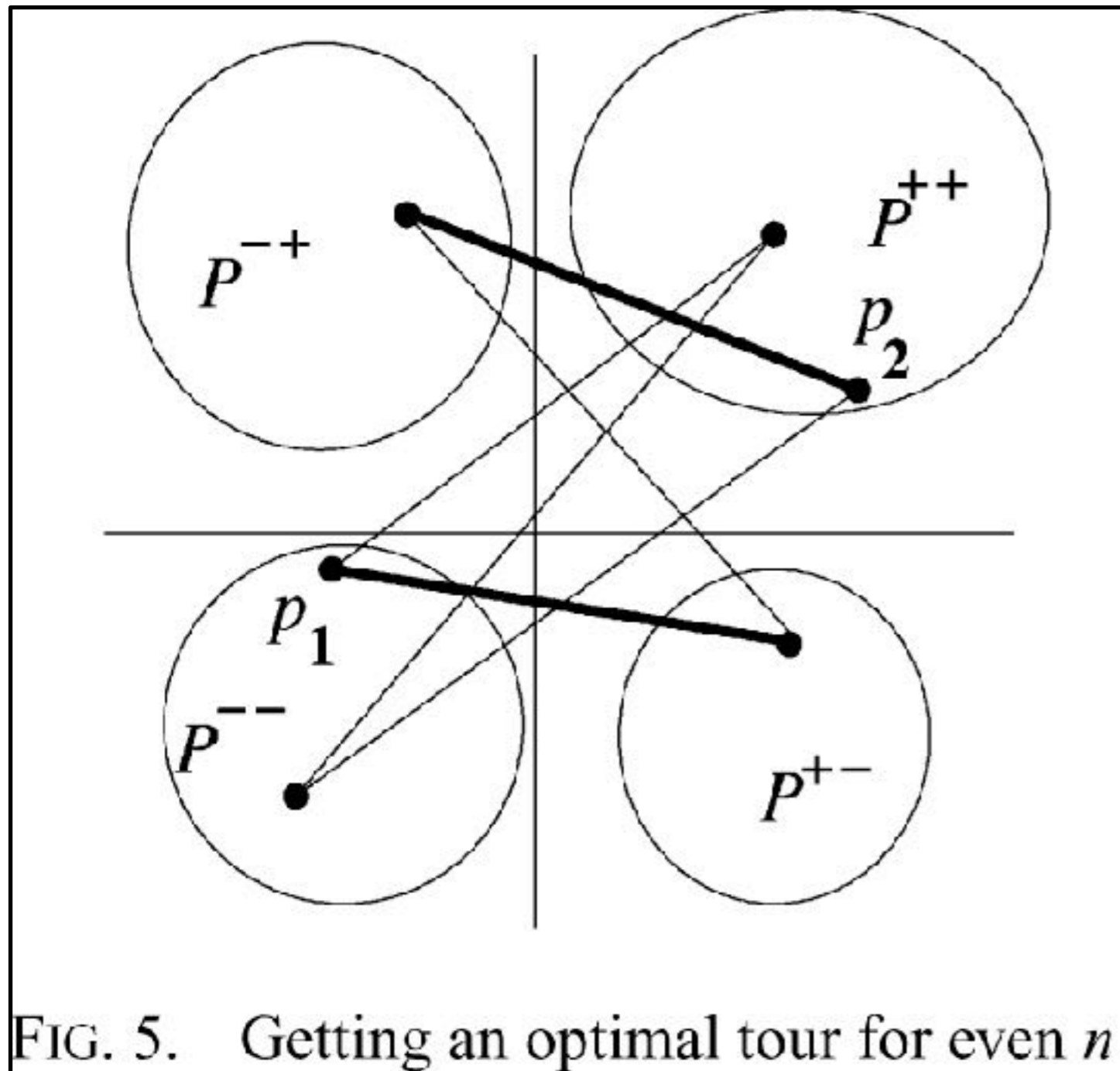


- Find a point that minimizes the sum of Manhattan distances: median in  $x$  and  $y$ !
- Find long connections.
- Find long subtours.
- Find longest tour.

## Theorem:

A longest tour for points in the plane with Manhattan distances can be computed in *linear* time.

# Longest Tours in Manhattan



- Find a point that minimizes the sum of Manhattan distances: median in  $x$  and  $y$ !
- Find long connections.
- Find long subtours.
- Find longest tour.

## Theorem:

A longest tour for points in the plane with Manhattan distances can be computed in *linear* time.

**Euclidean distances??**

# Longest Tours in Manhattan

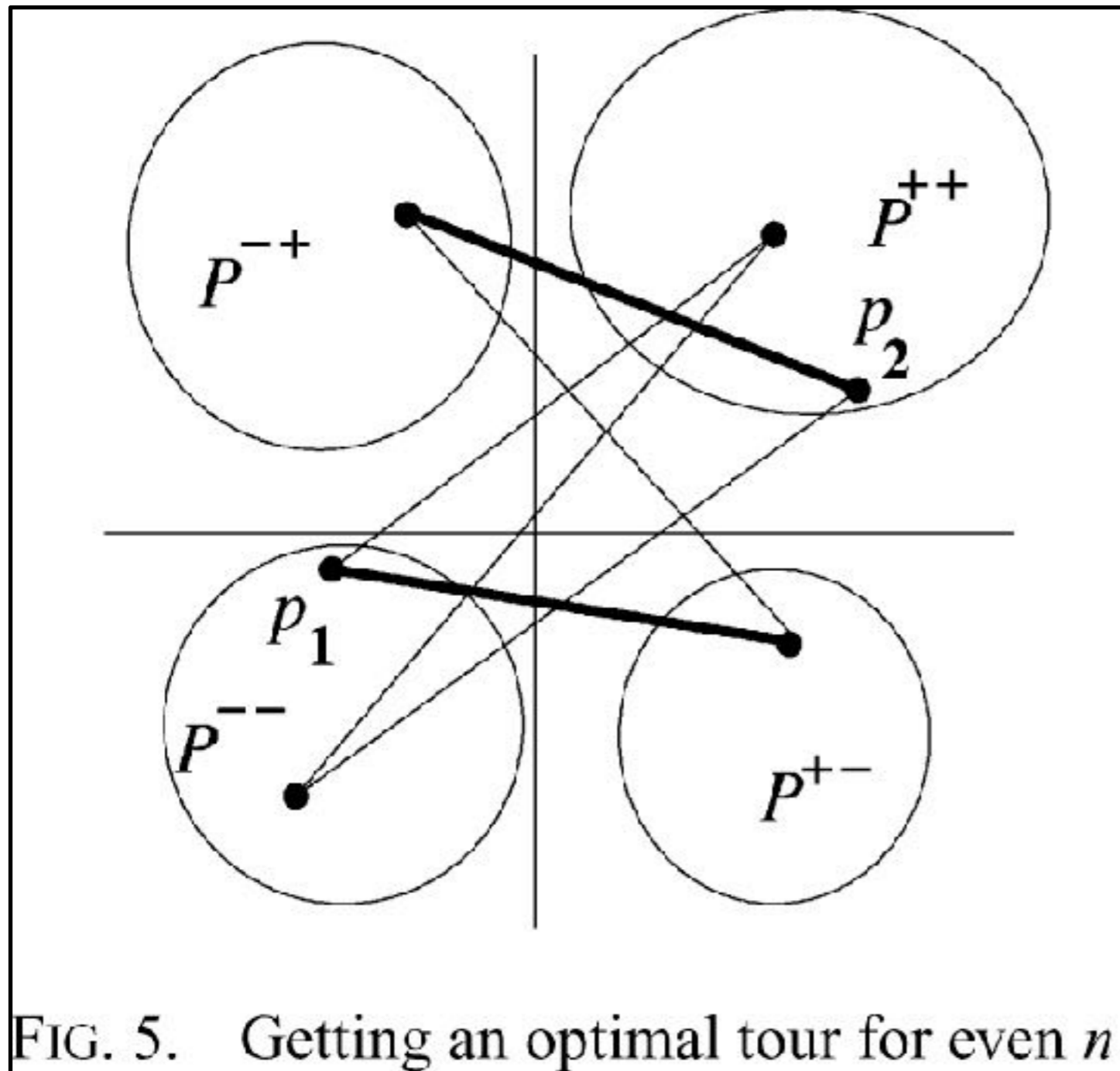


FIG. 5. Getting an optimal tour for even  $n$ .

- Find a point that minimizes the sum of Manhattan distances: median in  $x$  and  $y$ !
- Find long connections.
- Find long subtours.
- Find longest tour.

## Theorem:

A longest tour for points in the plane with Manhattan distances can be computed in *linear* time.

**Euclidean distances??**

# Think Globally

# Think Globally



SEOUL	9.636 Km	TAIPEI	9.329 Km
LONDON	19.271 Km	LOS ANGELES	10.479 Km
SYDNEY	21.59 Km	NEW YORK	16.334 Km
TOKYO	8.831 Km	FRANKFURT	19.314 Km
SINGAPORE	8.404 Km	HAWAII	7.086 Km
HONG KONG	9.144 Km	TAHITI	4.091 Km
FIJI	2.157 Km	BUENOS AIRES	10.327 Km

# Think Globally



SEOUL	9.636 Km	TAIPEI	9.329 Km
LONDON	19.271 Km	LOS ANGELES	10.479 Km
SYDNEY	21.59 Km	NEW YORK	16.334 Km
TOKYO	8.831 Km	FRANKFURT	19.314 Km
SINGAPORE	8.404 Km	HAWAII	7.086 Km
HONG KONG	9.144 Km	TAHITI	4.091 Km
FIJI	2.157 Km	BUENOS AIRES	10.327 Km



# Think Globally

SEOUL	9.636 Km	TAIPEI	9.329 Km
LONDON	19.271 Km	LOS ANGELES	10.479 Km
SYDNEY	21.59 Km	NEW YORK	15.334 Km
TOKYO	8.831 Km	FRANKFURT	19.314 Km
SINGAPORE	8.404 Km	HAWAII	7.086 Km
HONG KONG	9.144 Km	TAHITI	4.091 Km
FIJI	2.157 Km	BUENOS AIRES	10.327 Km

# Think Globally



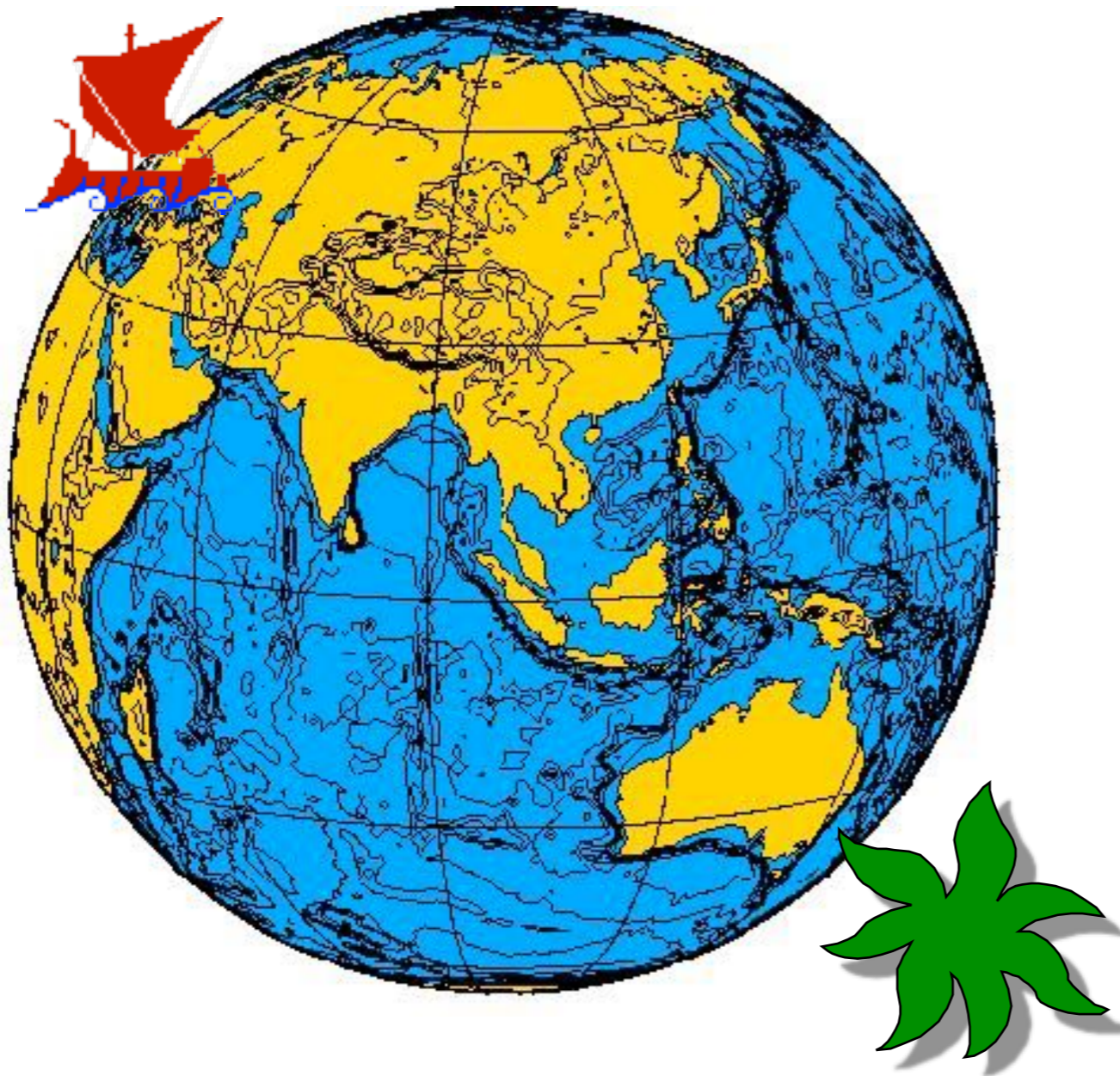
SEOUL	9.636 Km	TAIPEI	9.329 Km
LONDON	19.271 Km	LOS ANGELES	10.479 Km
SYDNEY	21.59 Km	NEW YORK	16.334 Km
TOKYO	8.831 Km	FRANKFURT	19.314 Km
SINGAPORE	8.404 Km	HAWAII	7.086 Km
HONG KONG	9.144 Km	TAHITI	4.091 Km
FIJI	2.157 Km	BUENOS AIRES	10.327 Km

# Think Globally

# Think Globally

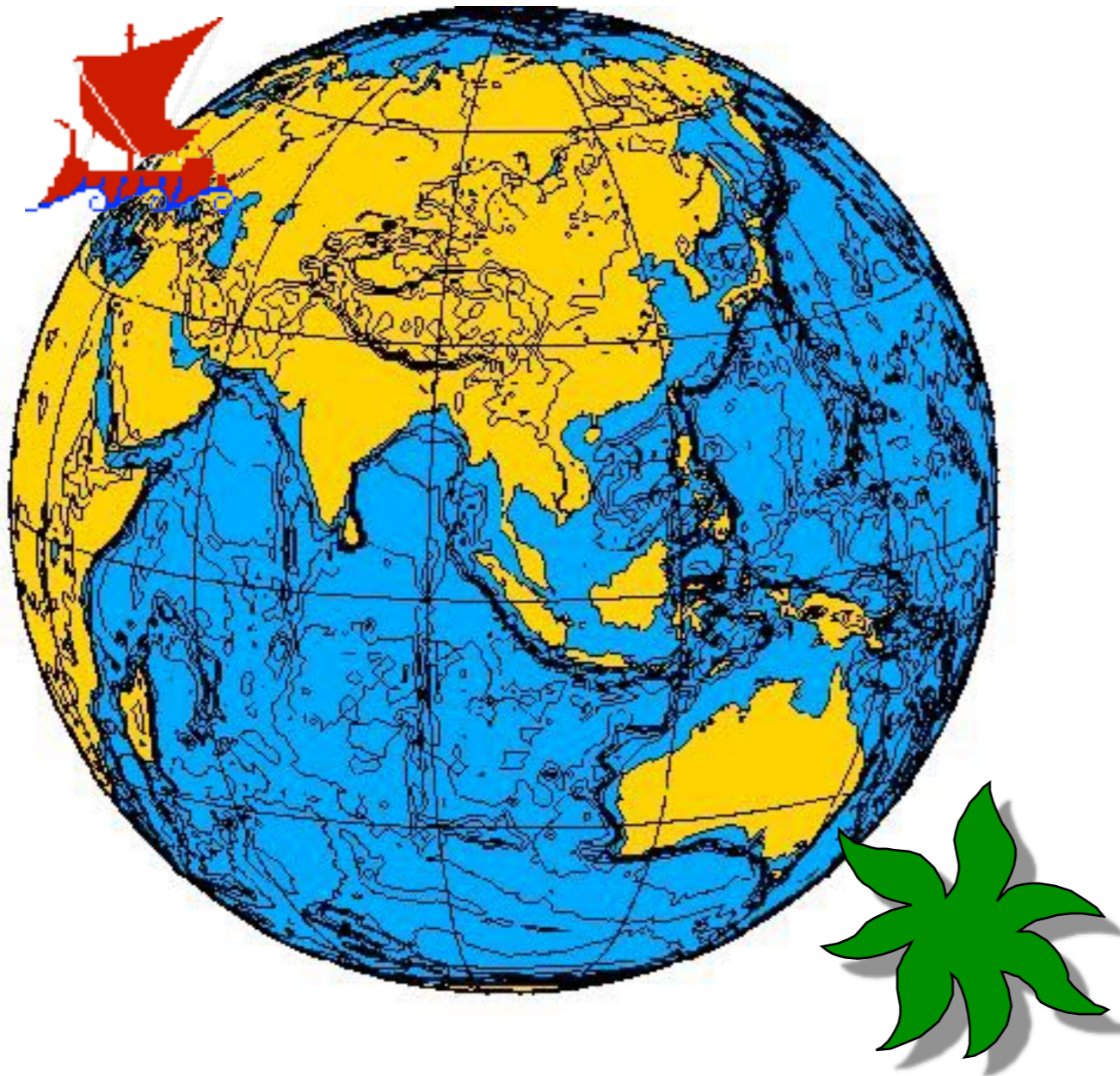


# Think Globally

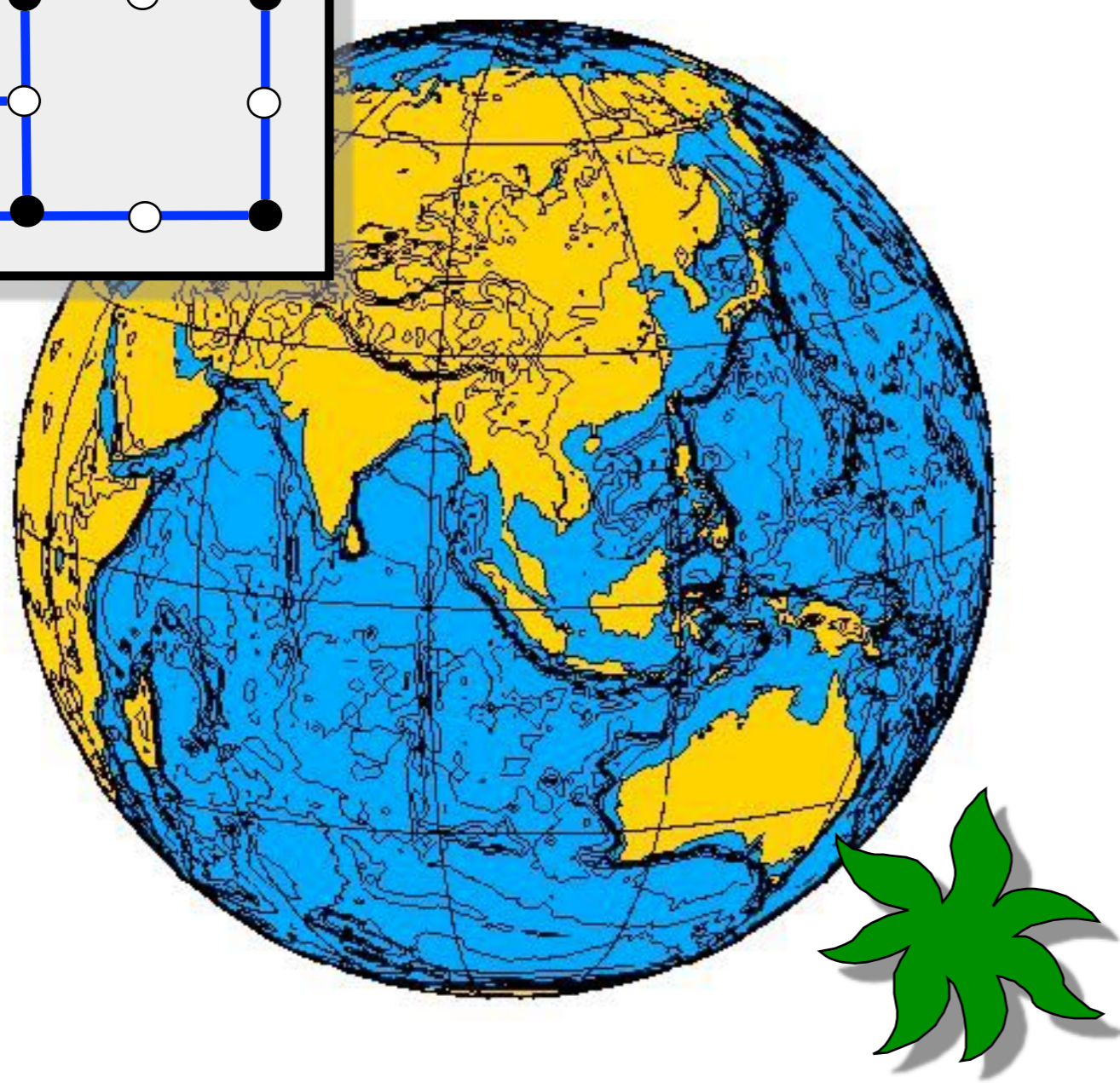
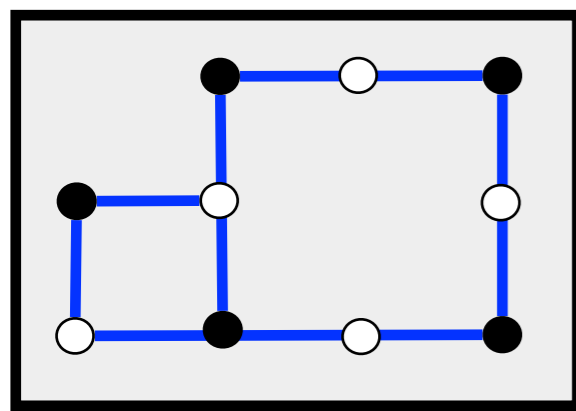


# Think Globally

- Consider a grid graph embedded on a sphere.

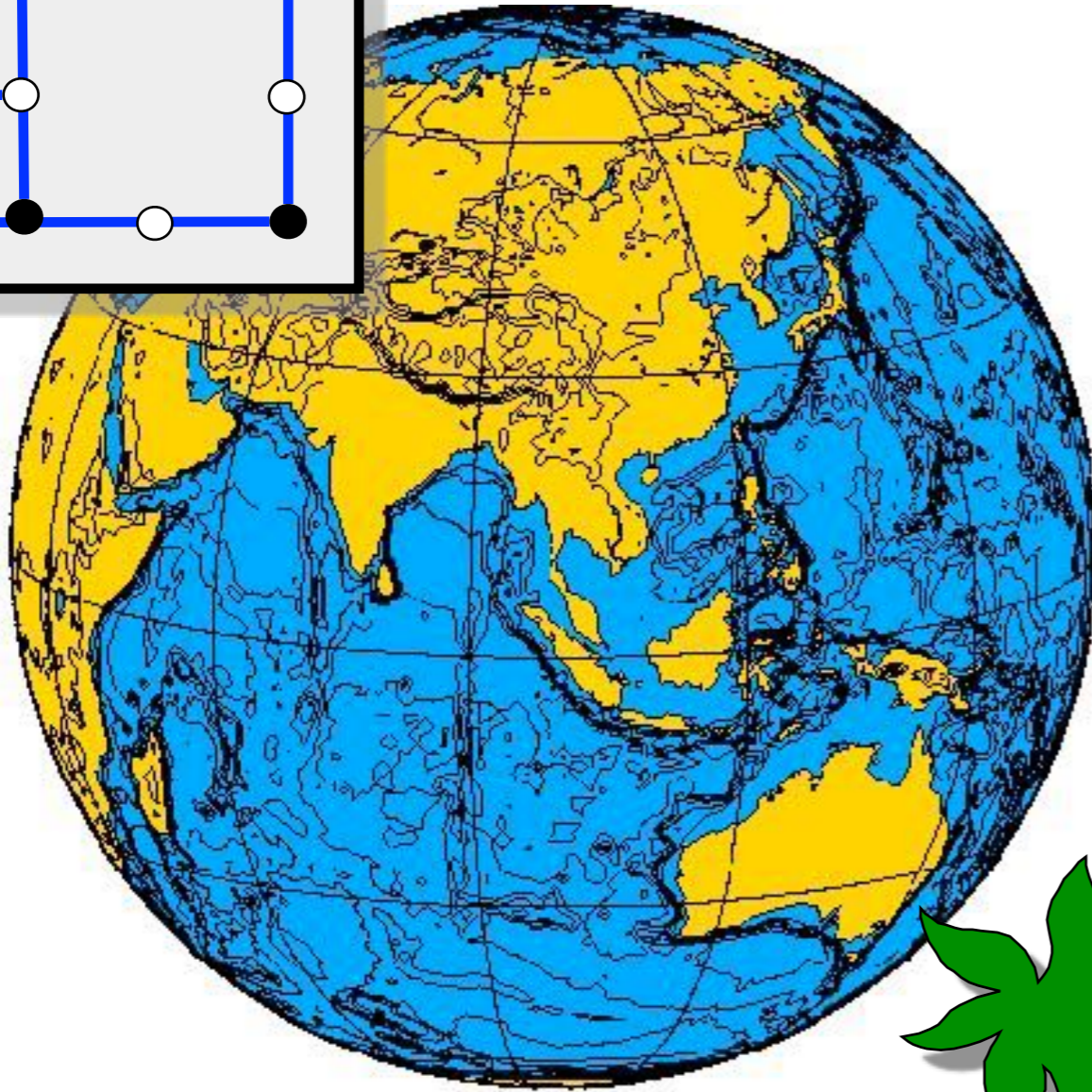
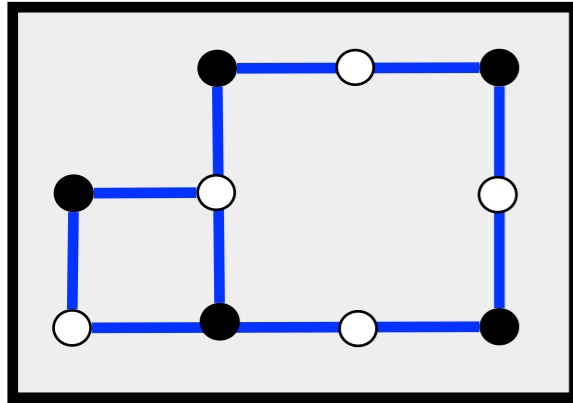


# Think Globally



- Consider a grid graph embedded on a sphere.

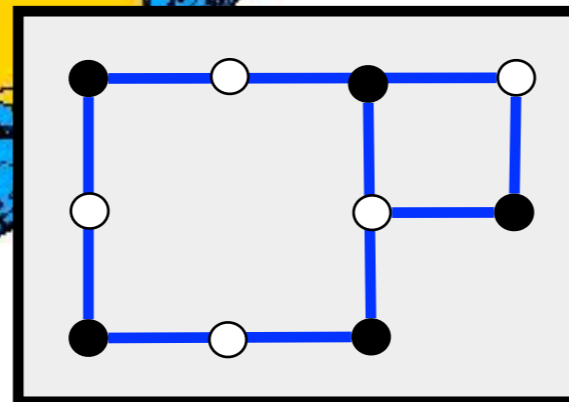
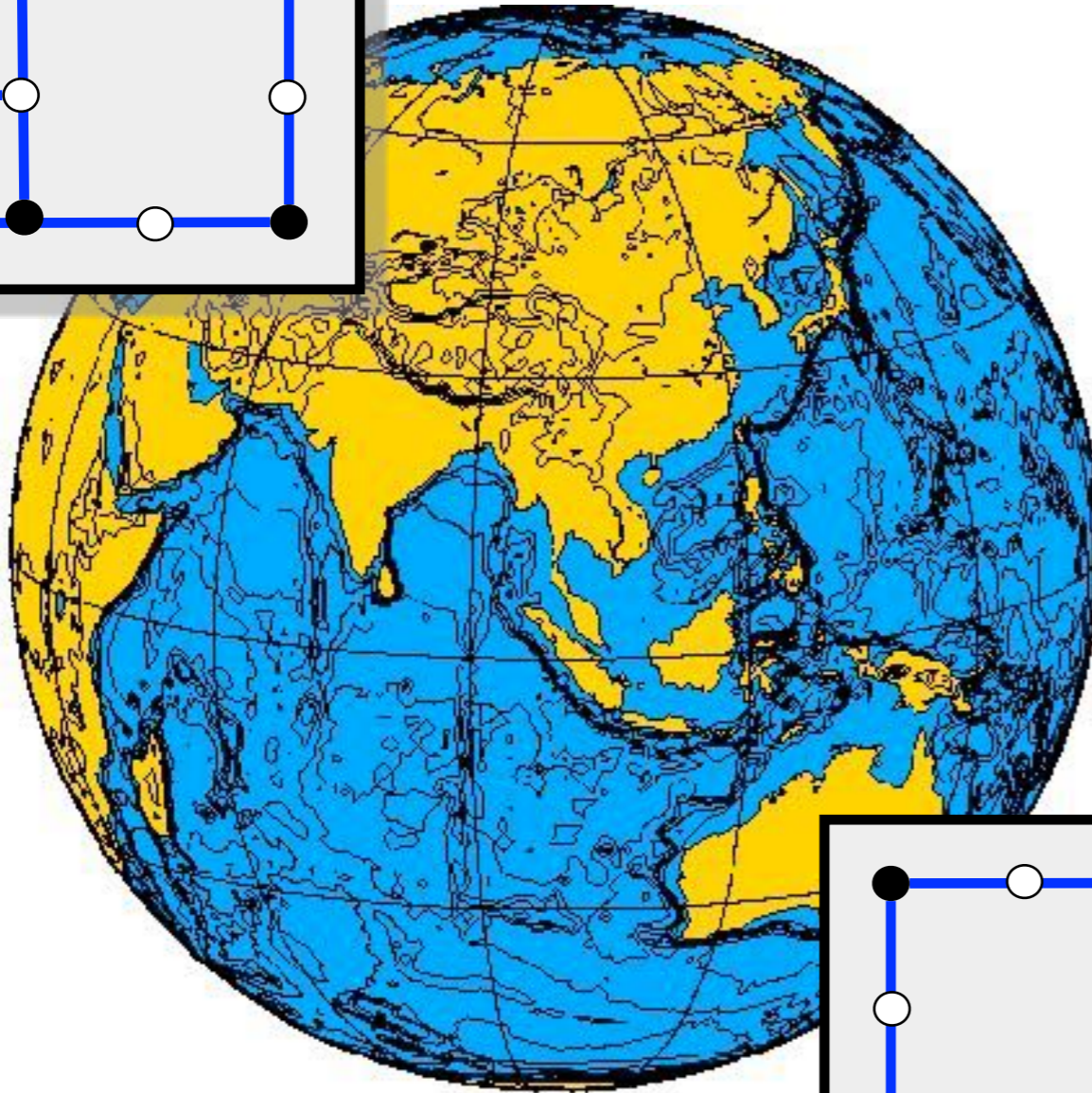
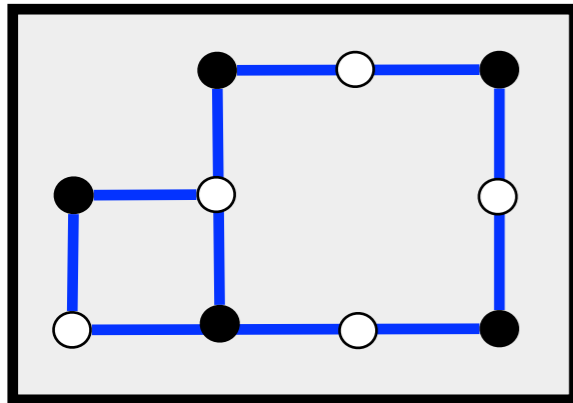
# Think Globally



- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.

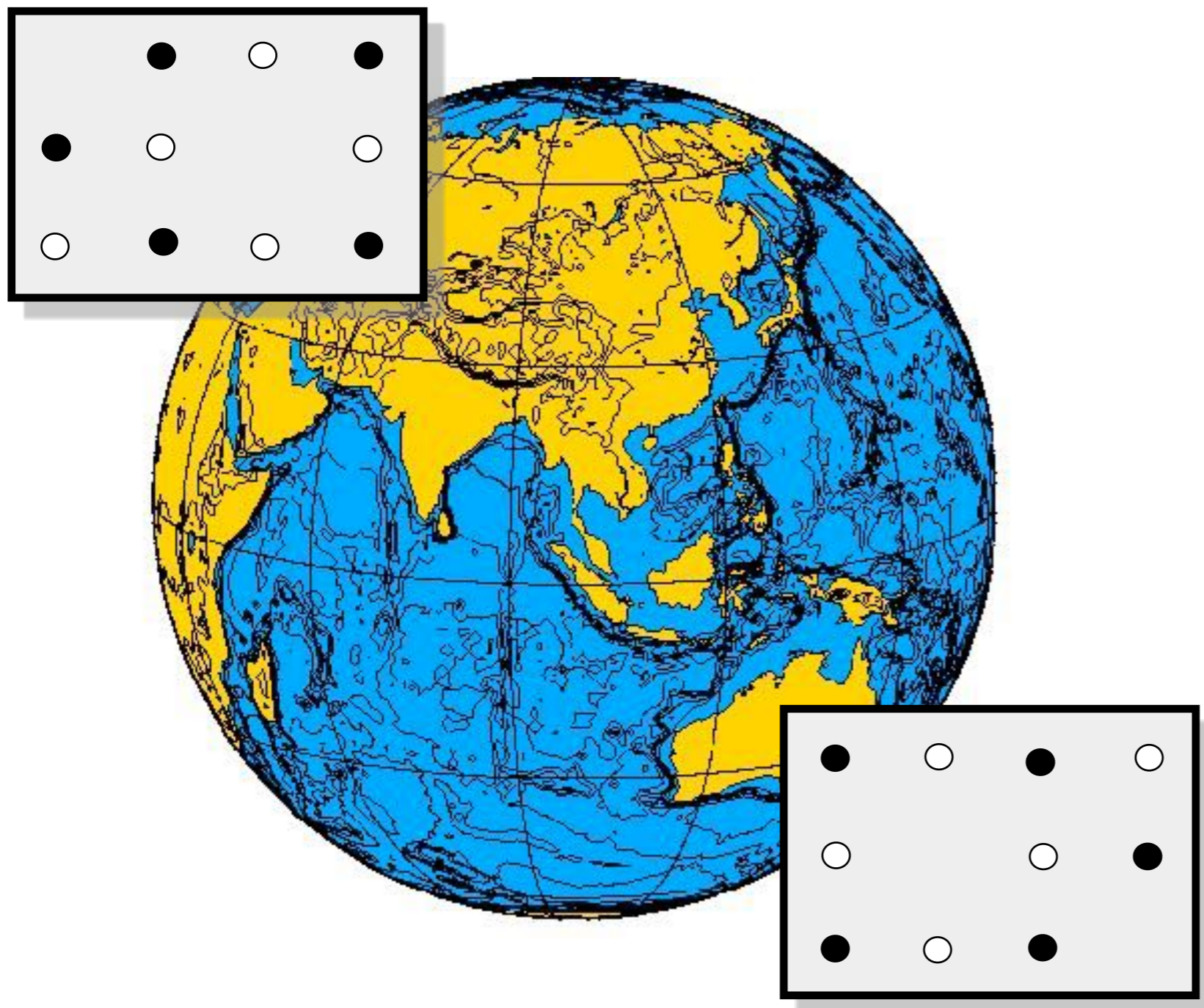


# Think Globally



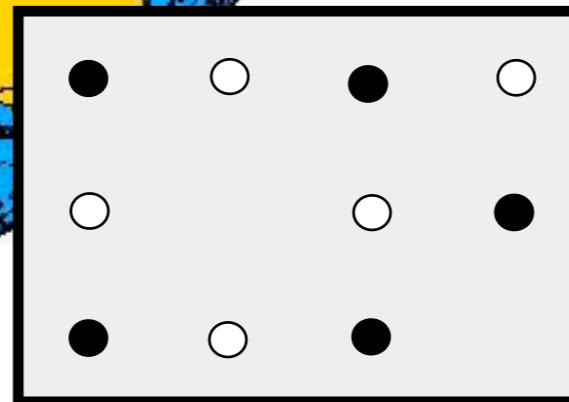
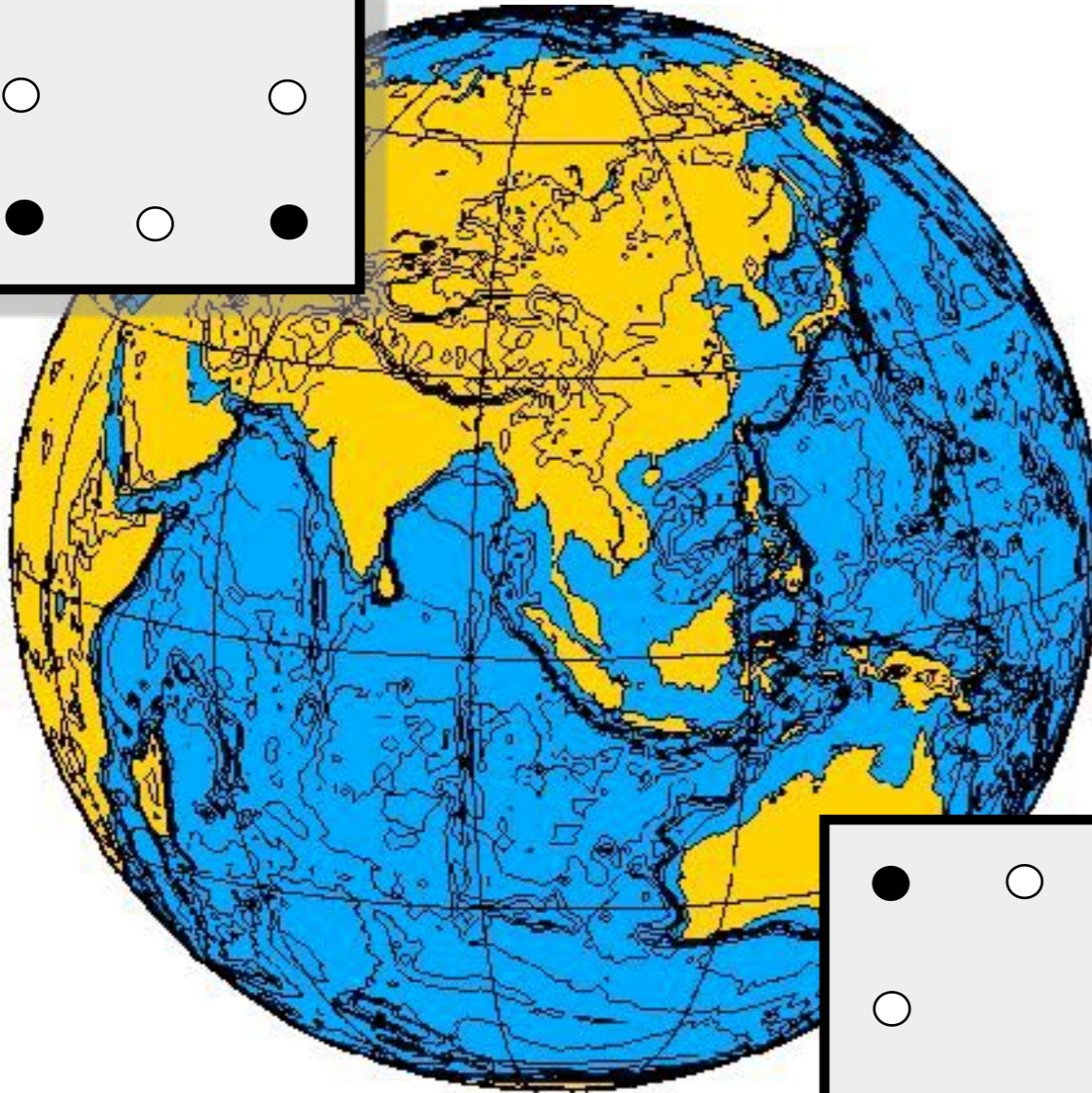
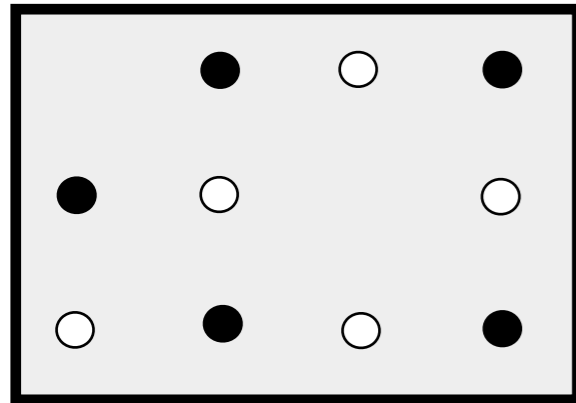
- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.

# Think Globally



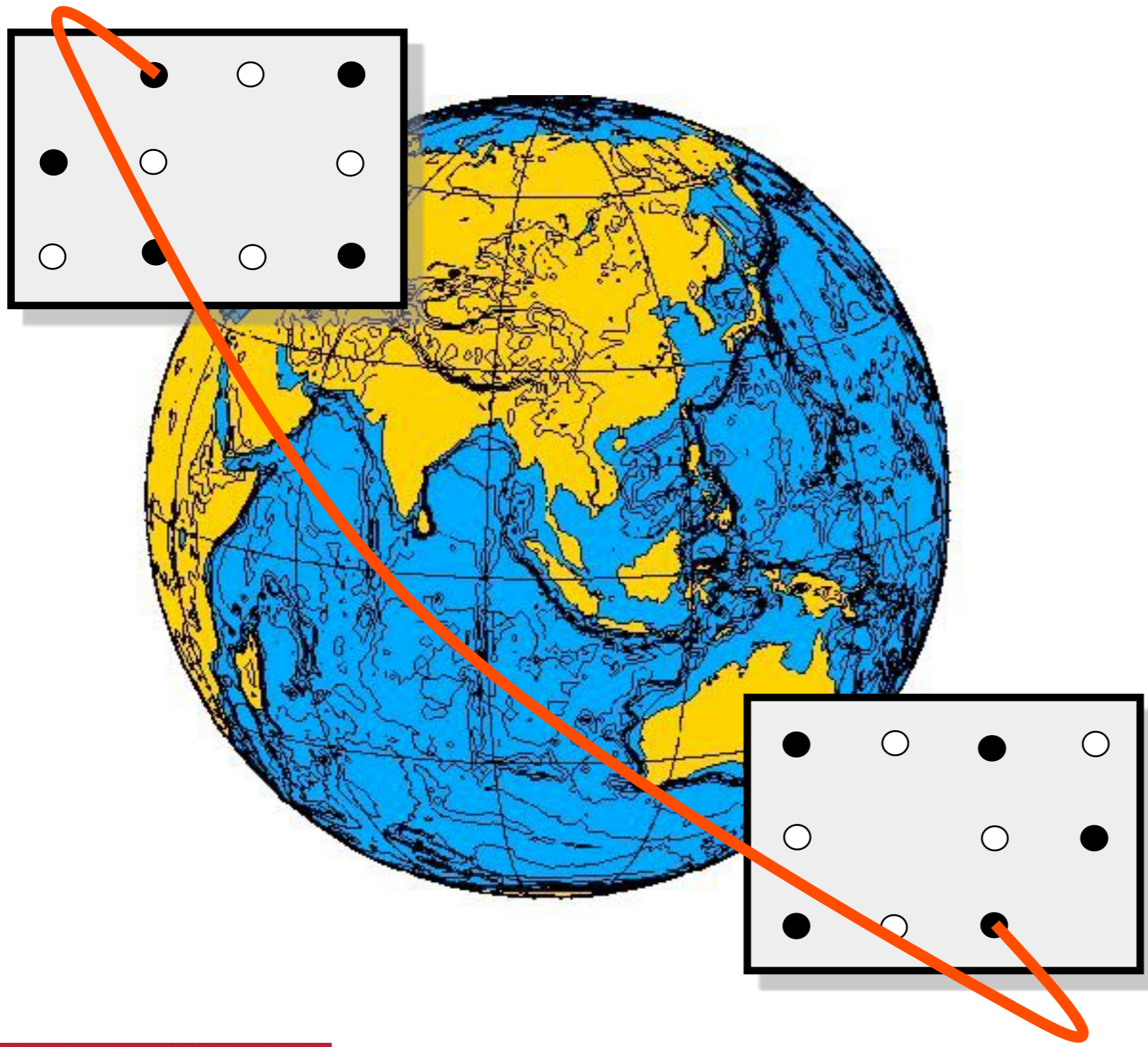
- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.

# Think Globally



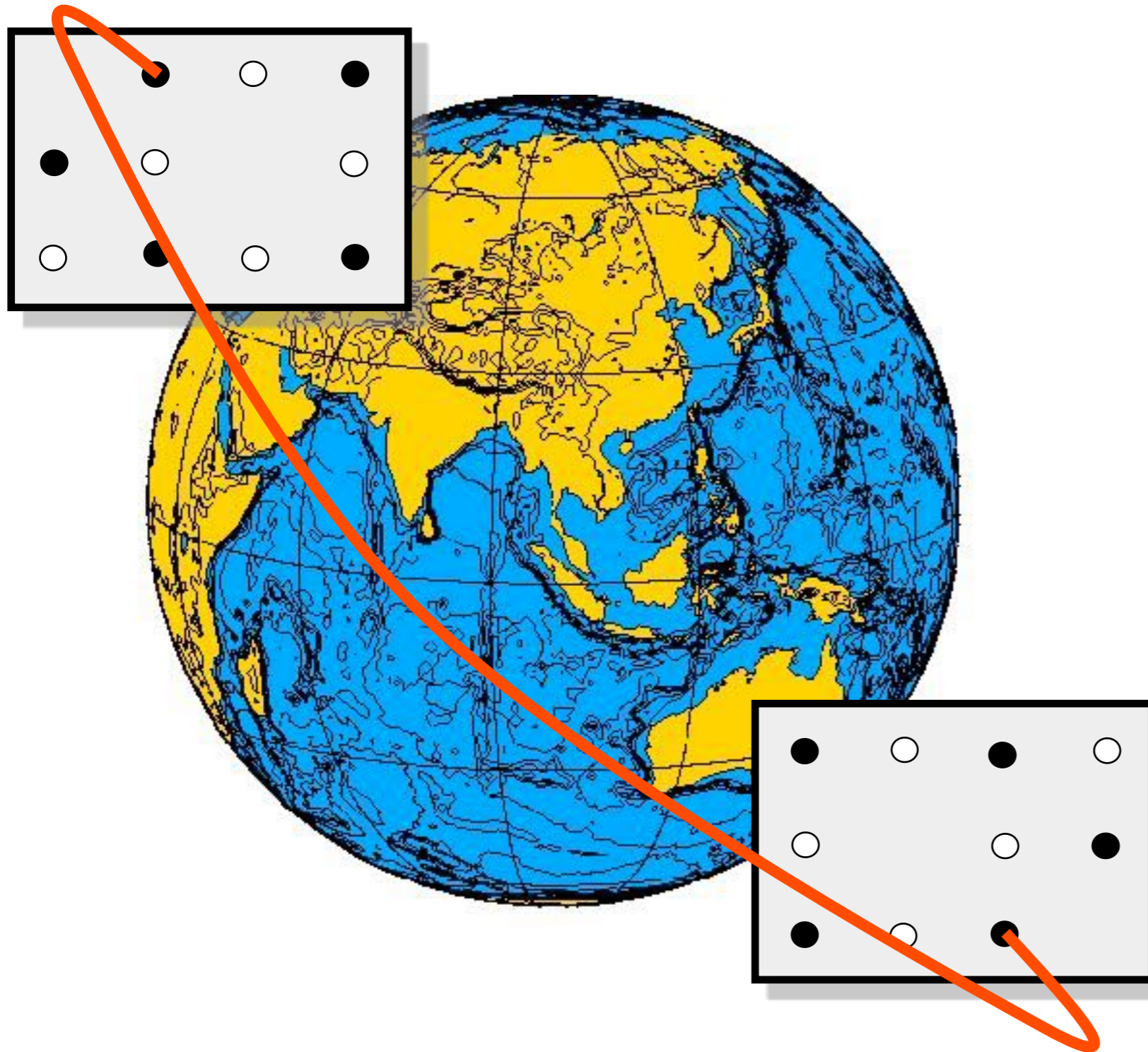
- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.

# Think Globally



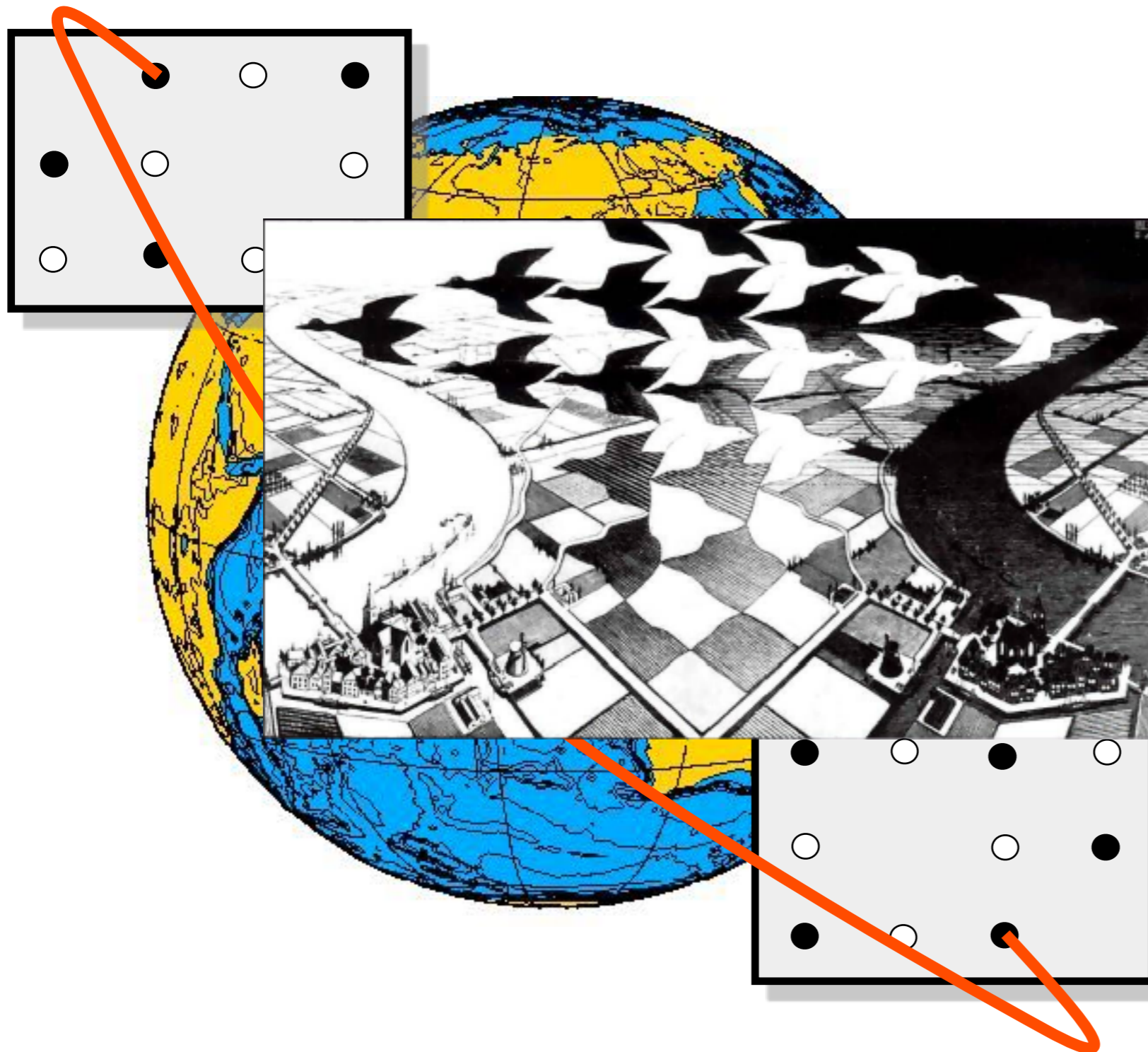
- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.

# Think Globally



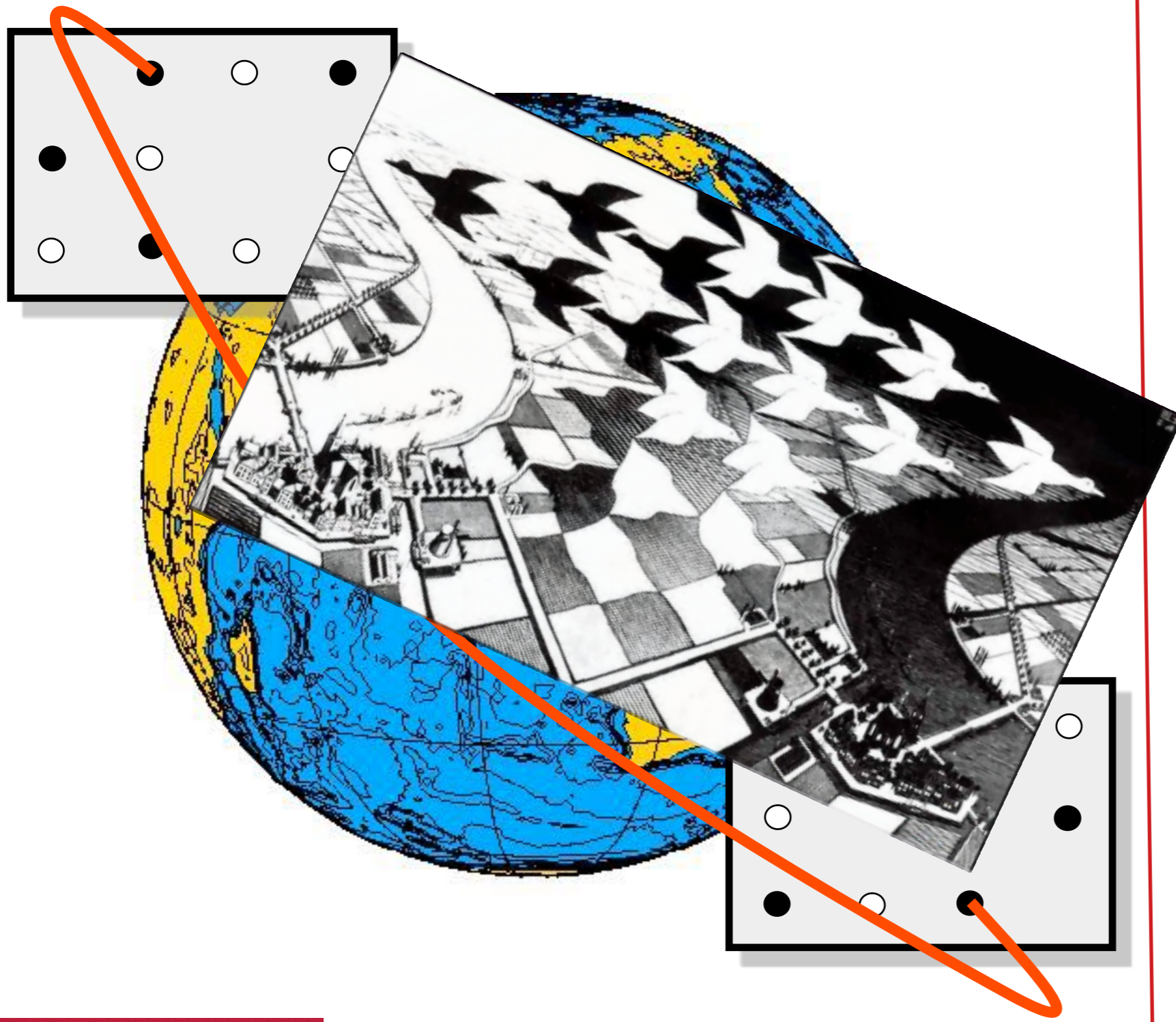
- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.

# Think Globally



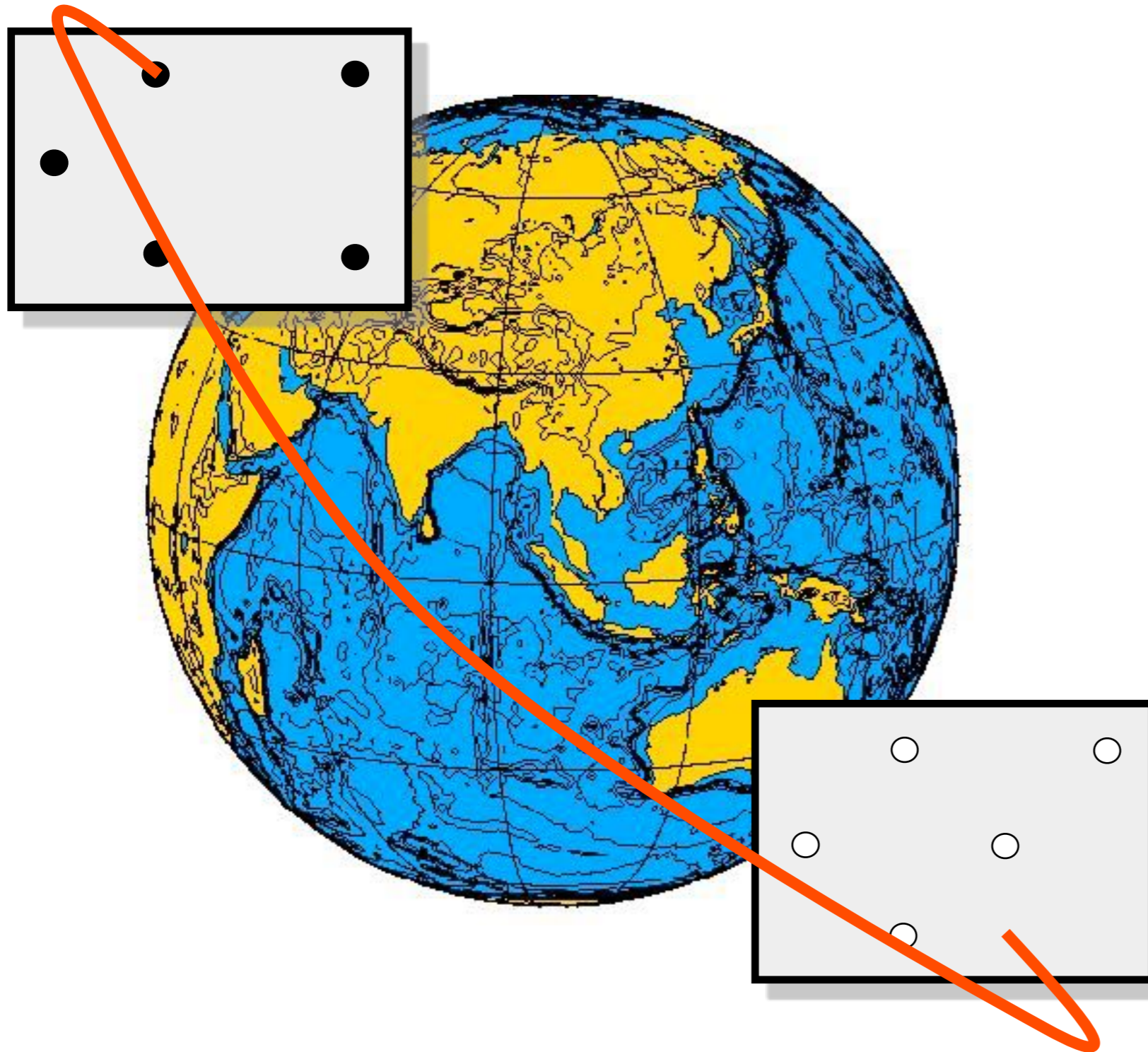
- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.

# Think Globally



- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.

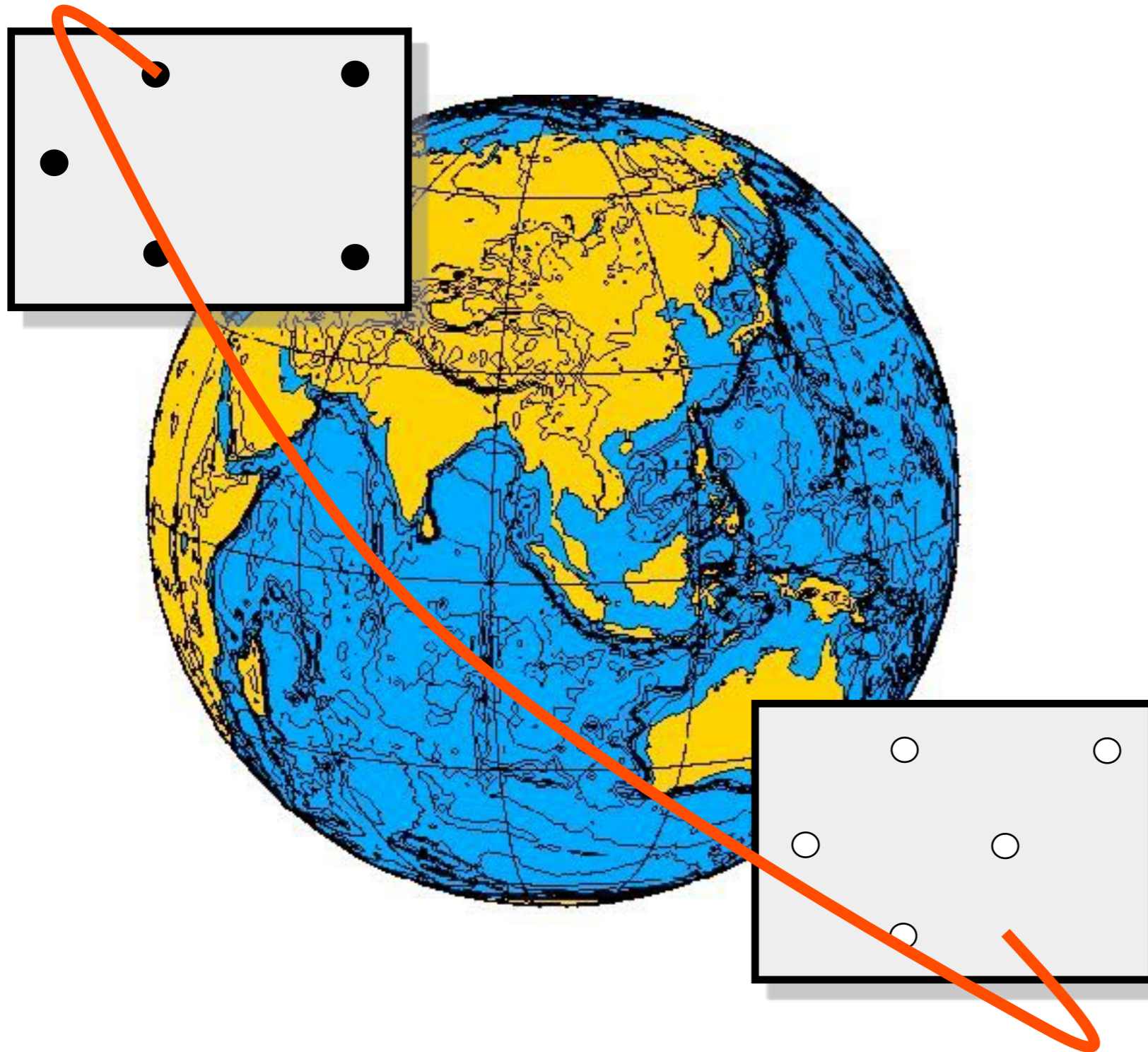
# Think Globally



- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.

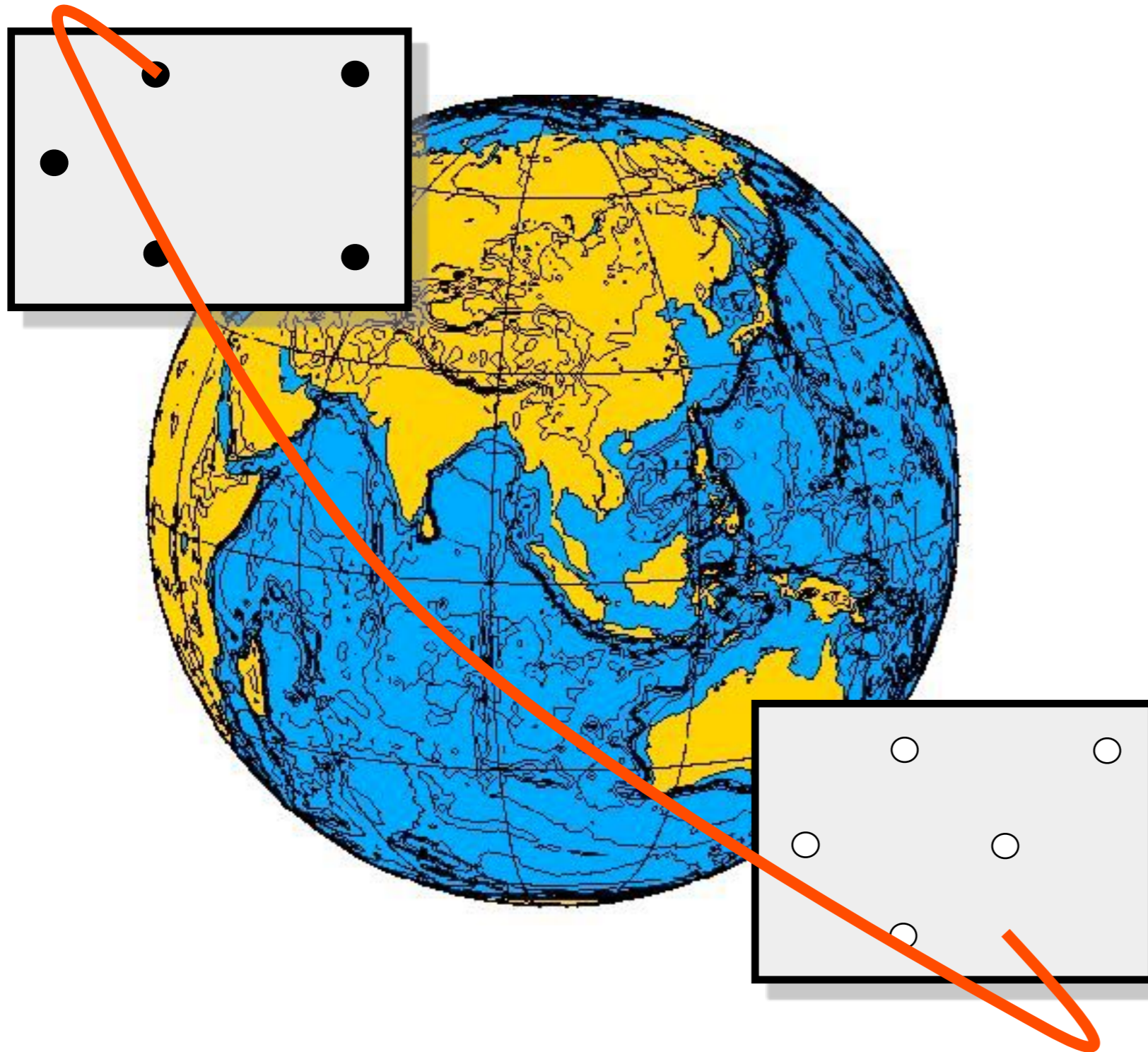


# Think Globally



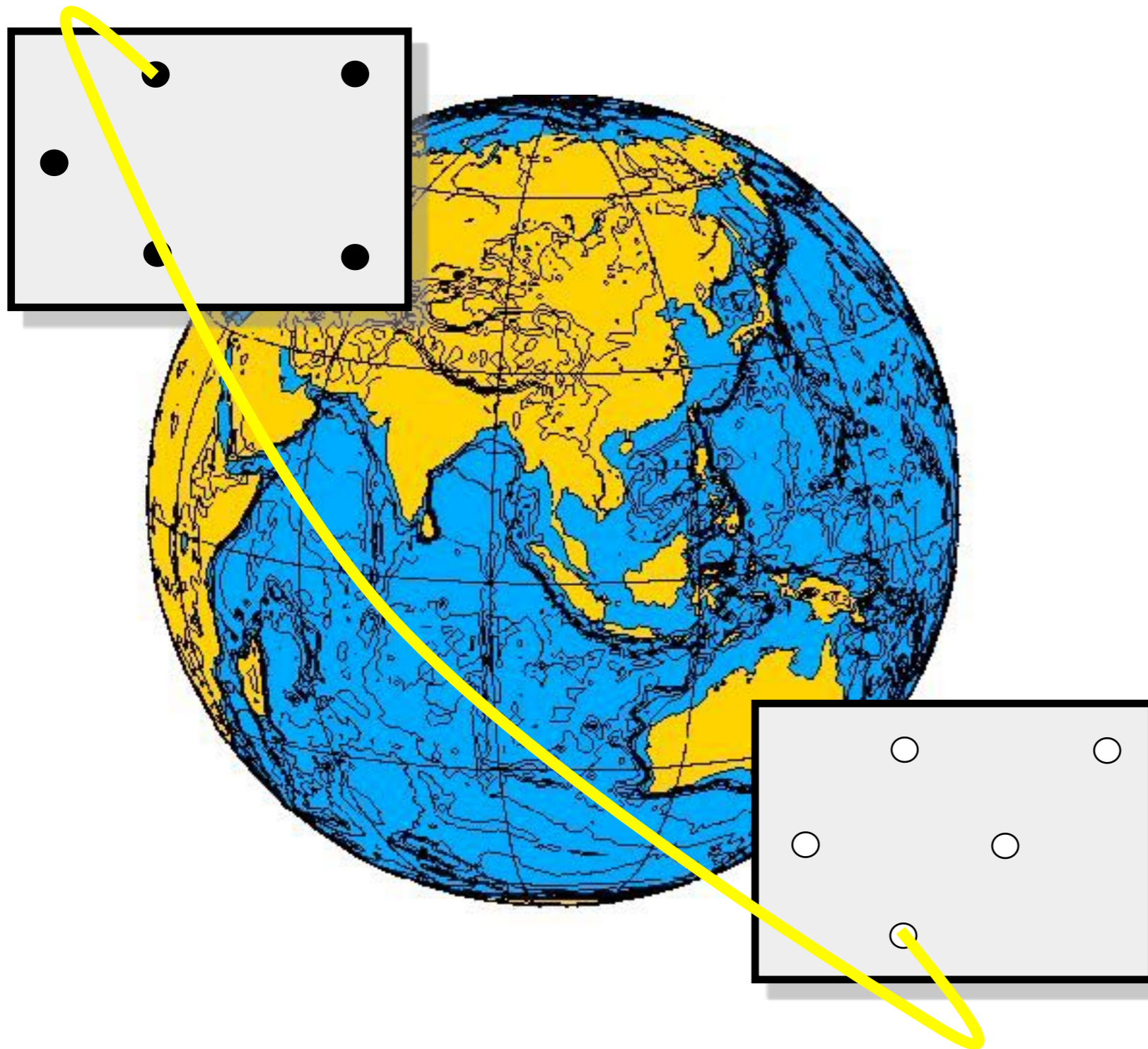
- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.
- Keep black and white points in opposite hemispheres.

# Think Globally



- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.
- Keep black and white points in opposite hemispheres.
- Long connections correspond to short ones in the original grid graph.

# Think Globally



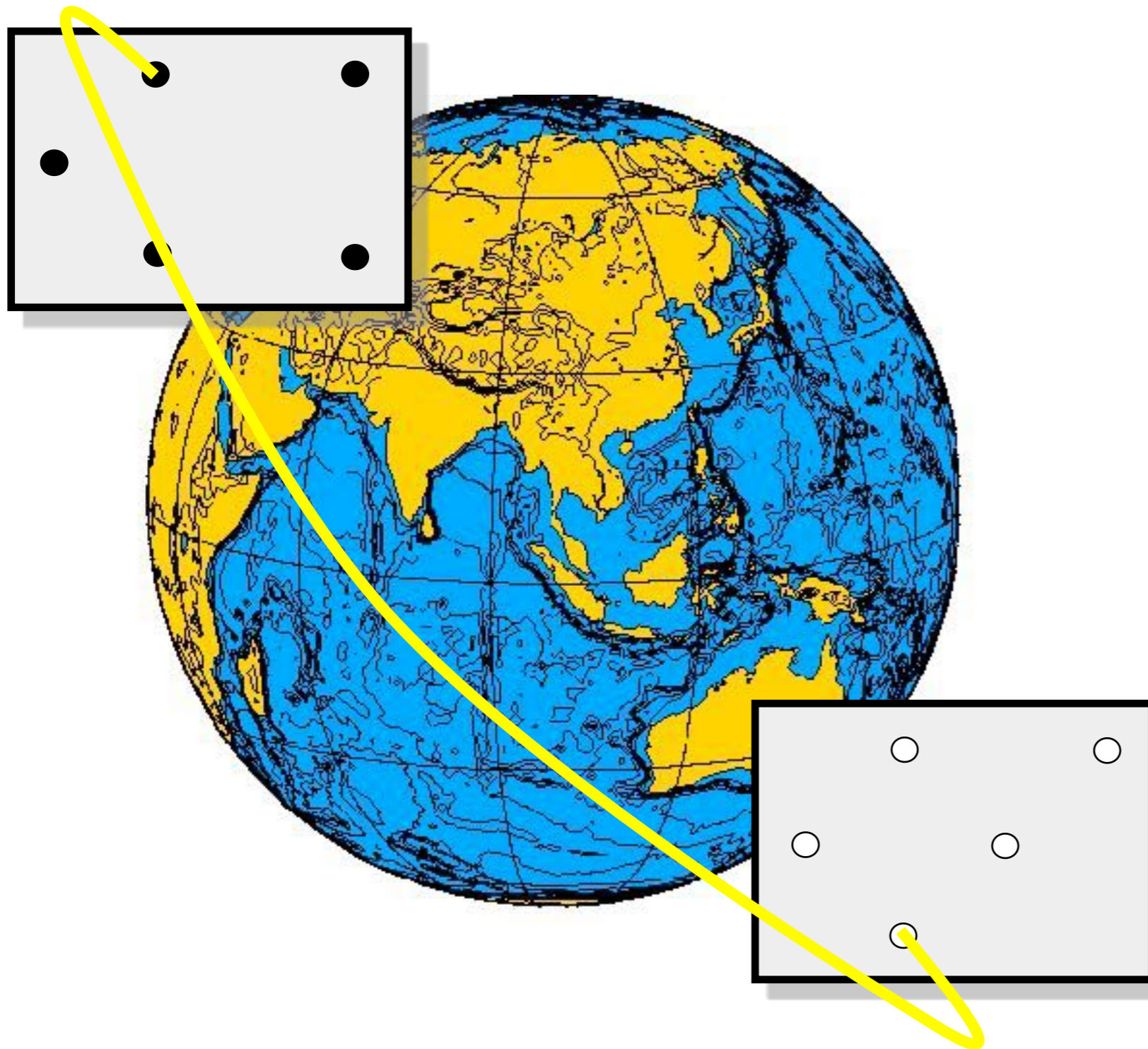
- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.
- Keep black and white points in opposite hemispheres.
- Long connections correspond to short ones in the original grid graph.

# Think Globally

$$\begin{aligned}
 & L_2(S(v_i), S(v_j))^2 \\
 &= [L_2((\cos(x_i \psi) \cos(y_i \psi), \sin(x_i \psi) \cos(y_i \psi), \sin(y_i \psi)), \\
 &\quad (\cos(\pi + x_j \psi) \cos(-y_j \psi), \sin(\pi + x_j \psi) \cos(-y_j \psi), \sin(-y_j \psi)))]^2 \\
 &= [\cos(x_i \psi) \cos(y_i \psi) + \cos(x_j \psi) \cos(y_j \psi)]^2 \\
 &\quad + [\sin(x_i \psi) \cos(y_i \psi) + \sin(x_j \psi) \cos(y_j \psi)]^2 + [\sin(y_i \psi) + \sin(y_j \psi)]^2 \\
 &= \left[ \left( 1 - \frac{(x_i \psi)^2}{2} + O((x_i \psi)^4) \right) \left( 1 - \frac{(y_i \psi)^2}{2} + O((y_i \psi)^4) \right) \right. \\
 &\quad \left. + \left( 1 - \frac{(x_j \psi)^2}{2} + O((x_j \psi)^4) \right) \left( 1 - \frac{(y_j \psi)^2}{2} + O((y_j \psi)^4) \right) \right]^2 \\
 &\quad + \left[ (x_i \psi - O((x_i \psi)^3)) \left( 1 - \frac{(y_i \psi)^2}{2} + O((y_i \psi)^4) \right) \right. \\
 &\quad \left. + (x_j \psi - O((x_j \psi)^3)) \left( 1 - \frac{(y_j \psi)^2}{2} + O((y_j \psi)^4) \right) \right]^2 \\
 &\quad + [y_i \psi - O((y_i \psi)^3) + y_j \psi - O((y_j \psi)^3)]^2 \\
 &= \left[ 2 - \frac{(x_i \psi)^2}{2} - \frac{(y_i \psi)^2}{2} - \frac{(x_j \psi)^2}{2} - \frac{(y_j \psi)^2}{2} + O(n^{-8}) \right]^2 \\
 &\quad + [x_i \psi + x_j \psi + O(n^{-6})]^2 + [y_i \psi + y_j \psi + O(n^{-6})]^2 \\
 &= [4 - 2(x_i \psi)^2 - 2(y_i \psi)^2 - 2(x_j \psi)^2 - 2(y_j \psi)^2 + O(n^{-8})] \\
 &\quad + [(x_i \psi)^2 + (x_j \psi)^2 + 2x_i x_j \psi^2 + O(n^{-8})] \\
 &\quad + [(y_i \psi)^2 + (y_j \psi)^2 + 2y_i y_j \psi^2 + O(n^{-8})] \\
 &= 4 - (x_i - x_j)^2 \psi^2 - (y_i - y_j)^2 \psi^2 + O(n^{-8}).
 \end{aligned}$$

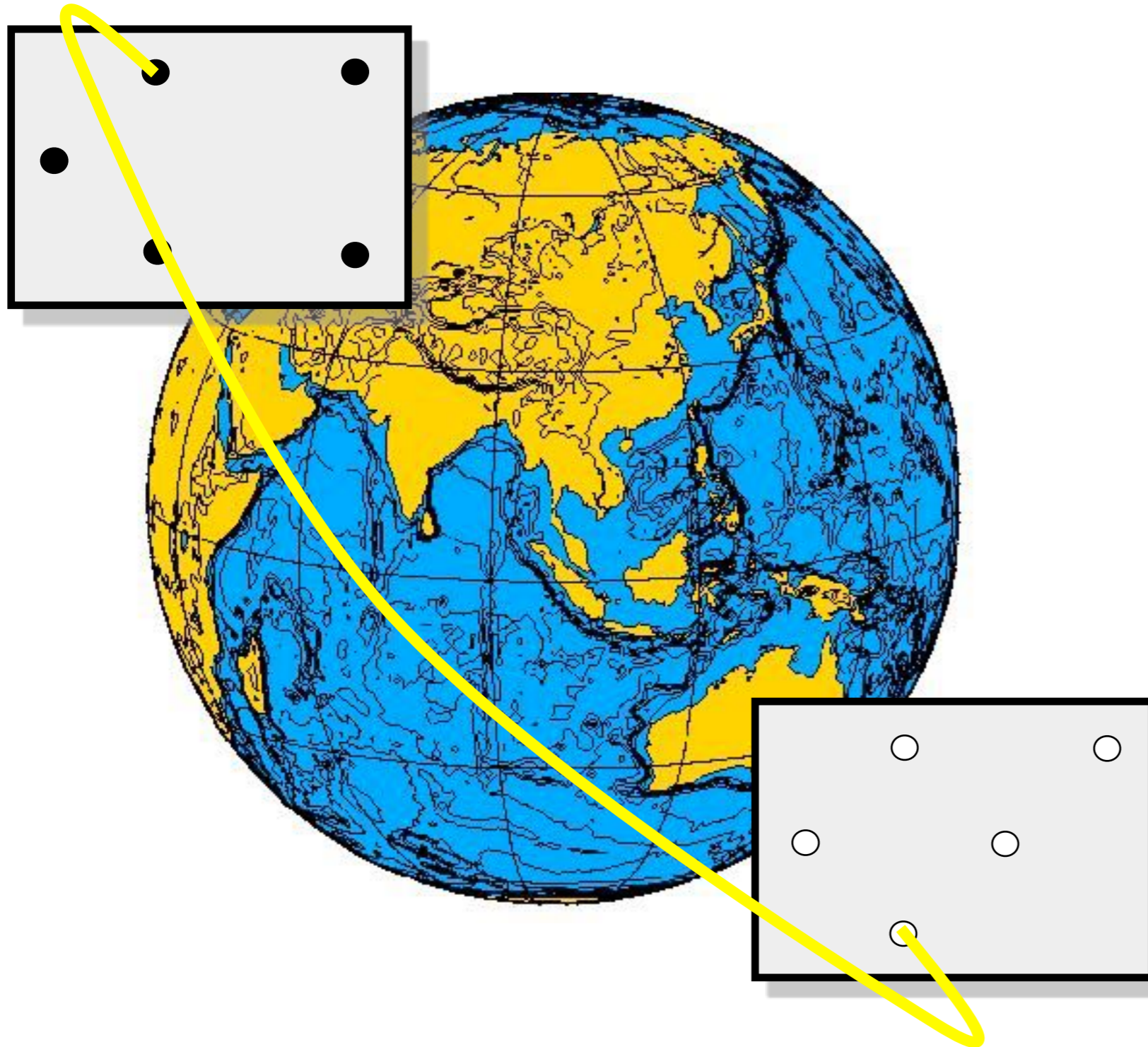
- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.
- Keep black and white points in opposite hemispheres.
- Long connections correspond to short ones in the original grid graph.

# Think Globally



- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.
- Keep black and white points in opposite hemispheres.
- Long connections correspond to short ones in the original grid graph.

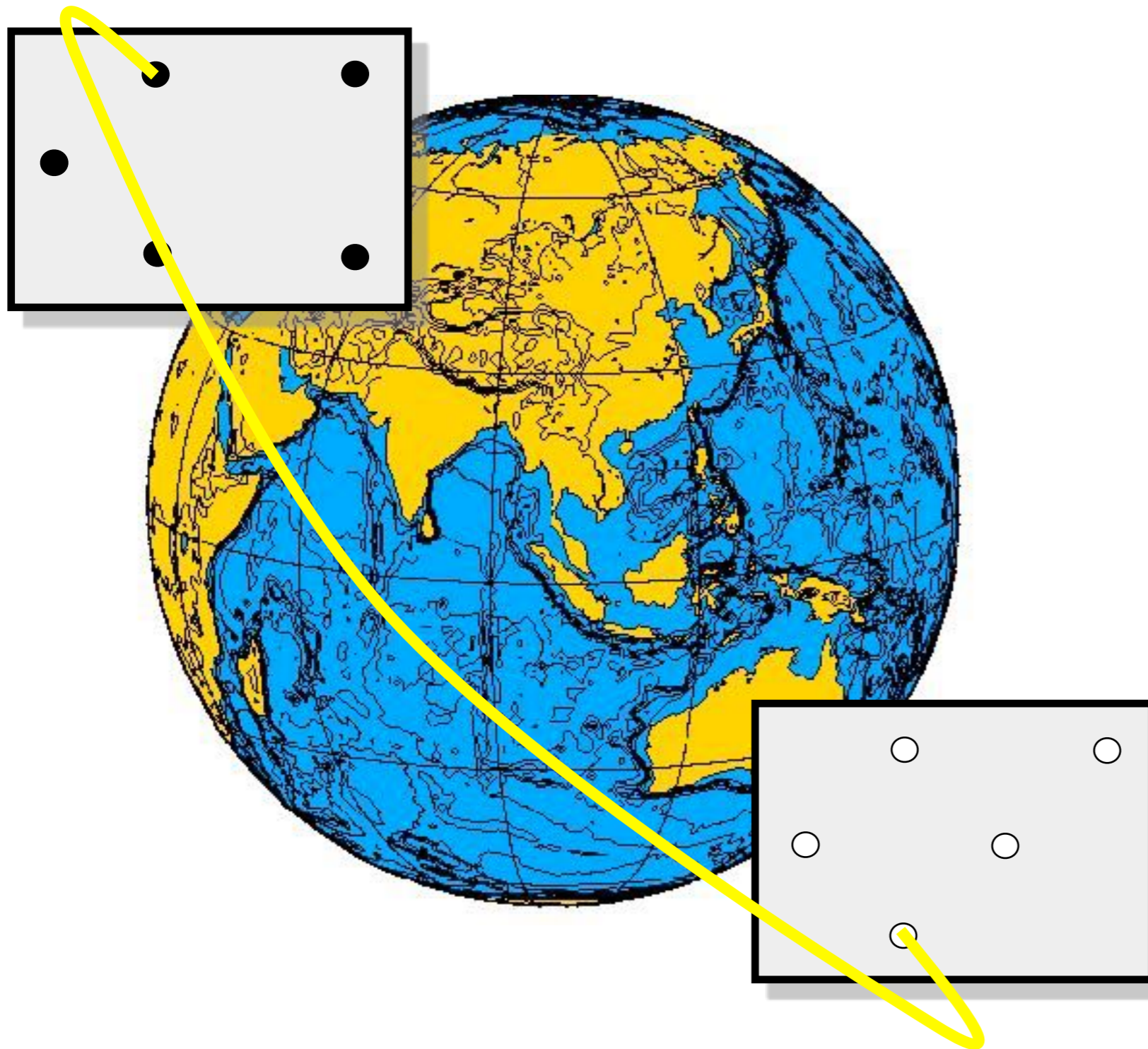
# Think Globally



- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.
- Keep black and white points in opposite hemispheres.
- Long connections correspond to short ones in the original grid graph.

**Theorem:**

# Think Globally



- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.
- Keep black and white points in opposite hemispheres.
- Long connections correspond to short ones in the original grid graph.

## Theorem:

Finding a longest tour for Euclidean distances in 3D is an NP-hard problem.

Andreas S. Schulz  
Martin Skutella  
Sebastian Stiller  
Dorothea Wagner  
*Editors*

# Gems of Combinatorial Optimization and Graph Algorithms

 Springer

## globally

- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.
- Keep black and white points in opposite hemispheres.
- Long connections correspond to short ones in the original grid graph.

### **Theorem:**

Finding a longest tour for Euclidean distances in 3D is an NP-hard problem.



Andreas S. Schulz  
Martin Skutella  
Sebastian Stiller  
Dorothea Wagner  
Editors

# Gems of Combinatorial Optimization and Graph Algorithms

 Springer

## Finding Longest Geometric Tours

Sándor P. Fekete

**Abstract** We discuss the problem of finding a longest tour for a set of points in a geometric space. In particular, we show that a longest tour for a set of  $n$  points in the plane can be computed in time  $O(n)$  if distances are determined by the Manhattan metric, while the same problem is NP-hard for points on a sphere under Euclidean distances.

### 1 Introduction: Short and Long Roundtrips

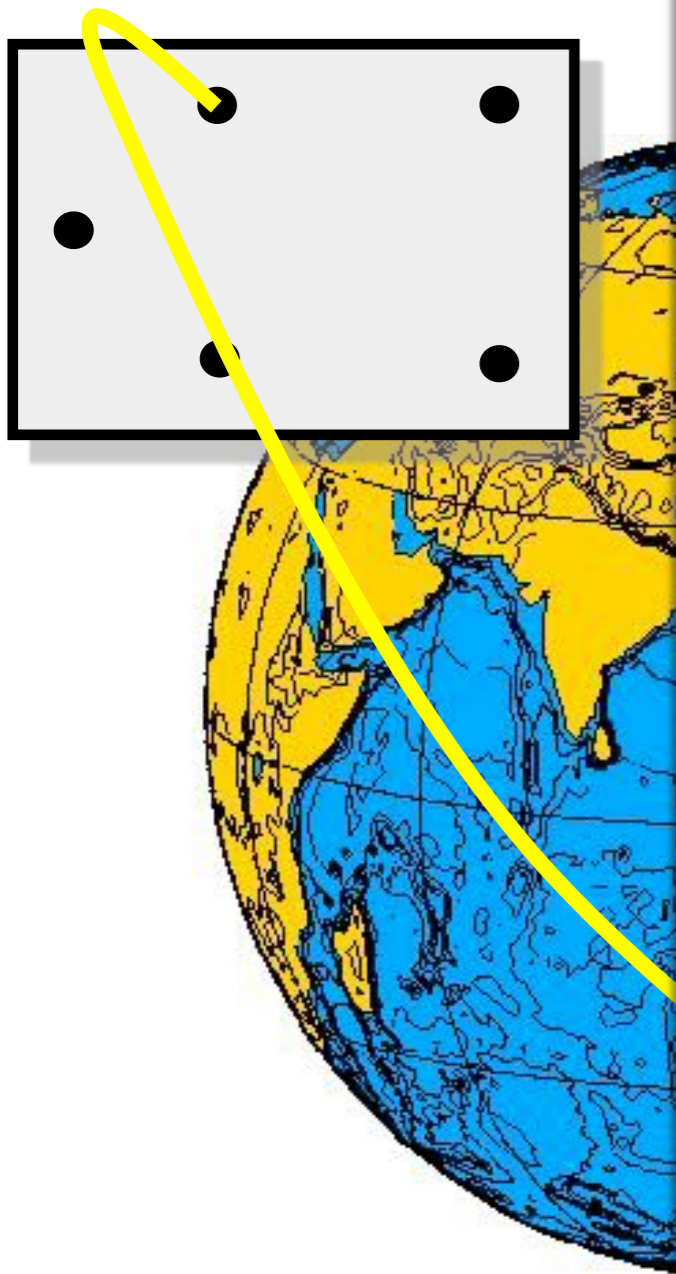
The Traveling Salesman Problem (TSP) is one of the classic problems of combinatorial optimization. Given a complete graph  $G = (V, E)$  with edge weights  $c(e)$  for all edges  $e \in E$ , find a shortest roundtrip through all vertices, i.e., a cyclic permutation  $\pi$  from the symmetric group  $S_n$  of all  $n$  vertices  $v_1, \dots, v_n$ , such that the total tour length  $\sum_{i=1}^n c(\{v_i, v_{\pi(i)}\})$  is minimized.

The difficulties of finding a good roundtrip are well known. The classical Odyssey is illustrated in Figure 1: according to legend, it took Ulysses many years to complete his voyage. One justification is the computational complexity of the TSP: it is one of the most famous NP-hard problems, so it does indeed take many years of CPU time to find provably optimal solutions for non-trivial instances.

However, there is an even more convincing justification for Ulysses' failure to be home in a more timely fashion: it was not him who chose his route. Instead, malevolent gods caused a deliberately long voyage—so the real objective was to *maximize* the traveled distance. This motivates the MaxTSP: Find a roundtrip that visits all vertices in a weighted graph, such that the total tour length is maximized.

In this chapter, we study longest tours in a geometric setting, in which the vertices are points in two- or three-dimensional space, and the edge weights are induced

Sándor P. Fekete  
Department of Computer Science, TU Braunschweig, 38106 Braunschweig, Germany, e-mail:  
s.fekete@tu-bs.de



## The Geometric Maximum Traveling Salesman Problem

ALEXANDER BARVINOK

*University of Michigan, Ann Arbor, Michigan*

SÁNDOR P. FEKETE

*Braunschweig University of Technology,  
Braunschweig, Germany*

DAVID S. JOHNSON

*AT&T Labs, Florham Park, New Jersey*

ARIE TAMIR

*Tel Aviv University, Tel Aviv, Israel*

GERHARD J. WOEGINGER

*University of Twente, Enschede,  
The Netherlands*

AND

RUSS WOODROOFE

*Cornell University, Ithaca, New York*

**Abstract.** We consider the traveling salesman problem when the cities are points in  $\mathbb{R}^d$  for some fixed  $d$  and distances are computed according to geometric distances, determined by some norm. We show that for any polyhedral norm, the problem of finding a tour of *maximum* length can be solved in

Preliminary versions of parts of this article appeared in the following two proceedings: BARVINOK, A. I., JOHNSON, D. S., WOEGINGER, G. J., AND WOODROOFE, R. 1998. The maximum traveling salesman problem under polyhedral norms. In *Proceedings of the 6th International Integer Programming Combinatorics Optimization Conference (IPCO IV)*. Lecture Notes in Computer Science, vol. 1412. Springer-Verlag, New York, 195–201, and FEKETE, S. P. 1999. Simplicity and hardness of the maximum Traveling Salesman Problem under geometric distances. In *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms (SODA 99)*. ACM, New York, 337–345.

A. Barvinok was supported by an Alfred P. Sloan Research Fellowship and National Science Foundation (NSF) grant DMS 9501129.

S. P. Fekete was partly supported by the Hermann-Minkowski-Minerva Center for Geometry at Tel Aviv University, while visiting the Center in March 1998; other parts were supported by the Deutsche Forschungsgemeinschaft, FE 407/3-1, when visiting Rice University in June 1998.

G. J. Woeginger was supported by the START program Y43-MAT of the Austrian Ministry of Science.

R. Woodroofe was supported by the NSF through the REU Program while at the Department of Mathematics, University of Michigan.

Authors' addresses: A. Barvinok, Department of Mathematics, University of Michigan, Ann Arbor, MI 48109-1009, e-mail: barvinok@math.lsa.umich.edu; S. P. Fekete, Department of Mathematical Optimization, Braunschweig University of Technology, 38106 Braunschweig, Germany, e-mail: s.fekete@tu-bs.de; D. S. Johnson, AT&T Research, AT&T Labs, Florham Park, NJ 07932-0971, e-mail: dsj@research.att.com; A. Tamir, School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel, e-mail: atamir@math.tau.ac.il; G. J. Woeginger, Department of Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands, e-mail: g.j.woeginger@math.utwente.nl; R. Woodroofe, Department of Mathematics, Cornell University, Ithaca, NY 14853-4201, e-mail: paranoia@math.cornell.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 1515 Broadway, New York, NY 10036 USA, fax: +1 (212) 869-0481, or permissions@acm.org.

© 2003 ACM 0004-5411/03/0900-0641 \$5.00

*Journal of the ACM*, Vol. 50, No. 5, September 2003, pp. 641–664.

consider a grid graph  
embedded on a sphere.

second copy at the  
nodes.

longest connections are  
between antipodal points.

grid graphs are bipartite.

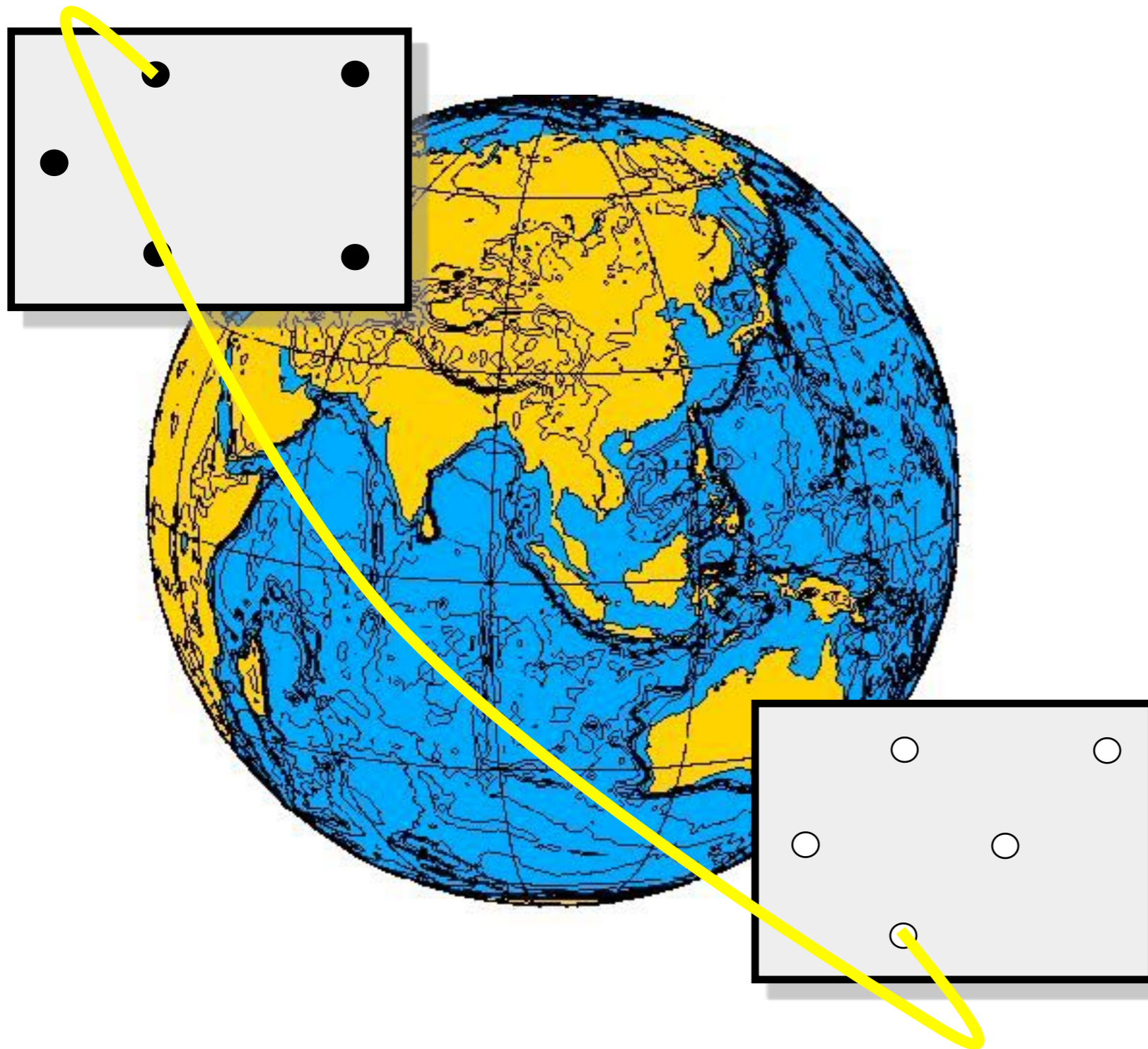
black and white points  
opposite hemispheres.

long connections correspond  
to short ones in the original  
graph.

**Remark:**

finding a longest tour for  
ideal distances in 3D is  
NP-hard problem.

# Think Globally



- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.
- Keep black and white points in opposite hemispheres.
- Long connections correspond to short ones in the original grid graph.

## Theorem:

Finding a longest tour for Euclidean distances in 3D is an NP-hard problem.

# Think Globally



- Consider a grid graph embedded on a sphere.
- Add second copy at the antipodes.
- Longest connections are between antipodal points.
- Grid graphs are bipartite.
- Keep black and white points in opposite hemispheres.
- Long connections correspond to short ones in the original grid graph.

## Theorem:

Finding a longest tour for Euclidean distances in 3D is an NP-hard problem.

# Think Globally

## The Open Problems Project

edited by [Erik D. Demaine](#)

[Joseph S. B. Mitchell](#)

[Joseph O'Rourke](#)

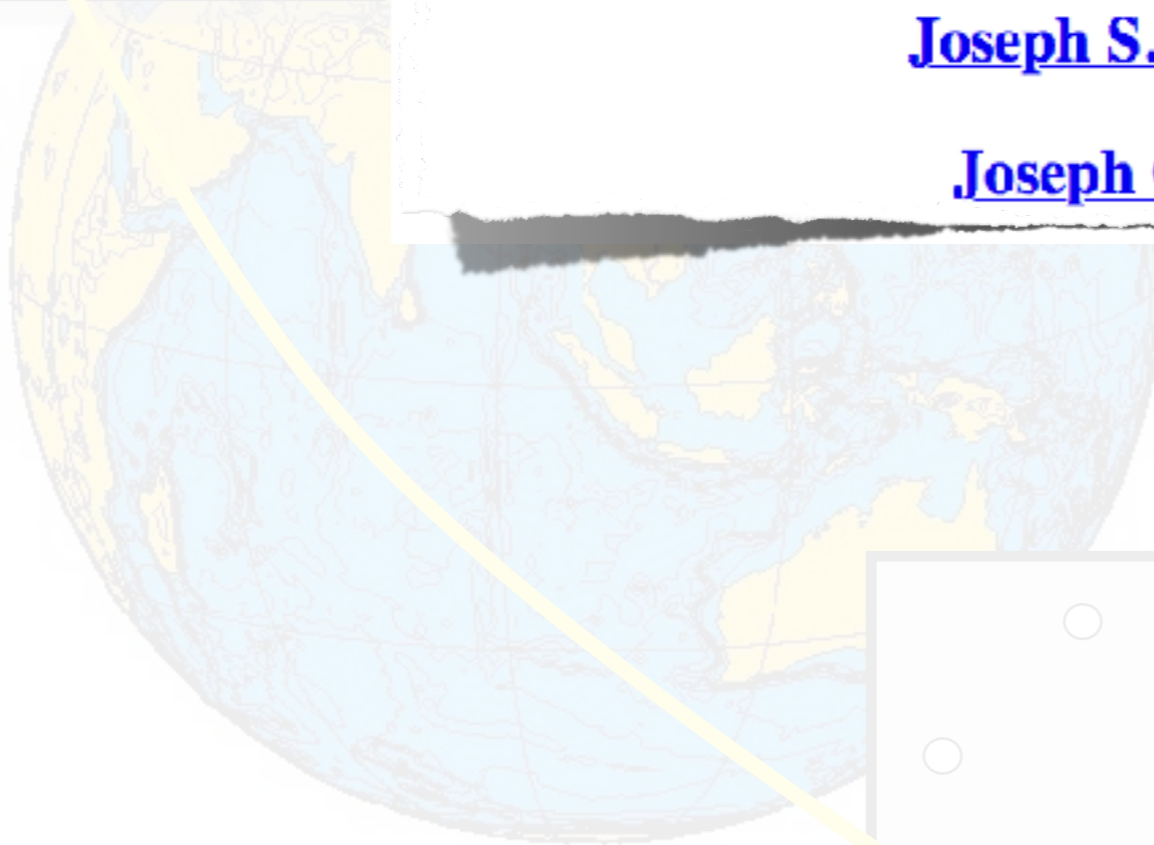
- Consider a grid graph on a sphere. Copy at the

connections are nodal points. are bipartite. and white points in opposite hemispheres.

- Long connections correspond to short ones in the original grid graph.

### Theorem:

Finding a longest tour for Euclidean distances in 3D is an NP-hard problem.



# Think Globally

## The Open Problems Project

edited by [Erik D. Demaine](#)

[Joseph S. B. Mitchell](#)

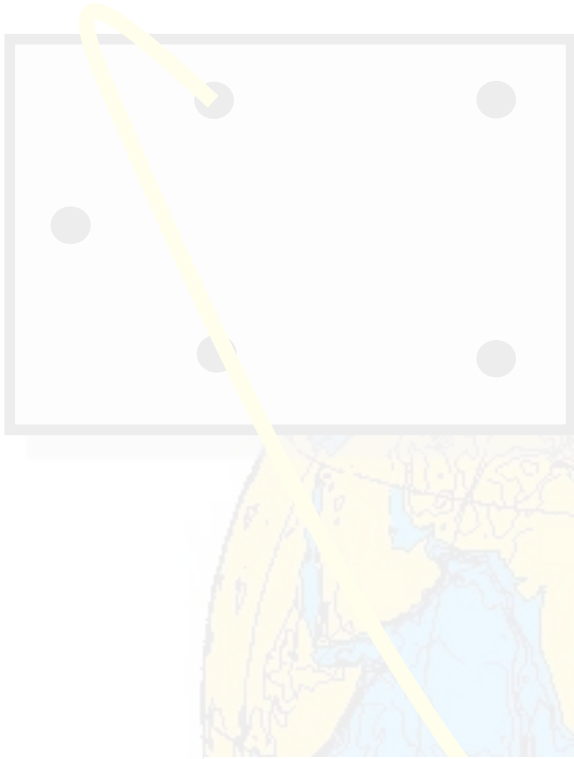
[Joseph O'Rourke](#)

### Problem 49: Planar Euclidean Maximum TSP

Theorem:

Finding a longest tour for Euclidean distances in 3D is an NP-hard problem.

# Think Globally



## The Open Problems Project

edited by [Erik D. Demaine](#)

[Joseph S. B. Mitchell](#)

[Joseph O'Rourke](#)

### Problem 49: Planar Euclidean Maximum TSP

**CONJECTURE 4.1.** *The Maximum TSP for Euclidean distances in the plane is an NP-hard problem.*

1. Introduction
2. Longest Tours
3. Stars and Matchings
4. Nonsimple Polygons
5. Optimal Area
6. Turn Cost





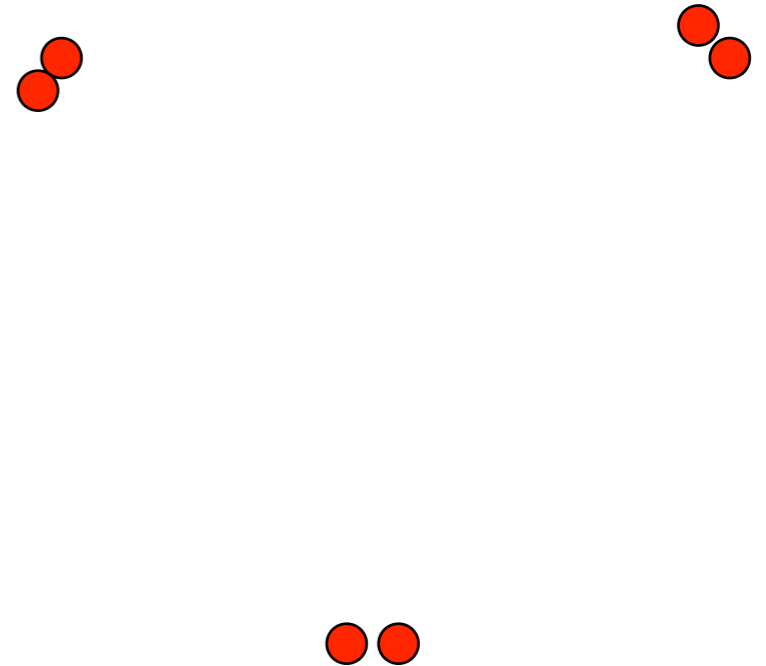
## Problem 7.1:

## Problem 7.1:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

## Problem 7.1:

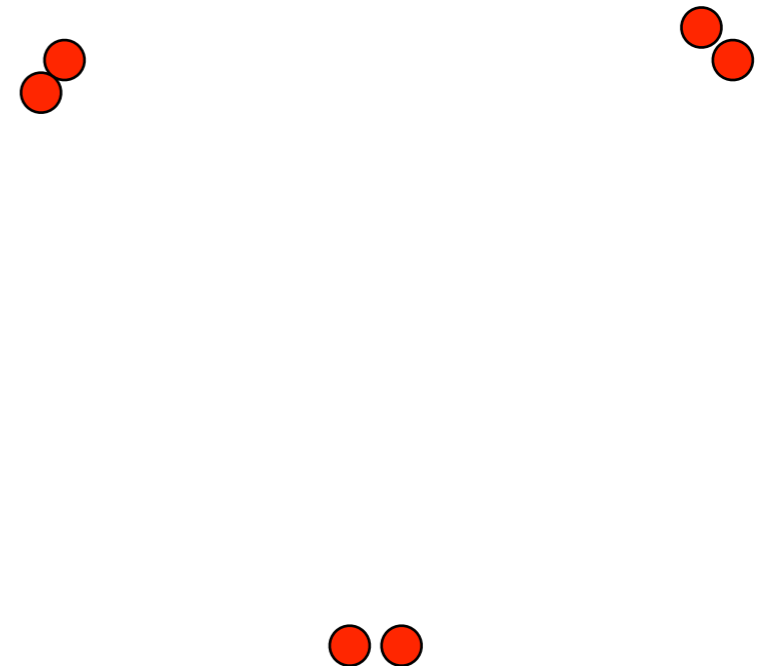
**Given:** A set  $P$  of points in  $\mathbb{R}^2$



## Problem 7.1:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

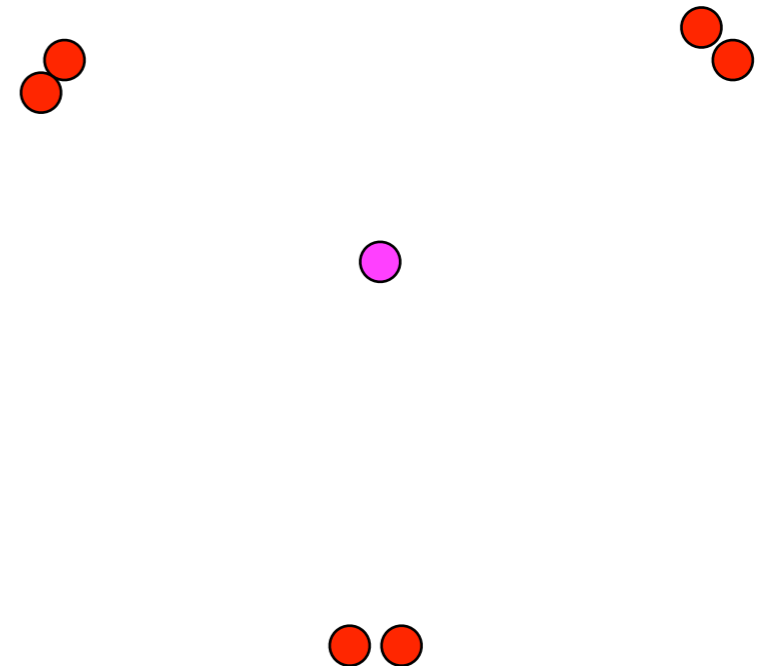
**Wanted:** A location  $c$  that minimizes the total distance to the given points



## Problem 7.1:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

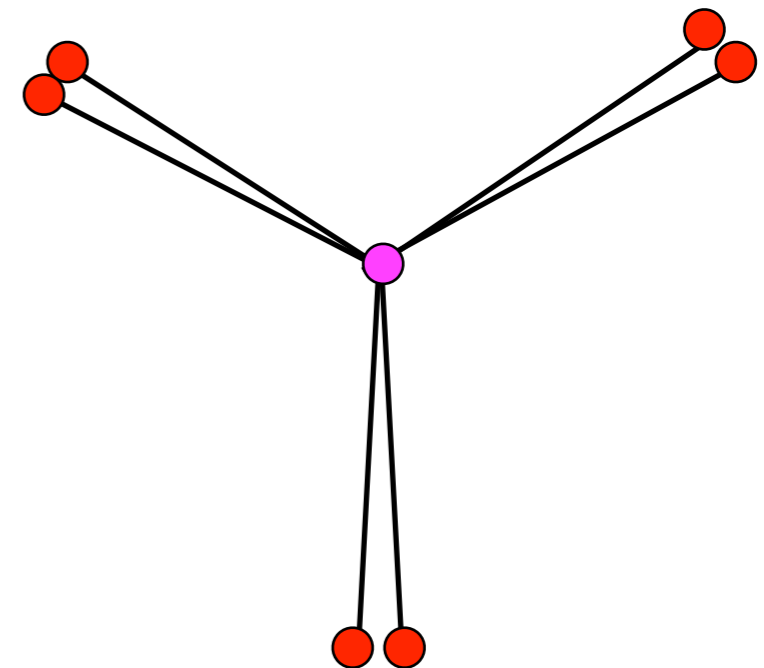
**Wanted:** A location  $c$  that minimizes the total distance to the given points



## Problem 7.1:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

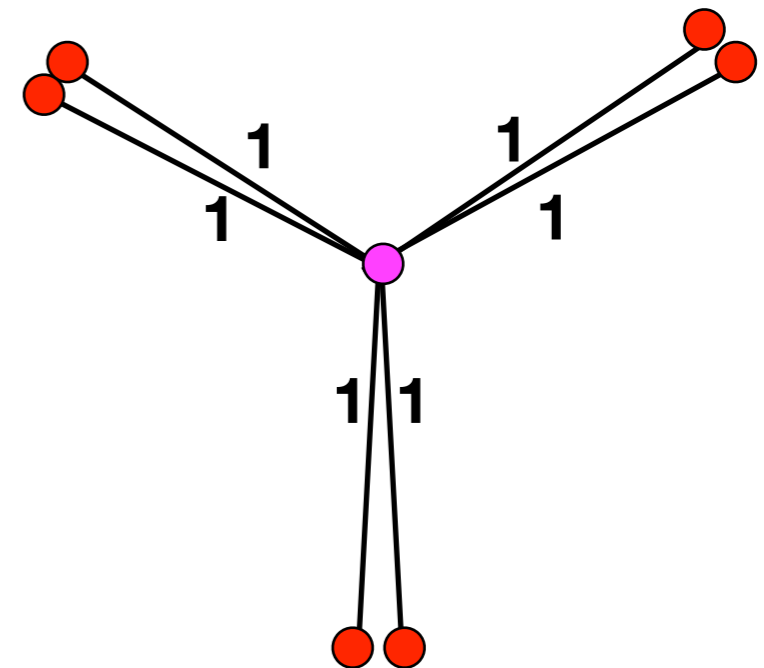
**Wanted:** A location  $c$  that minimizes the total distance to the given points



## Problem 7.1:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the total distance to the given points



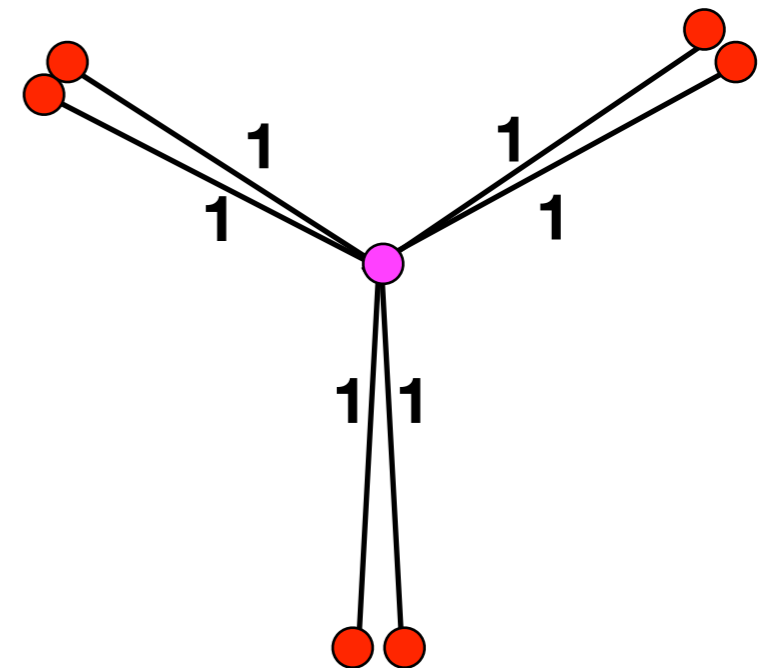


## Problem 7.1:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the total distance to the given points

## Problem 7.3:



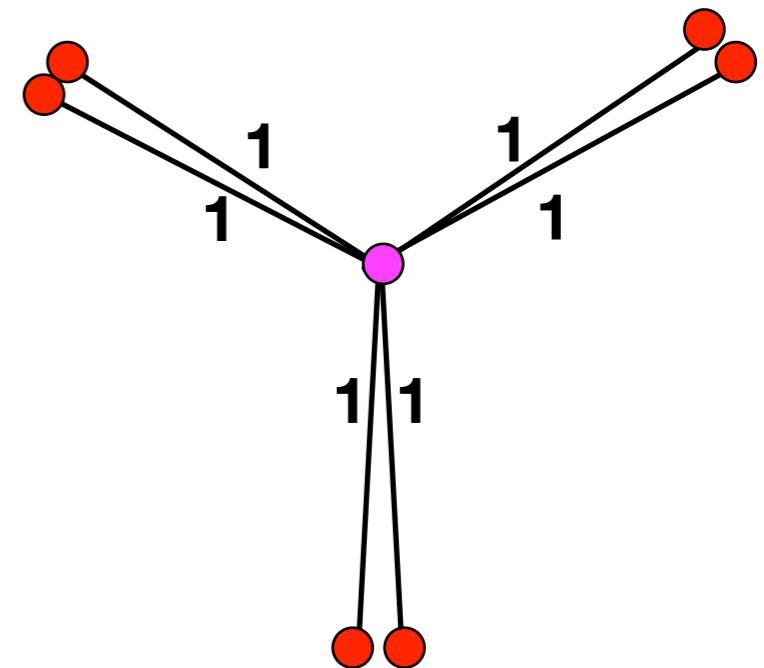
## Problem 7.1:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the total distance to the given points

## Problem 7.3:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$



## Problem 7.1:

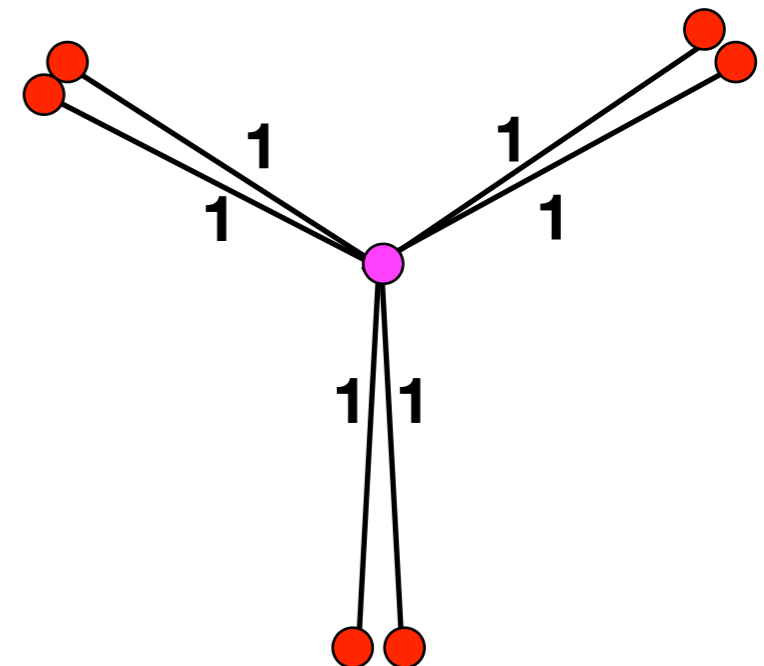
**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the total distance to the given points

## Problem 7.3:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching of the given points



## Problem 7.1:

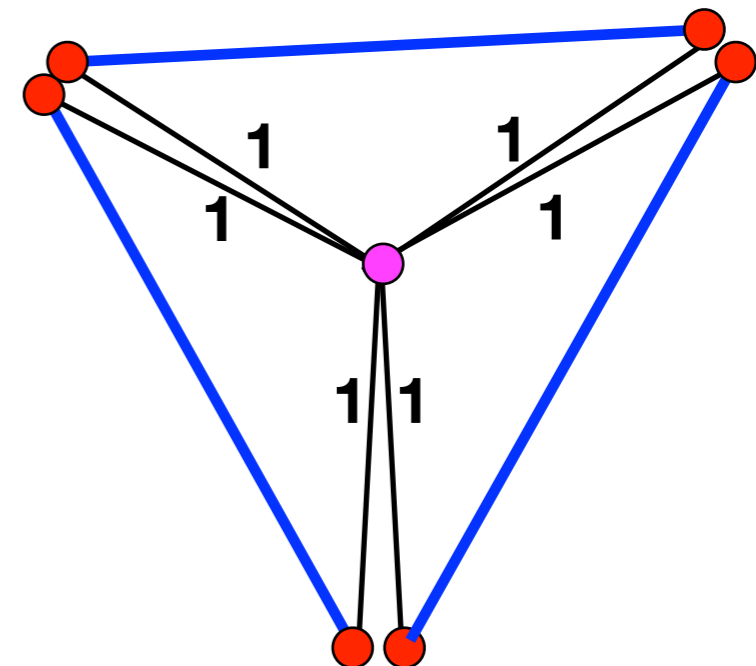
**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the total distance to the given points

## Problem 7.3:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching of the given points



## Problem 7.1:

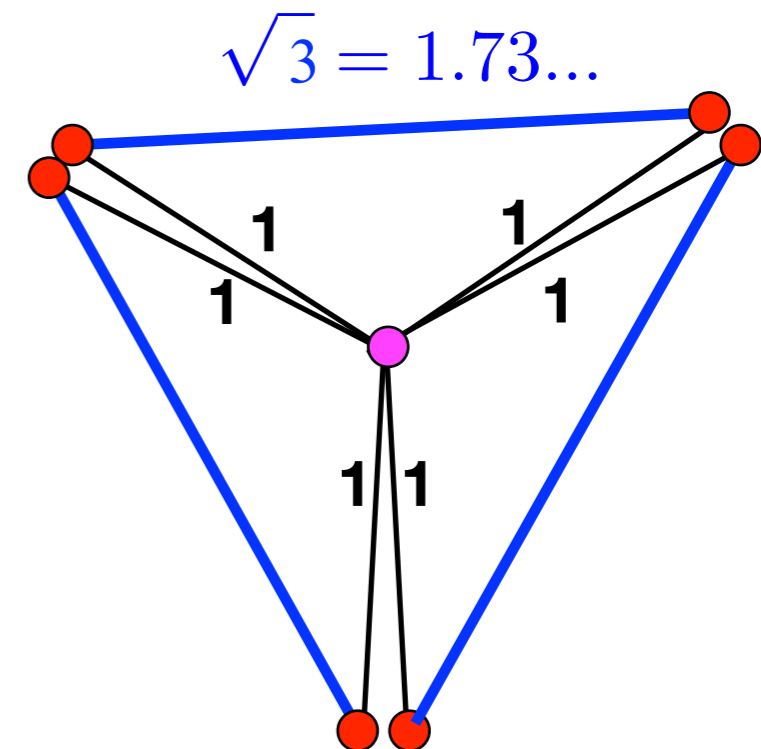
**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the total distance to the given points

## Problem 7.3:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching of the given points



## Problem 7.1:

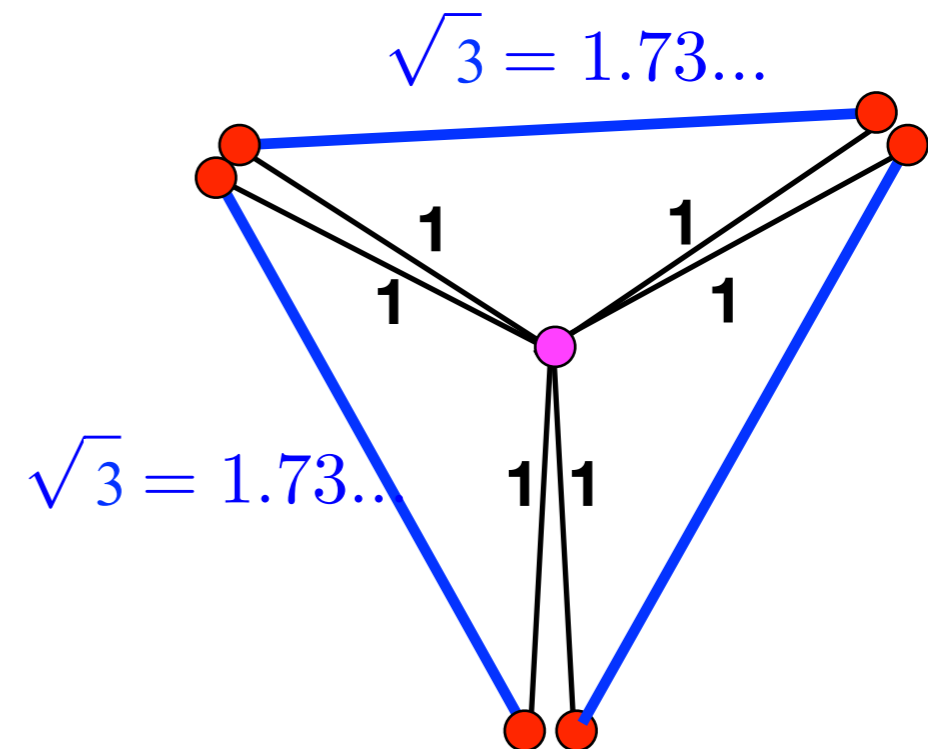
**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the total distance to the given points

## Problem 7.3:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching of the given points



## Problem 7.1:

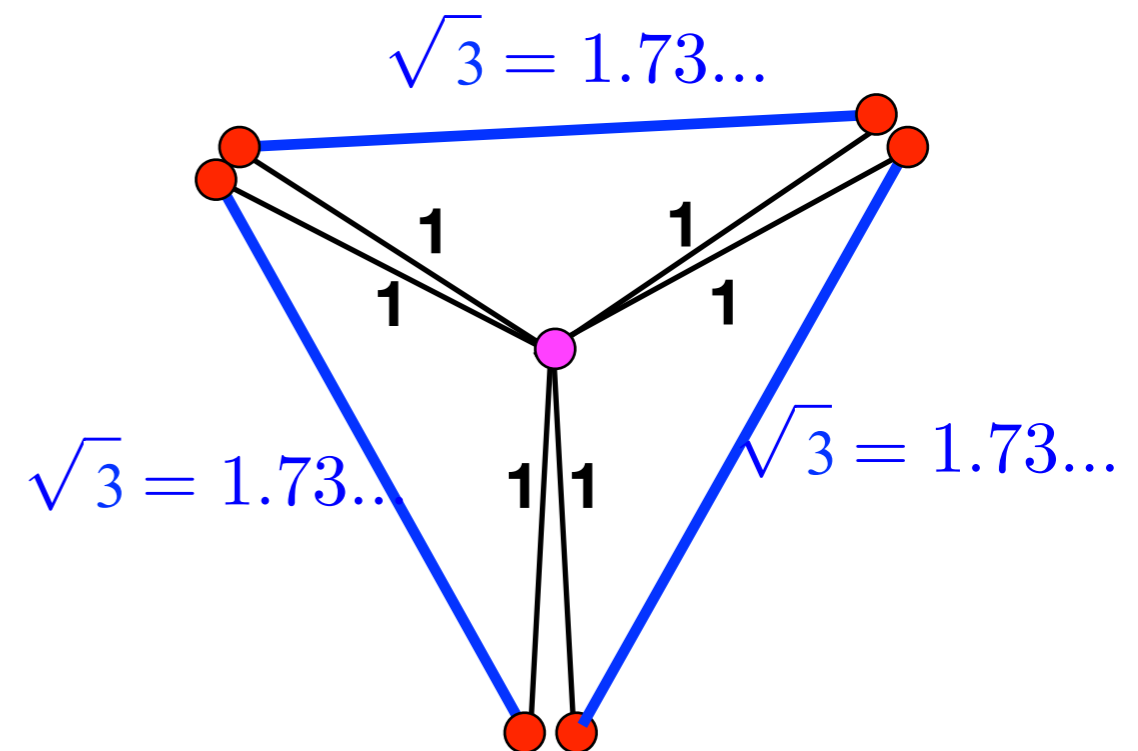
**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the total distance to the given points

## Problem 7.3:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching of the given points



## Problem 7.1:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$


**Wanted:** A location  $c$  that minimizes the total distance to the given points

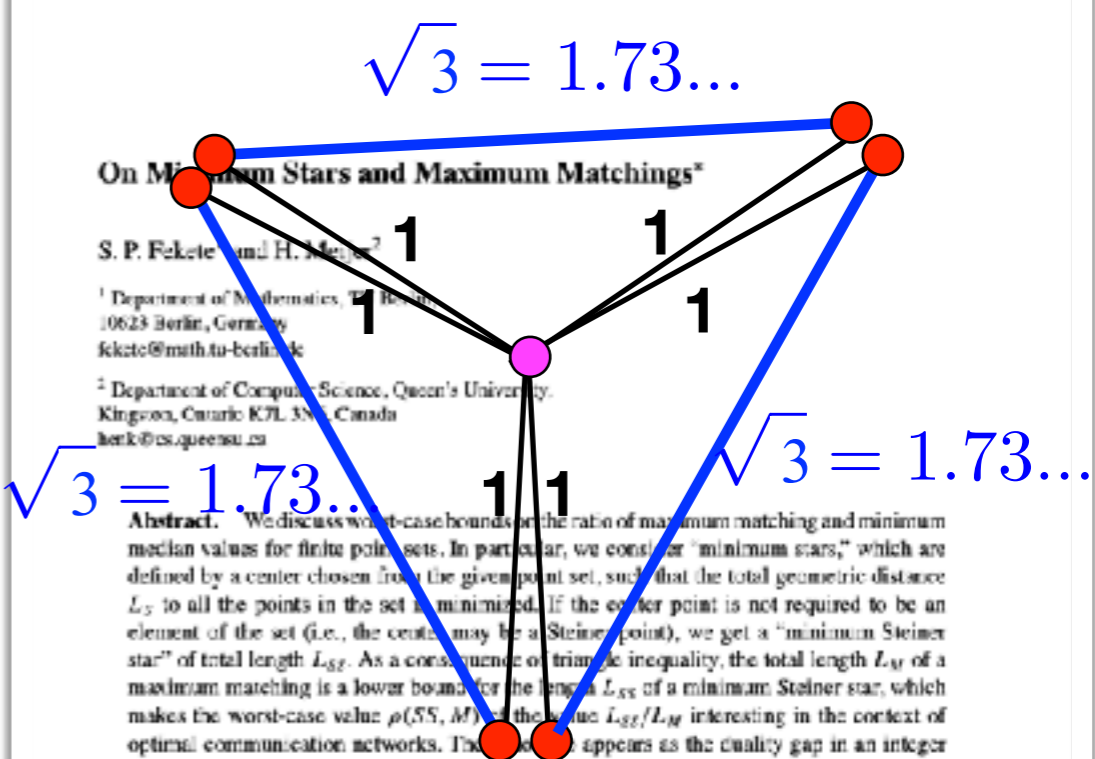
## Problem 7.3:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching of the given points

Discrete Comput Geom 23:389–407 (2000)  
DOI: 10.1007/s004540010007





$\sqrt{3} = 1.73\dots$

**On Minimum Stars and Maximum Matchings\***

S. P. Fekete<sup>1</sup> and H. Meijer<sup>2</sup>

<sup>1</sup> Department of Mathematics, TU Berlin, 10623 Berlin, Germany  
fekete@math.tu-berlin.de

<sup>2</sup> Department of Computer Science, Queen's University, Kingston, Ontario K7L 3N6, Canada  
henk@cs.queensu.ca

**Abstract.** We discuss worst-case bounds on the ratio of maximum matching and minimum median values for finite point sets. In particular, we consider “minimum stars,” which are defined by a center chosen from the given point set, such that the total geometric distance  $L_S$  to all the points in the set is minimized. If the center point is not required to be an element of the set (i.e., the center may be a Steiner point), we get a “minimum Steiner star” of total length  $L_{SS}$ . As a consequence of triangle inequality, the total length  $L_M$  of a maximum matching is a lower bound for the length  $L_{SS}$  of a minimum Steiner star, which makes the worst-case value  $\rho(S, M)$  of the ratio  $L_{SS}/L_M$  interesting in the context of optimal communication networks. This ratio appears as the duality gap in an integer programming formulation of a location problem by Tamir and Mitchell.

In this paper we show that for a finite set that consists of an even number of points in the plane and Euclidean distances, the worst-case ratio  $\rho(S, M)$  cannot exceed  $2/\sqrt{3}$ . This proves a conjecture of Suri, who gave an example where this bound is achieved. For the case of Euclidean distances in two and three dimensions, we also prove upper and lower bounds for the worst-case value  $\rho(S, SS)$  of the ratio  $L_S/L_{SS}$ , and for the worst-case value  $\rho(S, M)$  of the ratio  $L_S/L_M$ . We give tight upper bounds for the case where distances are measured according to the Manhattan metric: we show that in three-dimensional space,  $\rho(SS, M)$  is bounded by  $\frac{3}{2}$ , while in two-dimensional space  $L_{SS} = L_M$ , extending some independent observations by Tamir and Mitchell. Finally, we show that  $\rho(S, SS)$  is  $\frac{3}{2}$  in the two-dimensional case, and  $\frac{3}{2}$  in the three-dimensional case.

\* Parts of this work were done while the first author was visiting Queen's University, partially supported by the Deutsche Forschungsgemeinschaft, BE-407/3-1. Parts of this work were done while the second author was visiting Universität zu Köln, partially supported by NSERC. A preliminary extended abstract of this article, titled “On minimum stars, minimum Steiner stars, and maximum matchings,” appears in the *Proceedings of the 15th ACM Symposium on Computational Geometry* [9].



## Problem 7.1:

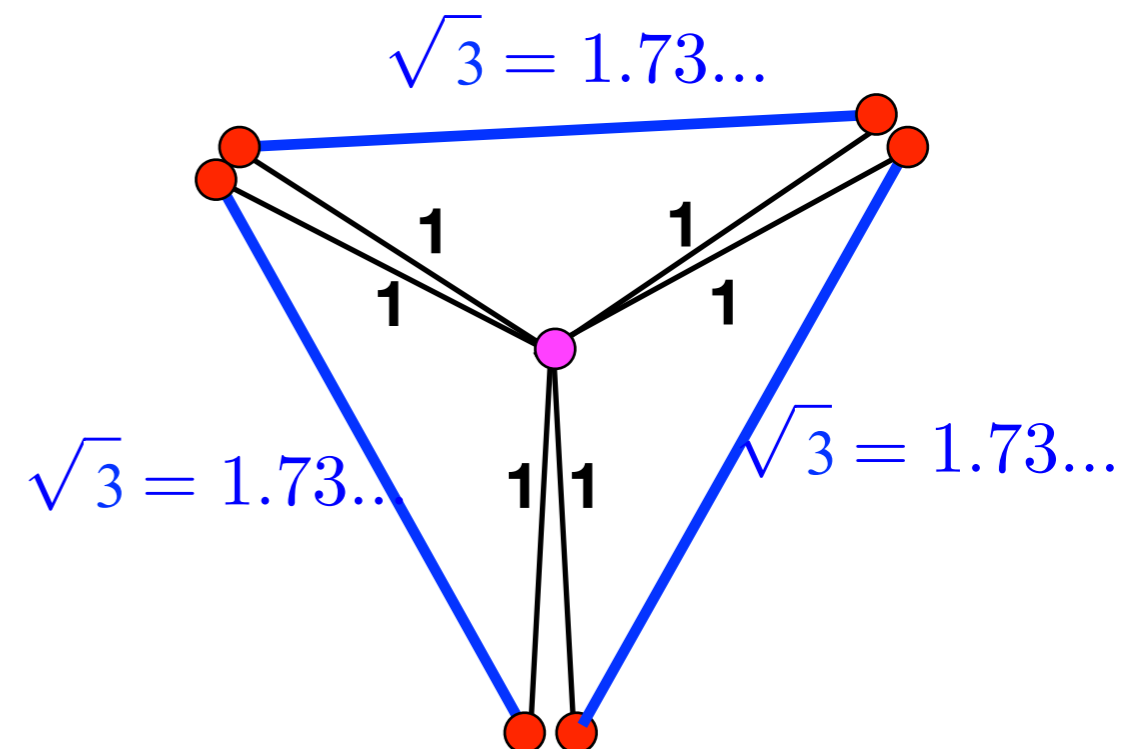
**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the total distance to the given points

## Problem 7.3:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching of the given points



## Problem 7.1:

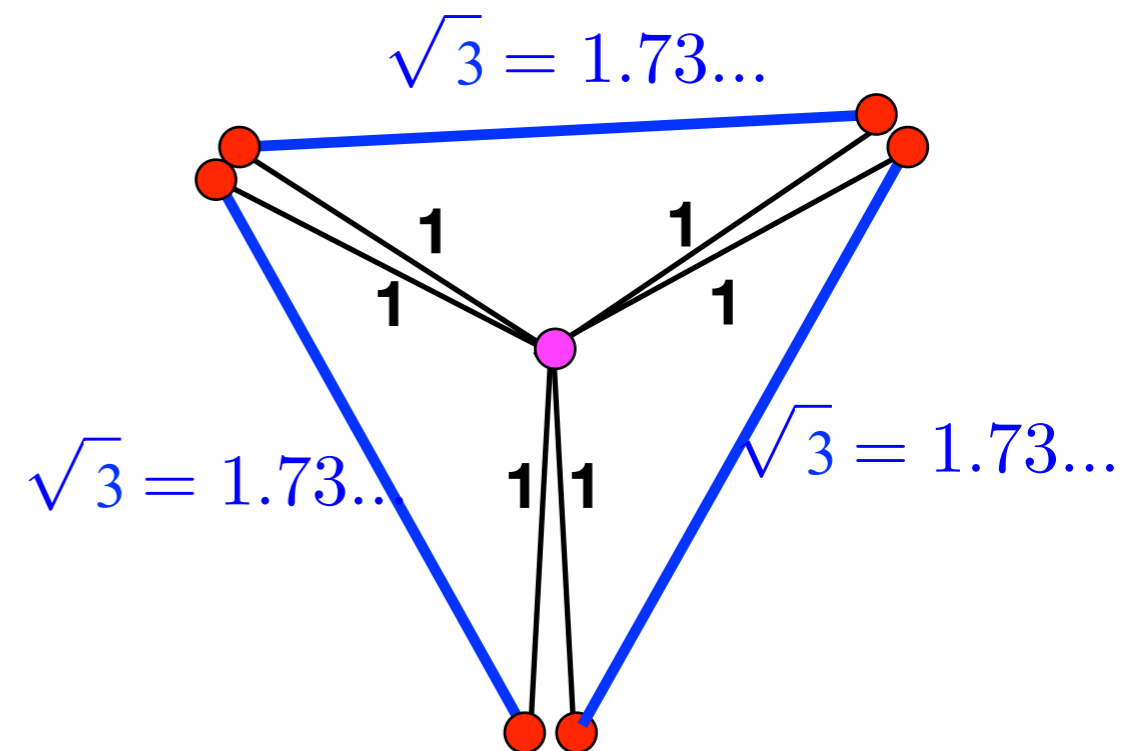
**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the total distance to the given points

## Problem 7.3:

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching of the given points



$$2/\sqrt{3} = 1.15\dots$$



**Proposition 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .*

**Proposition 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .*

**Theorem 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .*

**Proposition 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .*

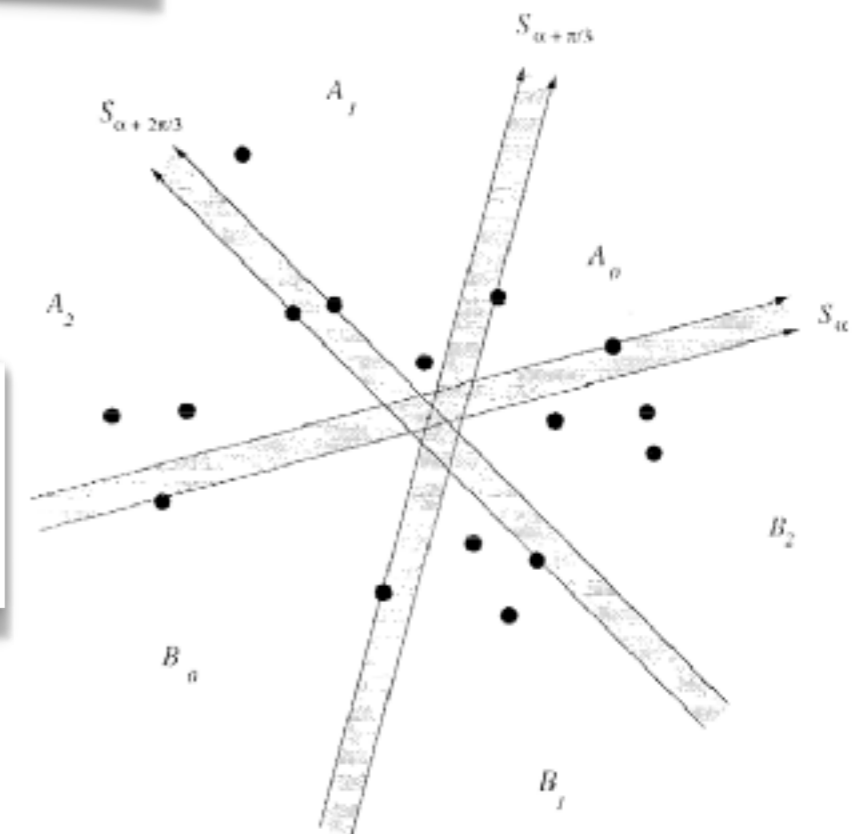
**Theorem 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .*

**Lemma 2.** *For point sets  $P$  in two-dimensional space we can find three directed lines  $l_0, l_1,$  and  $l_2$  such that the three lines intersect in a common point, all three lines are halving lines of  $P$  and the smallest angle between any two lines is  $\pi/3$ .*

**Proposition 1.** For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .

**Theorem 1.** For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .

**Lemma 2.** For point sets  $P$  in two-dimensional space we can find three directed lines  $l_0, l_1,$  and  $l_2$  such that the three lines intersect in a common point, all three lines are halving lines of  $P$  and the smallest angle between any two lines is  $\pi/3$ .



**Fig. 1.** Finding a small Steiner star and a large matching.

**Proposition 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .*

**Theorem 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .*



**Proposition 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .*

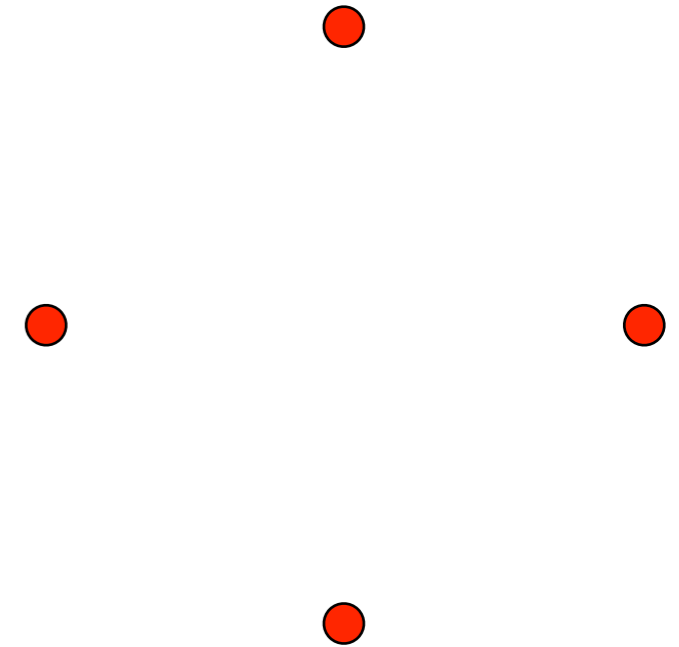
**Theorem 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .*

**Theorem 6.** *For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .*

**Proposition 1.** For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .

**Theorem 1.** For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .

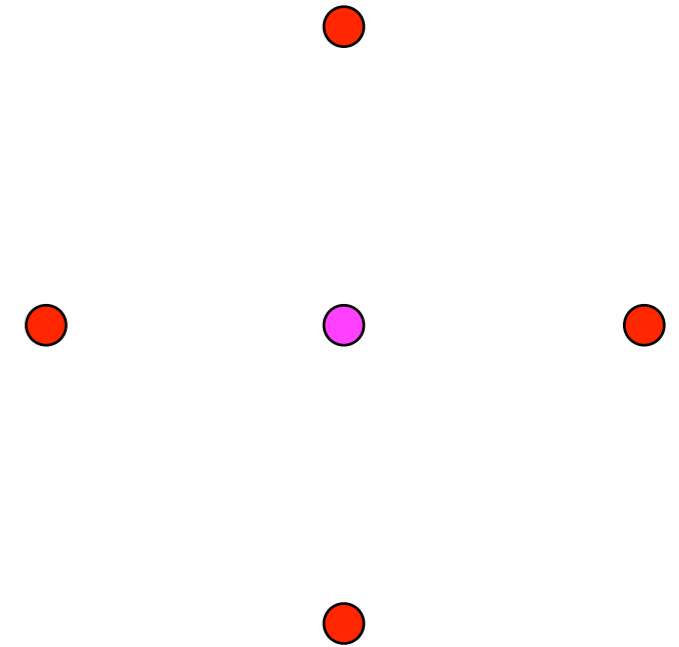
**Theorem 6.** For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .



**Proposition 1.** For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .

**Theorem 1.** For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .

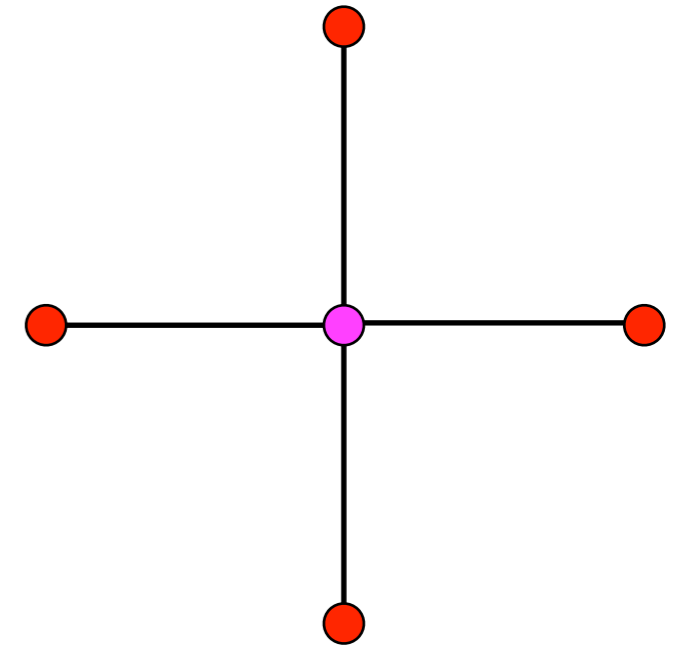
**Theorem 6.** For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .



**Proposition 1.** For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .

**Theorem 1.** For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .

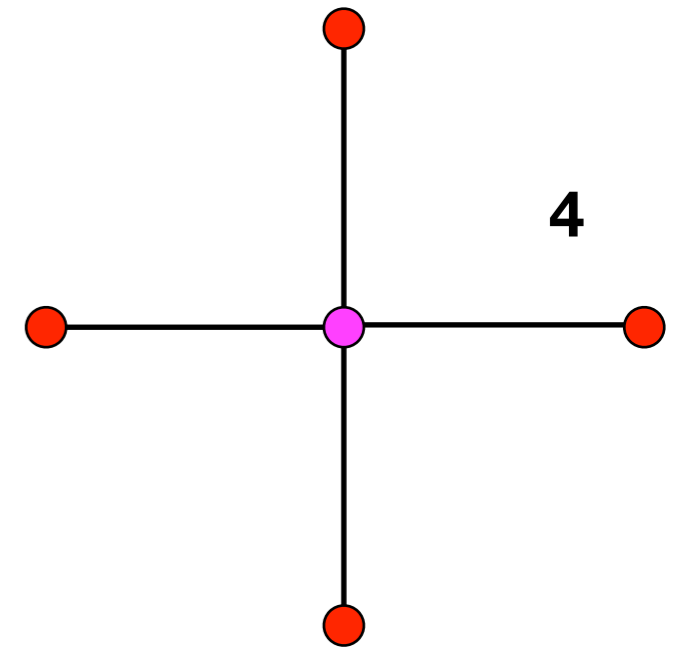
**Theorem 6.** For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .



**Proposition 1.** For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .

**Theorem 1.** For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .

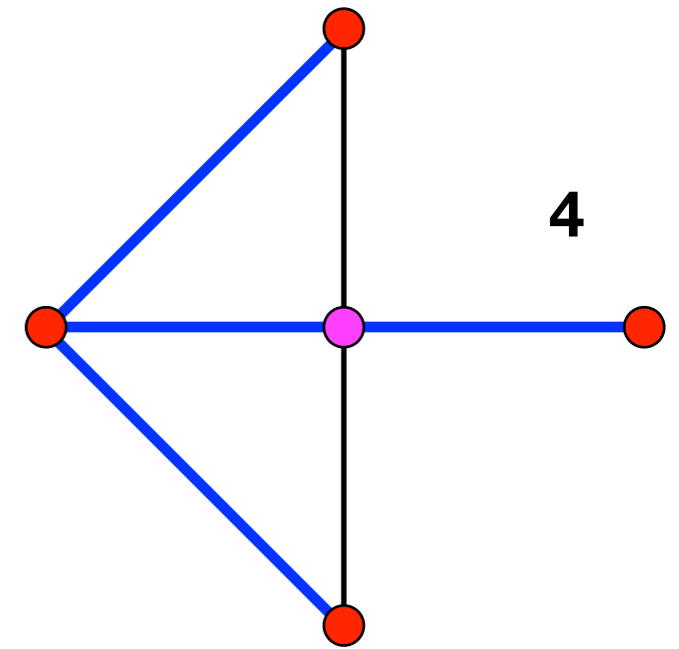
**Theorem 6.** For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .



**Proposition 1.** For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .

**Theorem 1.** For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .

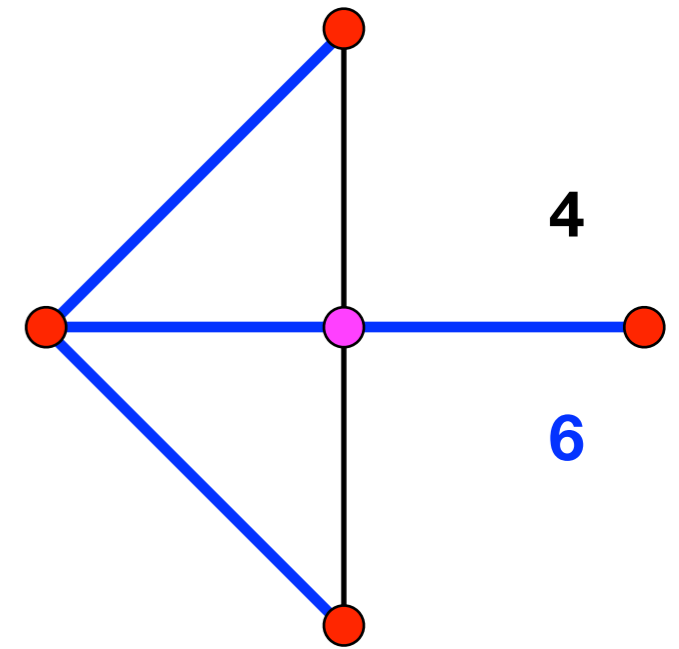
**Theorem 6.** For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .



**Proposition 1.** For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .

**Theorem 1.** For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .

**Theorem 6.** For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .



**Proposition 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .*

**Theorem 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .*

**Theorem 6.** *For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .*



**Proposition 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .*

**Theorem 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .*

**Theorem 6.** *For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .*

**Theorem 7.** *For point sets  $P$  of even cardinality in three-dimensional space with Manhattan distances, we have  $\rho(SS, M) = \frac{3}{2}$ .*

**Proposition 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .*

**Theorem 1.** *For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .*

**Theorem 6.** *For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .*

**Theorem 7.** *For point sets  $P$  of even cardinality in three-dimensional space with Manhattan distances, we have  $\rho(SS, M) = \frac{3}{2}$ .*

**Theorem 8.** *For point sets  $P$  in three-dimensional space with Manhattan distances, we have  $\rho(SS, S) = \frac{5}{3}$ .*

**Proposition 1.** For point sets  $P$  of even cardinality in two-dimensional space with Manhattan distances, we have  $L_M = L_{SS}$ .

**Theorem 1.** For point sets  $P$  of even cardinality in two-dimensional space with Euclidean distances, we have  $\rho(SS, M) = 2/\sqrt{3}$ .

**Theorem 6.** For point sets  $P$  in two-dimensional space with Manhattan distances, we have  $\rho(S, SS) = \frac{3}{2}$ .

**Theorem 7.** For point sets  $P$  of even cardinality in three-dimensional space with Manhattan distances, we have  $\rho(SS, M) = \frac{3}{2}$ .

**Theorem 8.** For point sets  $P$  in three-dimensional space with Manhattan distances, we have  $\rho(SS, S) = \frac{5}{3}$ .

**Conjecture 1.** For point sets  $P$  in two-dimensional space with Euclidean distances, we have  $\rho(S, SS) = 4/\pi$ .

**Conjecture 2.** For point sets  $P$  in three-dimensional space with Euclidean distances, we have  $\rho(S, SS) = \frac{4}{3}$ .

**Table 1.** Lower and upper bounds for worst-case ratios.

Distances	Dimension	Ratio	Lower bound	Upper bound
Euclidean	Two	$\rho(SS, M)$	$\frac{2}{\sqrt{3}} = 1.15 \dots$	$\frac{2}{\sqrt{3}} = 1.15 \dots$
		$\rho(S, SS)$	$\frac{4}{\pi} = 1.27 \dots$	$\sqrt{2} = 1.41 \dots$
		$\rho(S, M)$	$\frac{4}{3} = 1.33 \dots$	$\frac{2\sqrt{2}}{\sqrt{3}} = 1.63 \dots$
	Three	$\rho(SS, M)$	$\frac{\sqrt{3}}{\sqrt{2}} = 1.22 \dots$	$\sqrt{2} = 1.41 \dots$
		$\rho(S, SS)$	$\frac{4}{3} = 1.33 \dots$	$\sqrt{2} = 1.41 \dots$
		$\rho(S, M)$	$\frac{3}{2} = 1.5$	2
Manhattan	Two	$\rho(SS, M)$	1	1
		$\rho(S, SS)$	$\frac{3}{2} = 1.5$	$\frac{3}{2} = 1.5$
		$\rho(S, M)$	$\frac{3}{2} = 1.5$	$\frac{3}{2} = 1.5$
	Three	$\rho(SS, M)$	$\frac{3}{2} = 1.5$	$\frac{3}{2} = 1.5$
		$\rho(S, SS)$	$\frac{5}{3} = 1.66 \dots$	$\frac{5}{3} = 1.66 \dots$
		$\rho(S, M)$	$\frac{5}{3} = 1.66 \dots$	2

# Solving a „hard“ problem to approximate an „easy“ one [Fekete, Meijer, Rohe, Tietze 2002]



**Problem 7.3:**

**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching  
of the given points



**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching  
of the given points

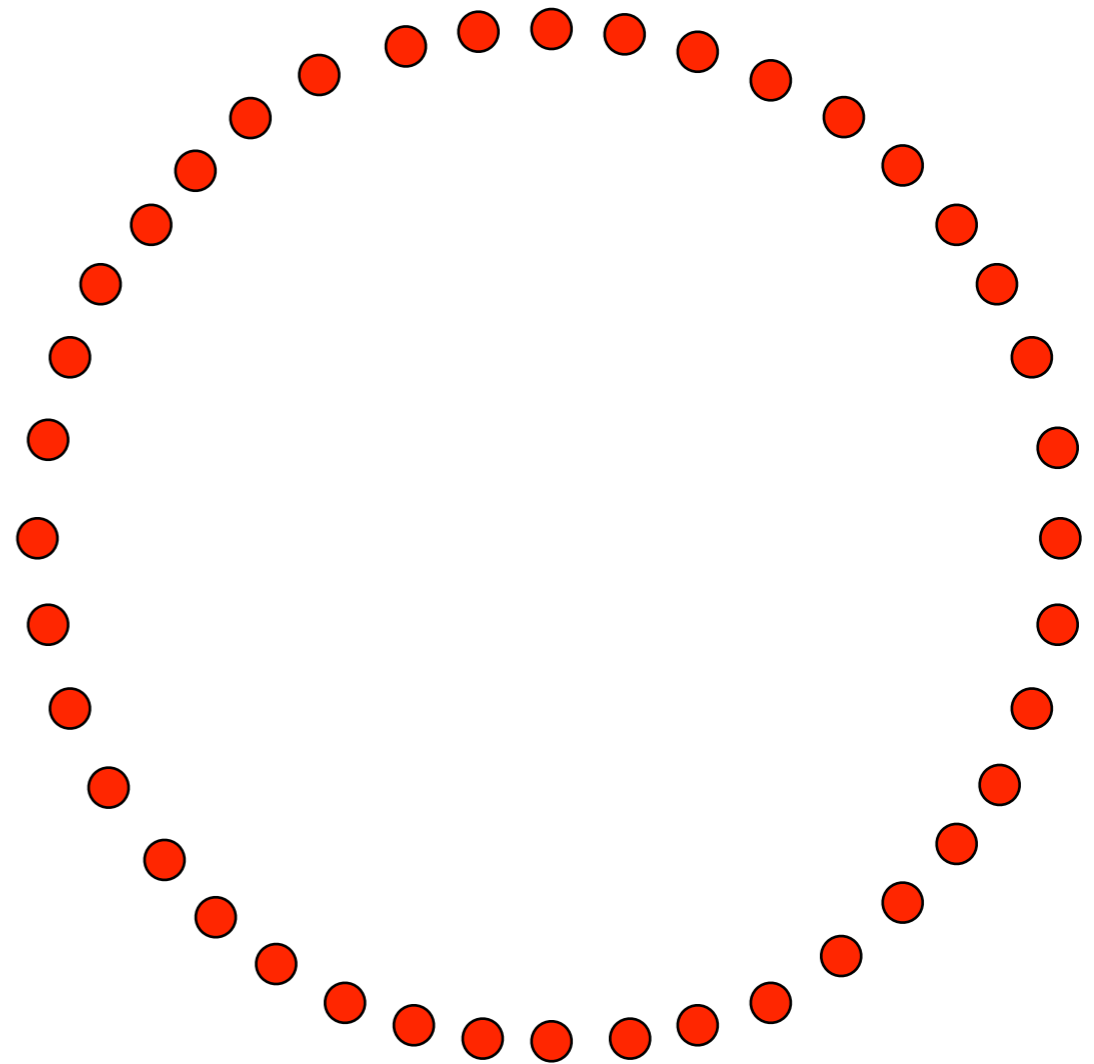
$$O(n^4)$$

**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching  
of the given points

$$O(n^4)$$



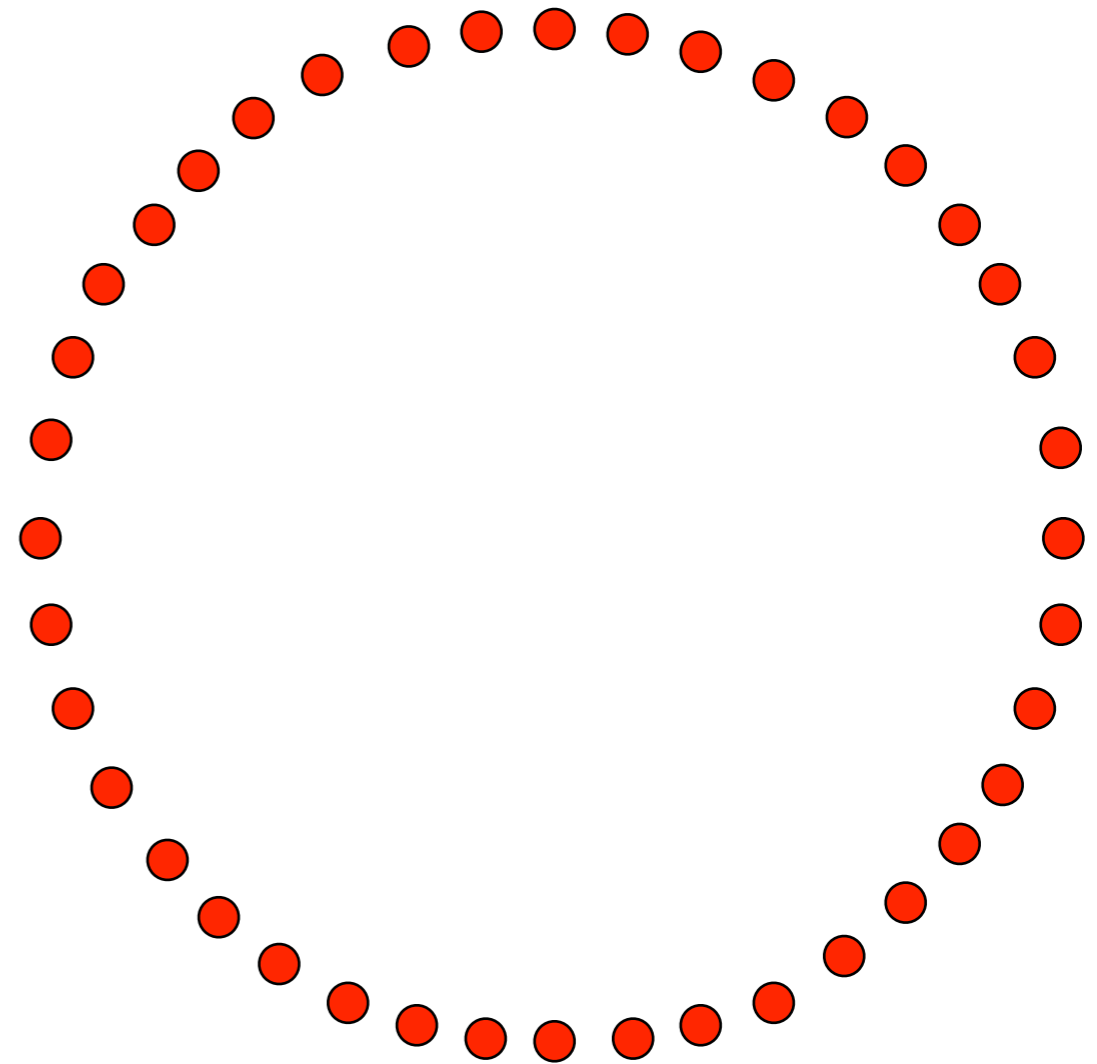
**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching  
of the given points

$$O(n^4)$$

**Problem 7.1:**



**Problem 7.3:**

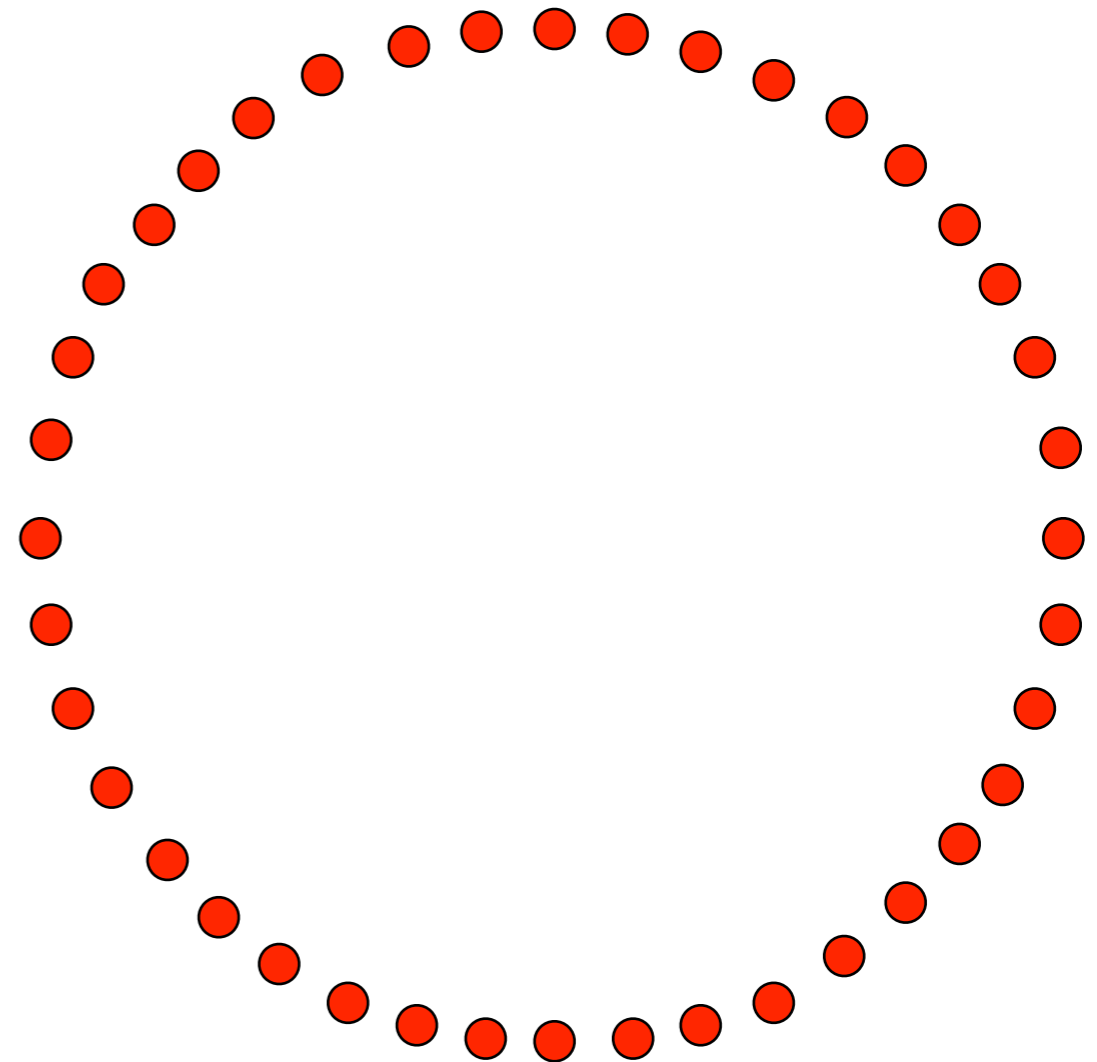
**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching  
of the given points

$$O(n^4)$$

**Problem 7.1:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$



**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

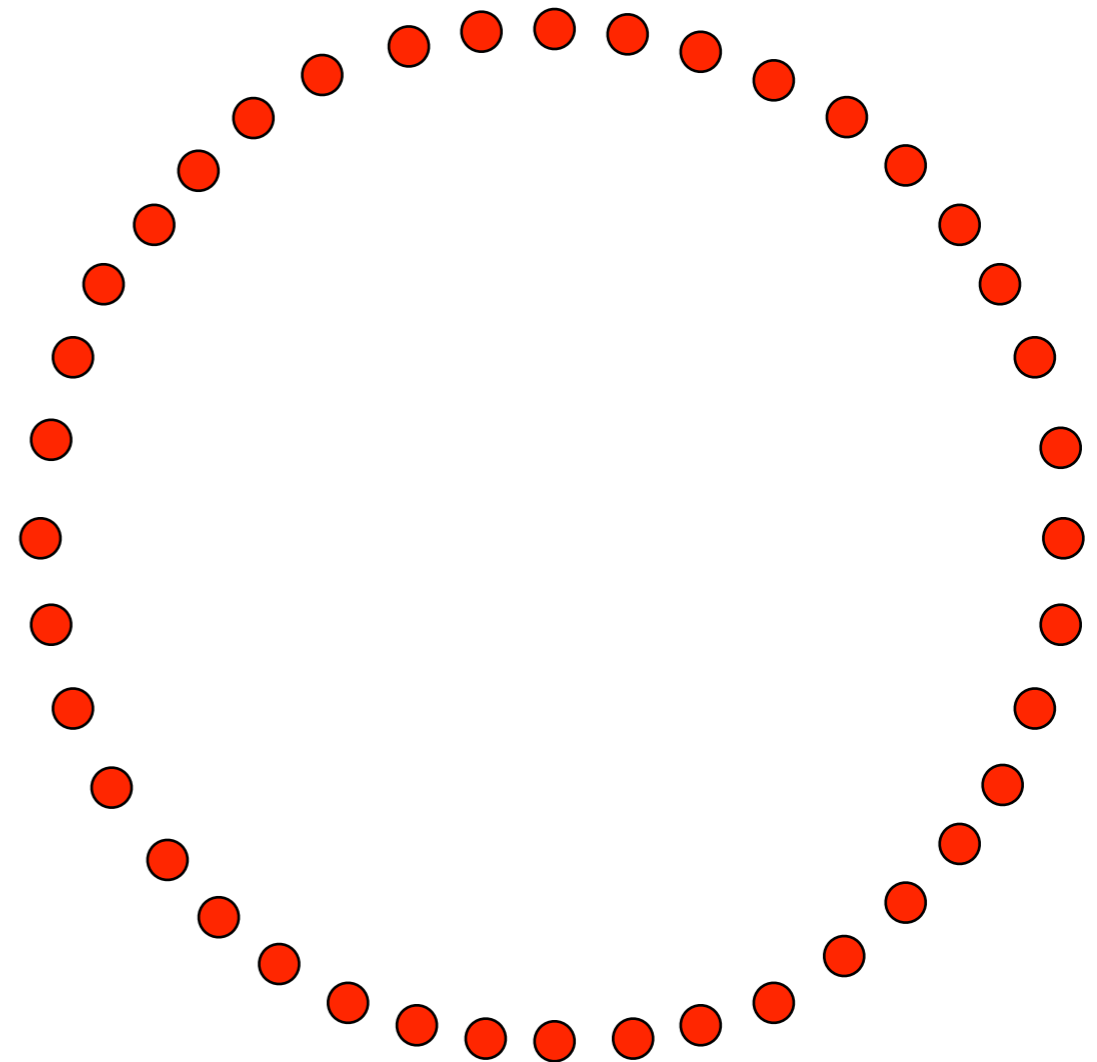
**Wanted:** A maximum-weight perfect matching  
of the given points

$$O(n^4)$$

**Problem 7.1:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the  
total distance to the given points



**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

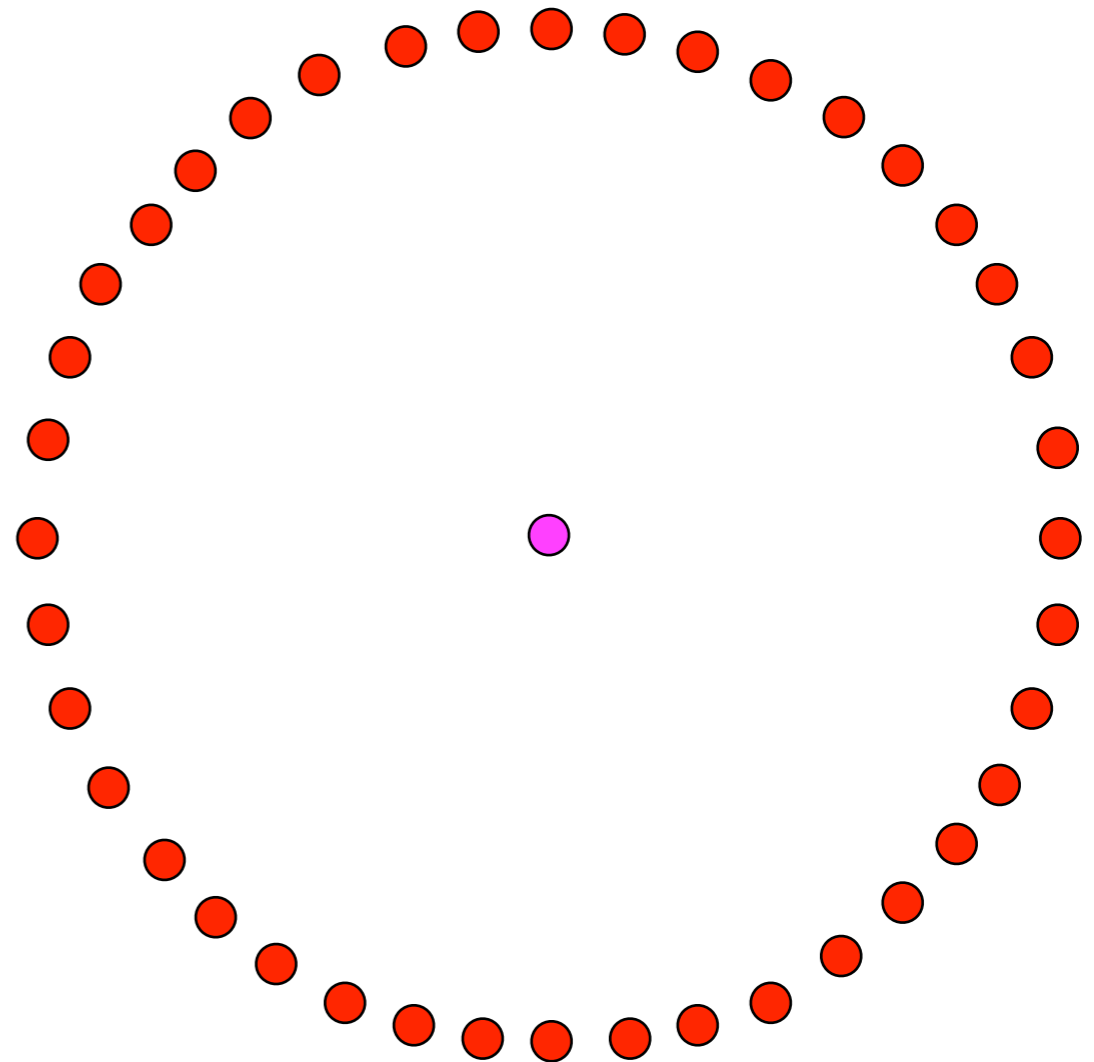
**Wanted:** A maximum-weight perfect matching  
of the given points

$$O(n^4)$$

**Problem 7.1:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the  
total distance to the given points



**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

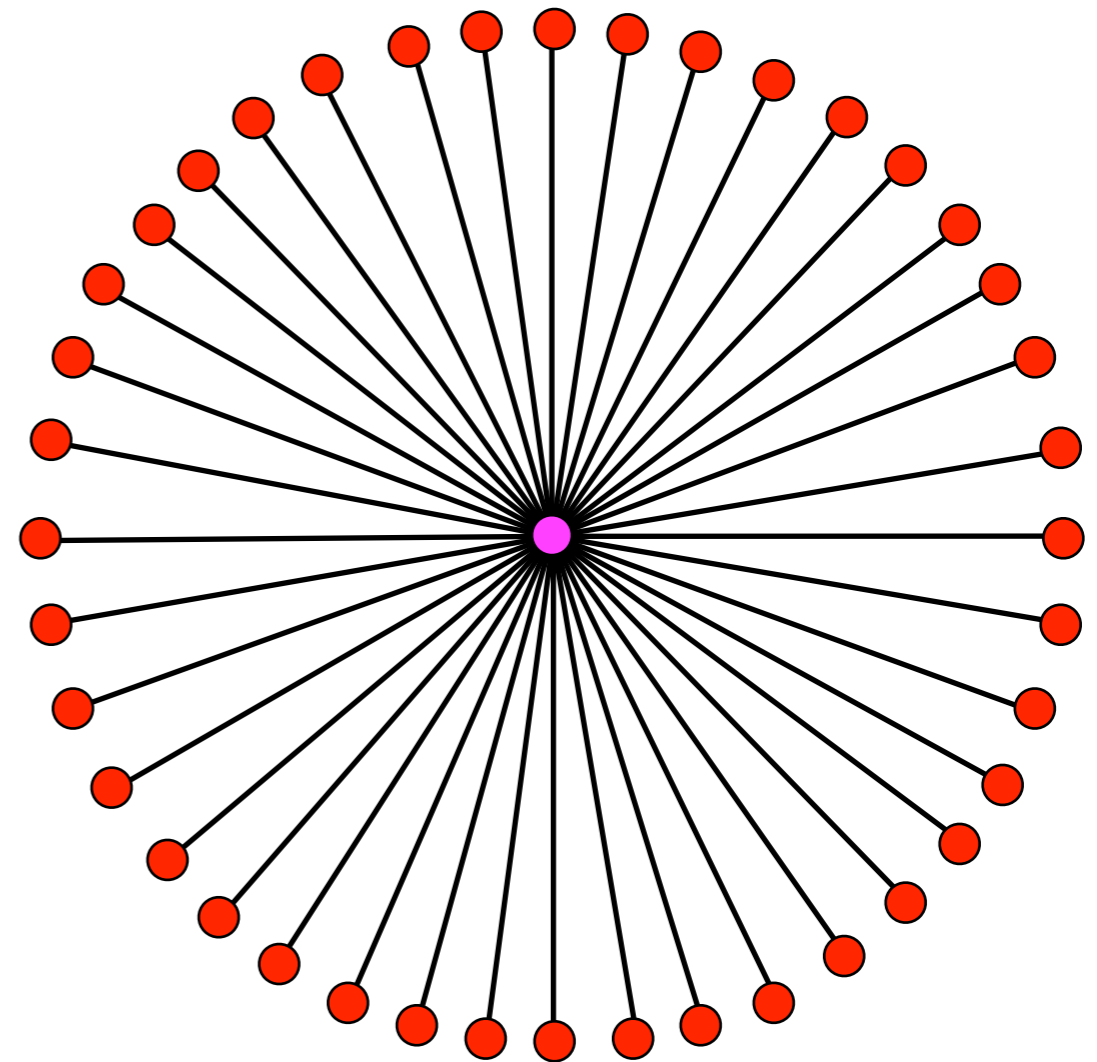
**Wanted:** A maximum-weight perfect matching  
of the given points

$$O(n^4)$$

**Problem 7.1:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the  
total distance to the given points



# Solving a „hard“ problem to approximate an „easy“ one [Fekete, Meijer, Rohe, Tietze 2002]

Discrete Comput Geom 3:177-191 (1988)

Discrete & Computational  
**Geometry**  
© 1988 Springer-Verlag New York Inc.

## The Algebraic Degree of Geometric Optimization Problems

Chanderjit Bajaj

Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA

**Abstract.** In this paper we apply Galois methods to certain fundamental *geometric optimization* problems whose exact computational complexity has been an open problem for a long time. In particular we show that the classic Weber problem, along with the *line-restricted* Weber problem and its *three-dimensional* version are in general not solvable by radicals over the field of rationals. One direct consequence of these results is that for these geometric optimization problems there exists *no exact algorithm* under models of computation where the root of an algebraic equation is obtained using arithmetic operations and the extraction of *k*th roots. This leaves only numerical or symbolic approximations to the solutions, where the complexity of the approximations is shown to be primarily a function of the algebraic degree of the optimum solution point.

### 1. Introduction

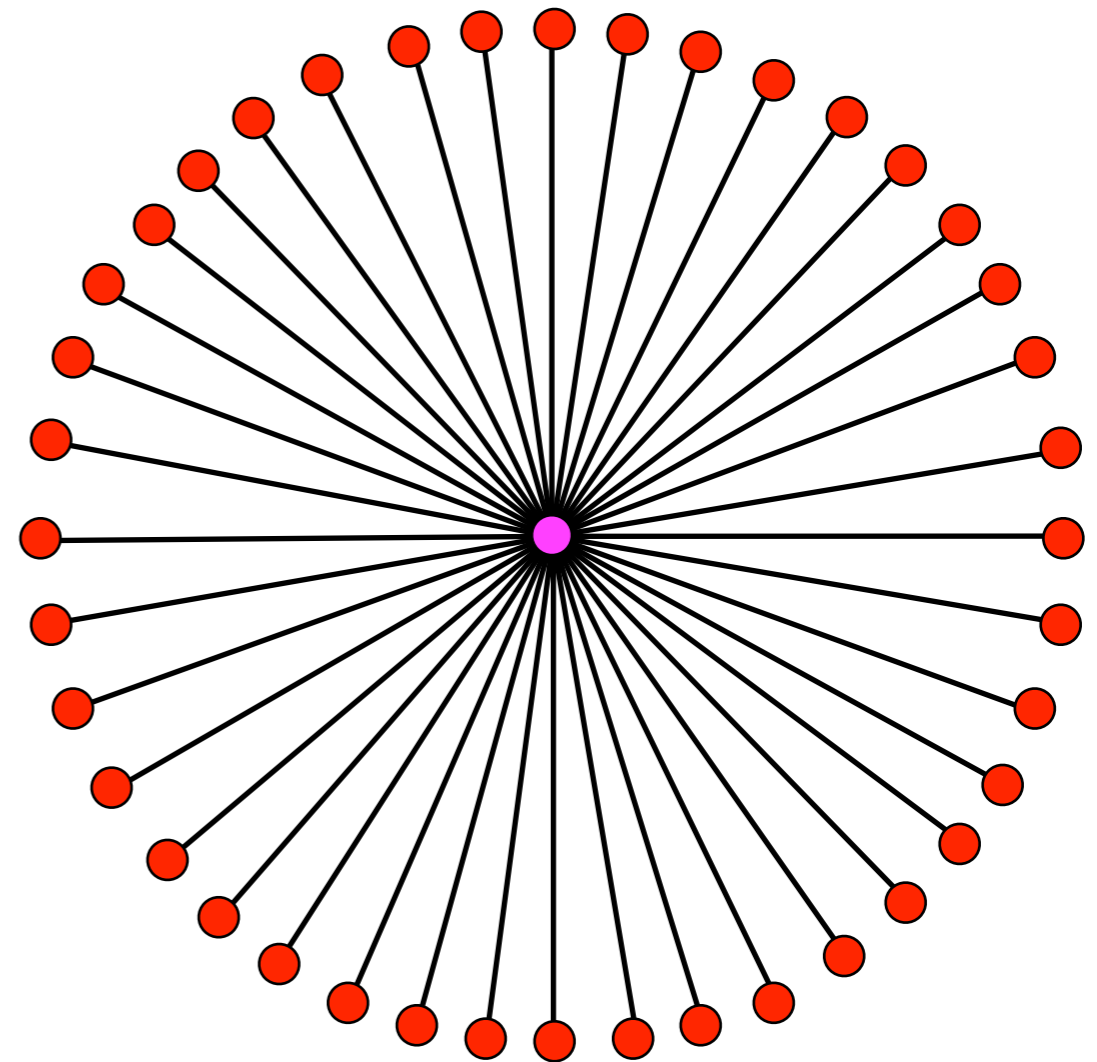
Geometric optimization problems are inherently not pure combinatorial problems since the optimal solution often belongs to an infinite feasible set, the entire real Euclidean space. Such problems frequently arise in computer-aided design and robotics. It has thus become increasingly important to devise appropriate methods to analyze the complexity of problems where combinatorial analysis methods seem to fail. Here we take a step in this direction by applying Galois algebraic methods to certain fundamental *geometric optimization* problems. These problems are noncombinatorial and have no known polynomial time solutions. Neither have these problems shown to be intractable (NP-hard, etc.). In fact, the recognition versions of these optimization problems are not even known to be in the class NP [10].

The use of algebraic methods for analyzing the complexity of geometric problems has been popular since the time of Descartes, Gauss, Abel, and Galois. The complexity of straight-edge and compass constructions has been known to

matching

$$O(n^4)$$

the  
oints





# „hard“ problem to approximate an „easy“ one [Fekete, Meijer, Rohe, Tietze 2002]

## Solving a “Hard” Problem to Approximate an “Easy” One: Heuristics for Maximum Matchings and Maximum Traveling Salesman Problems

Sándor P. Fekete

Department of Mathematical Optimization

TU Braunschweig

38106 Braunschweig, GERMANY

s.fekete@tu-bs.de

and

Henk Meijer

Department of Computing and Information Science

Queen's University

Kingston, Ontario K7L 3N6, CANADA

henk@cs.queensu.ca

and

André Rohe

Forschungsinstitut für

Diskrete Mathematik

Universität Bonn

53113 Bonn, GERMANY

rohe@or.uni-bonn.de

and

Walter Tietze

Department of Mathematics

TU Berlin

10623 Berlin, GERMANY

tietze@math.tu-berlin.de

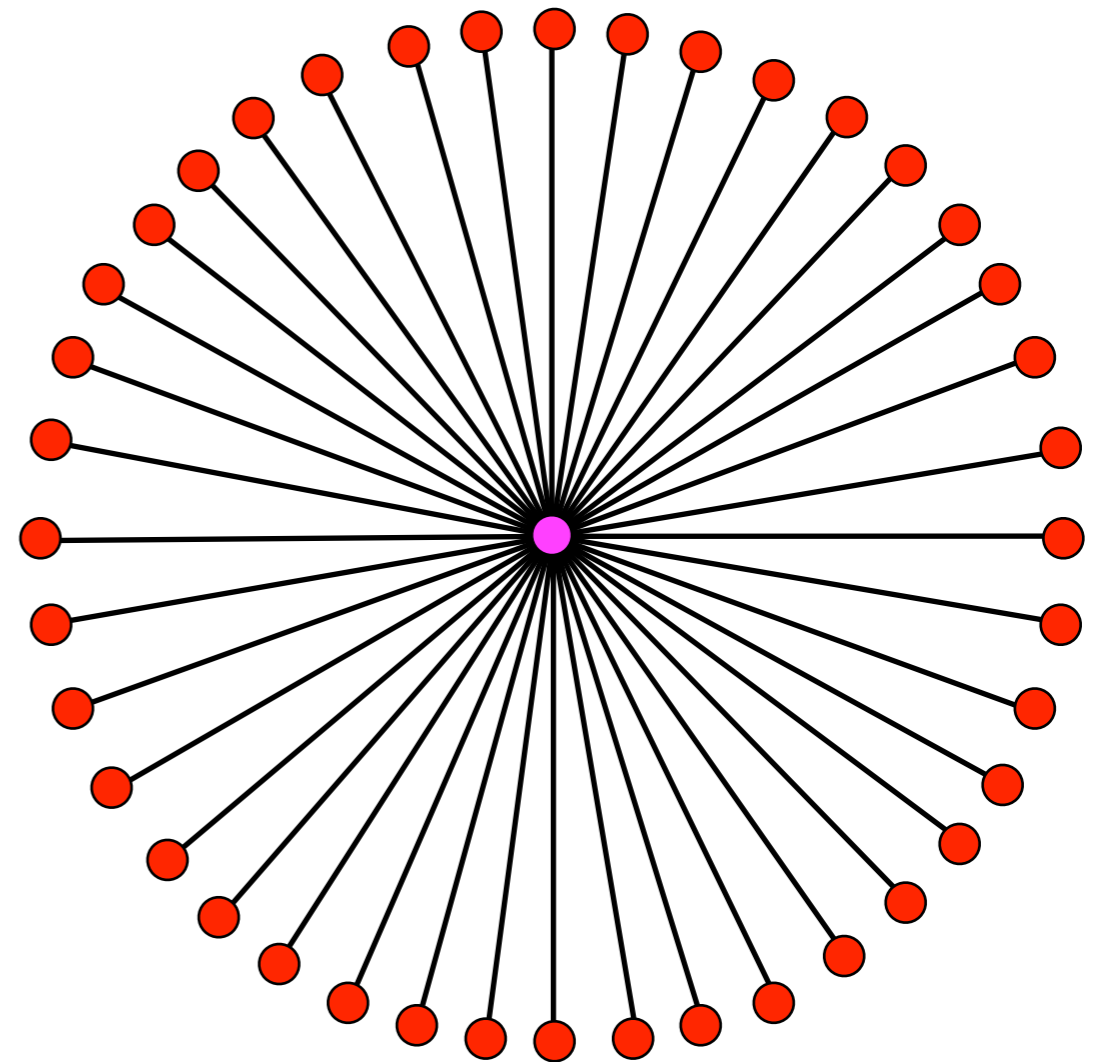
An extended abstract appears in the proceedings of ALENEX'01 [Fekete et al. 2001].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works, requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept, ACM Inc., 1515 Broadway, New York, NY 10036 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

matching

$$O(n^4)$$

the  
oints



# Solving a “Hard” Problem to Approximate an “Easy” One: Heuristics for Maximum Matchings and Maximum Traveling Salesman Problems

Sándor P. Fekete

Department of Mathematical Optimization

TU Braunschweig

38106 Braunschweig, GERMANY

s.fekete@tu-bs.de

and

Henk Meijer

Department of Computing and Information Science

Queen's University

Kingston, Ontario K7L 3N6, CANADA

henk@cs.queensu.ca

and

André Rohe

Forschungsinstitut für

Diskrete Mathematik

Universität Bonn

53113 Bonn, GERMANY

rohe@or.uni-bonn.de

and

Walter Tietze

Department of Mathematics

TU Berlin

10623 Berlin, GERMANY

tietze@math.tu-berlin.de

An extended abstract appears in the proceedings of ALENEX'01 [Fekete et al. 2001].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works, requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept, ACM Inc., 1515 Broadway, New York, NY 10036 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

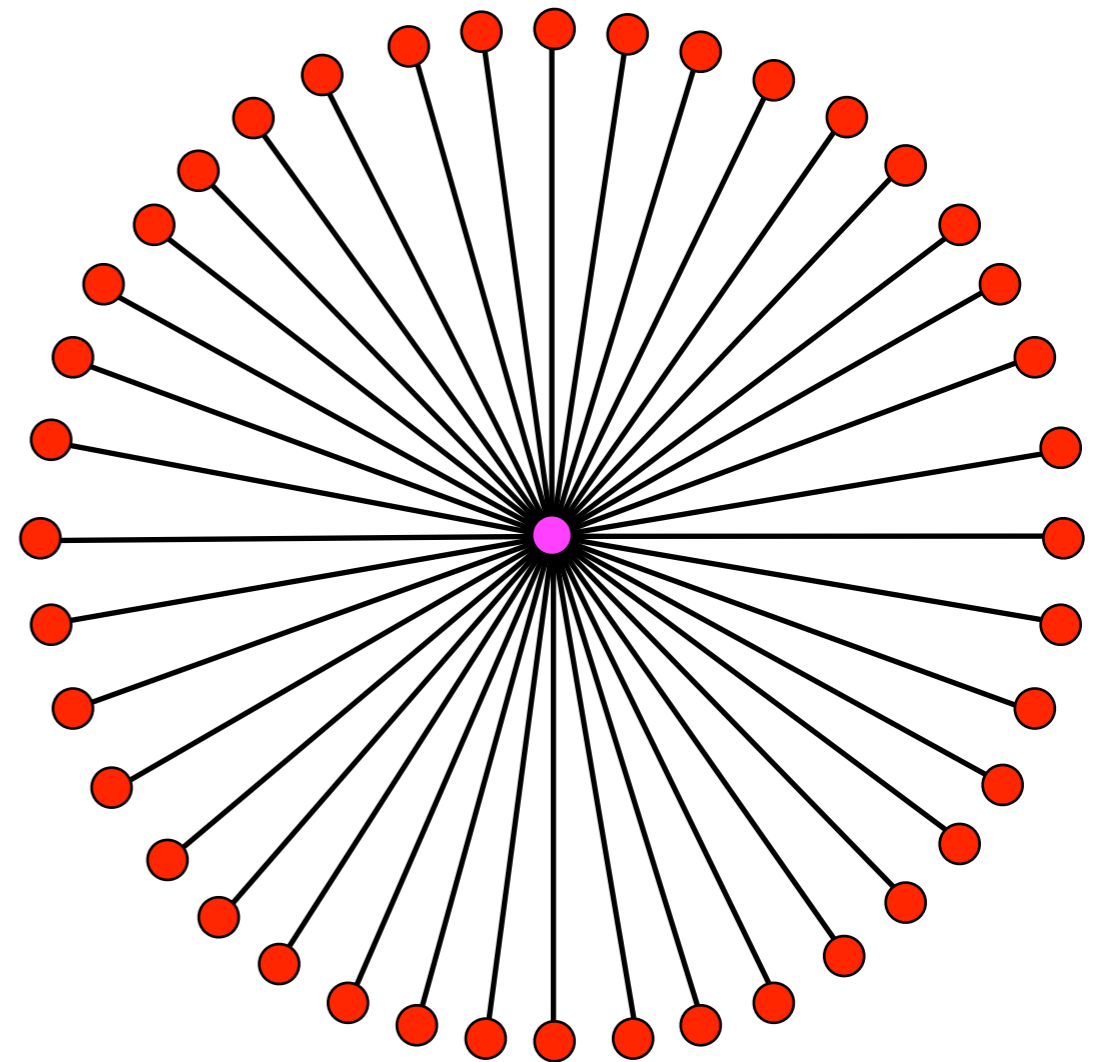
„hard“ problem to approximate an „easy“ one  
[Fekete, Meijer, Rohe, Tietze 2002]

matching

$$O(n^4)$$

$T$

he  
ints



**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching  
of the given points

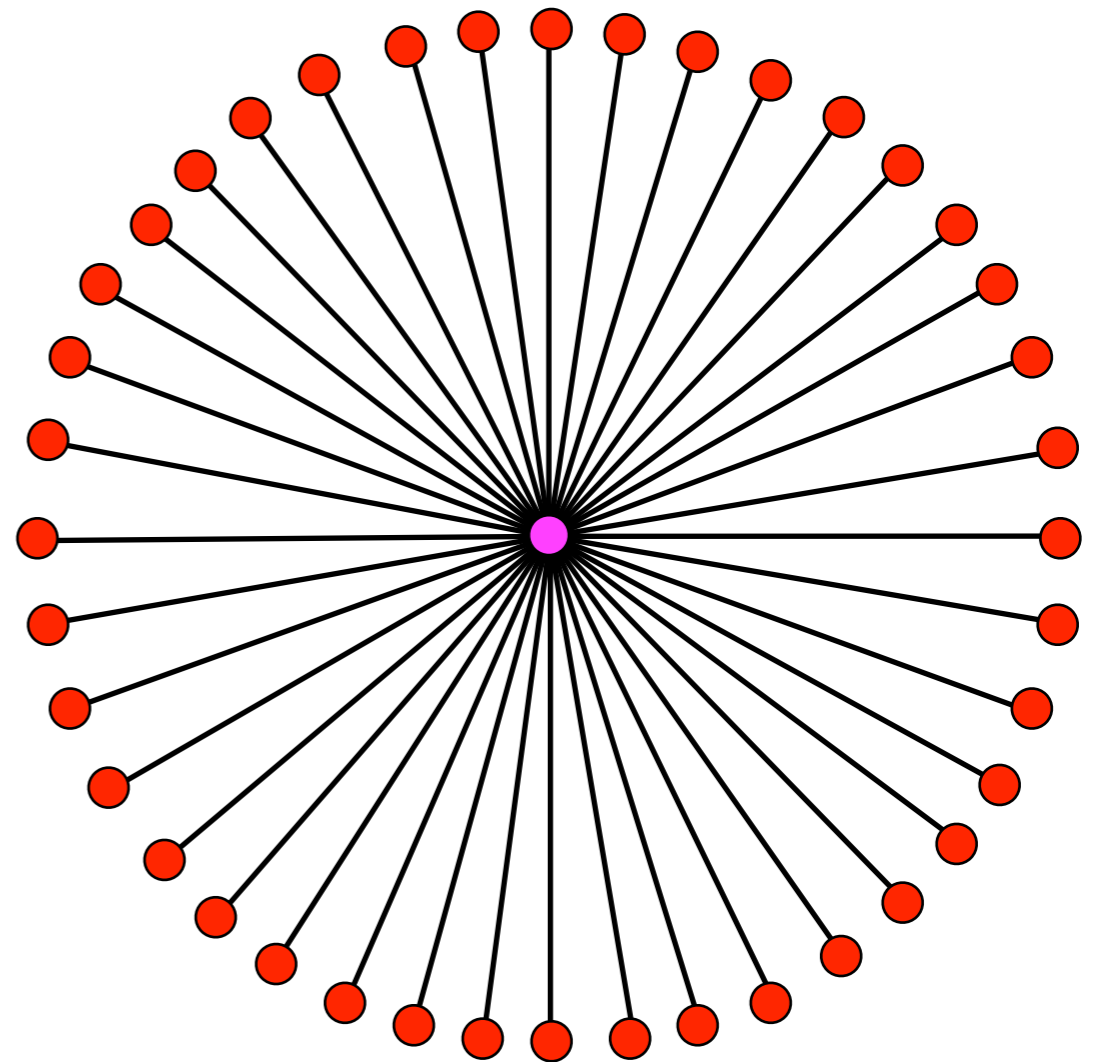
$$O(n^4)$$

**Problem 7.1:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the  
total distance to the given points

$$T$$



**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching  
of the given points

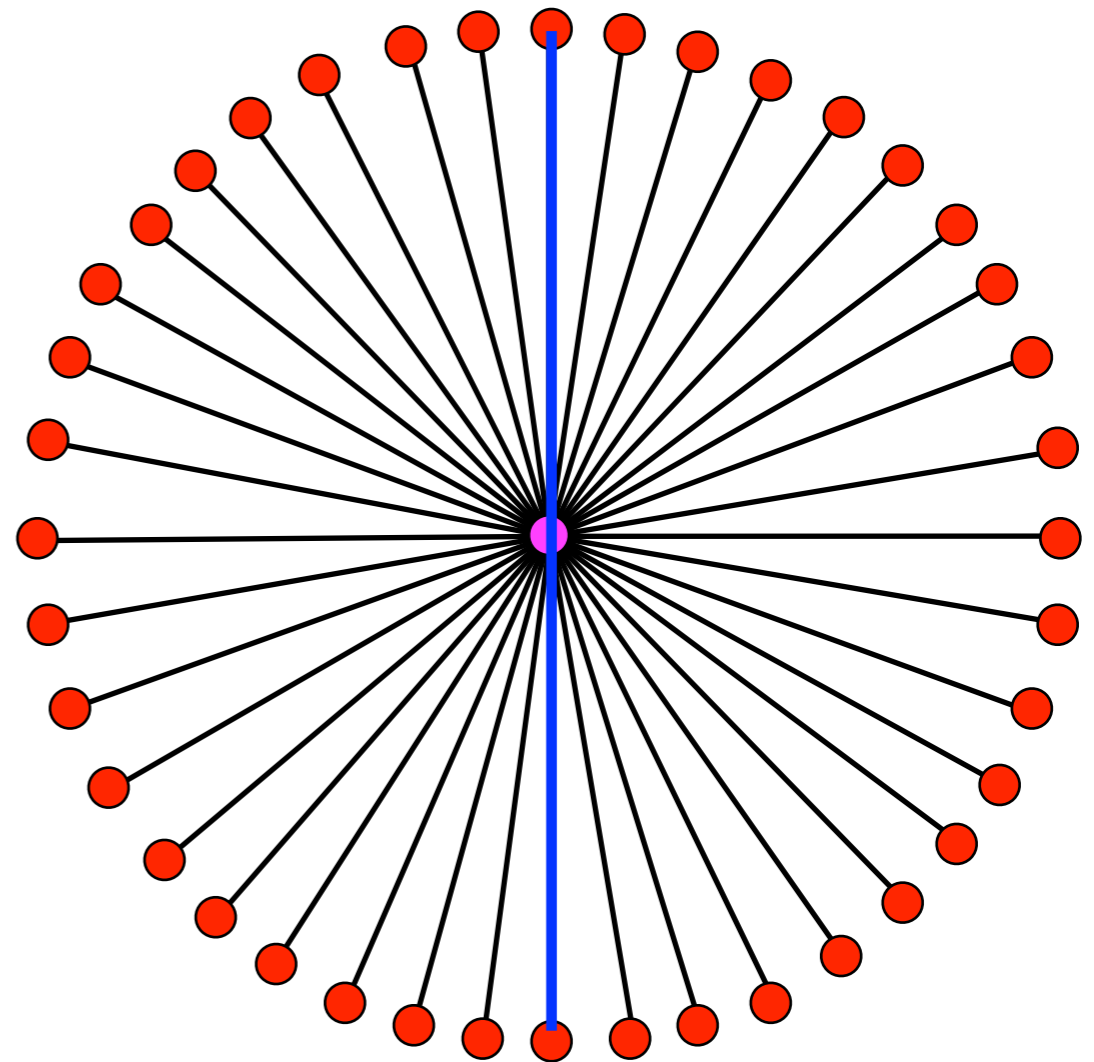
$$O(n^4)$$

**Problem 7.1:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the  
total distance to the given points

$$T$$



**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A maximum-weight perfect matching  
of the given points

$$O(n^4)$$

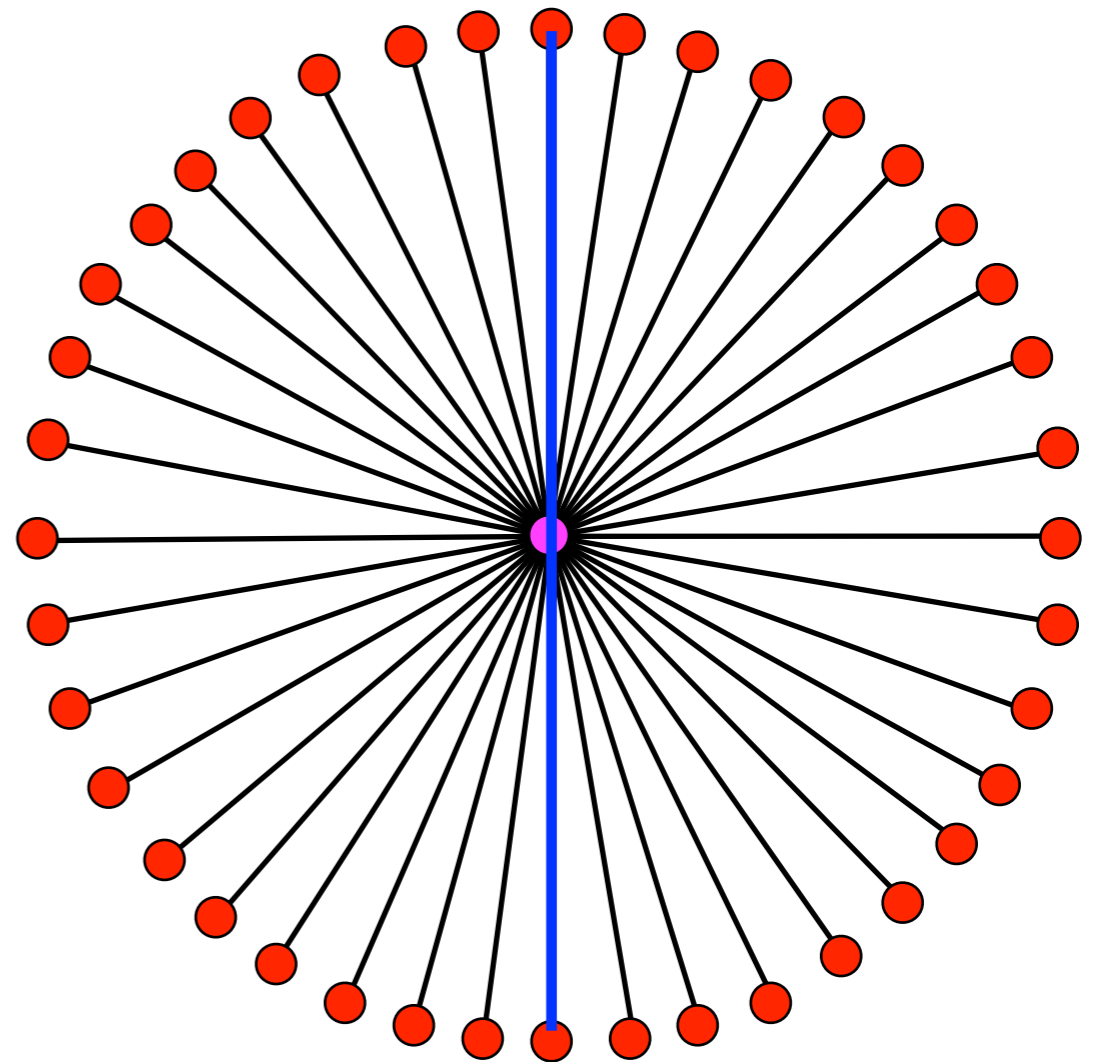
**Problem 7.1:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

**Wanted:** A location  $c$  that minimizes the  
total distance to the given points

$$T$$

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

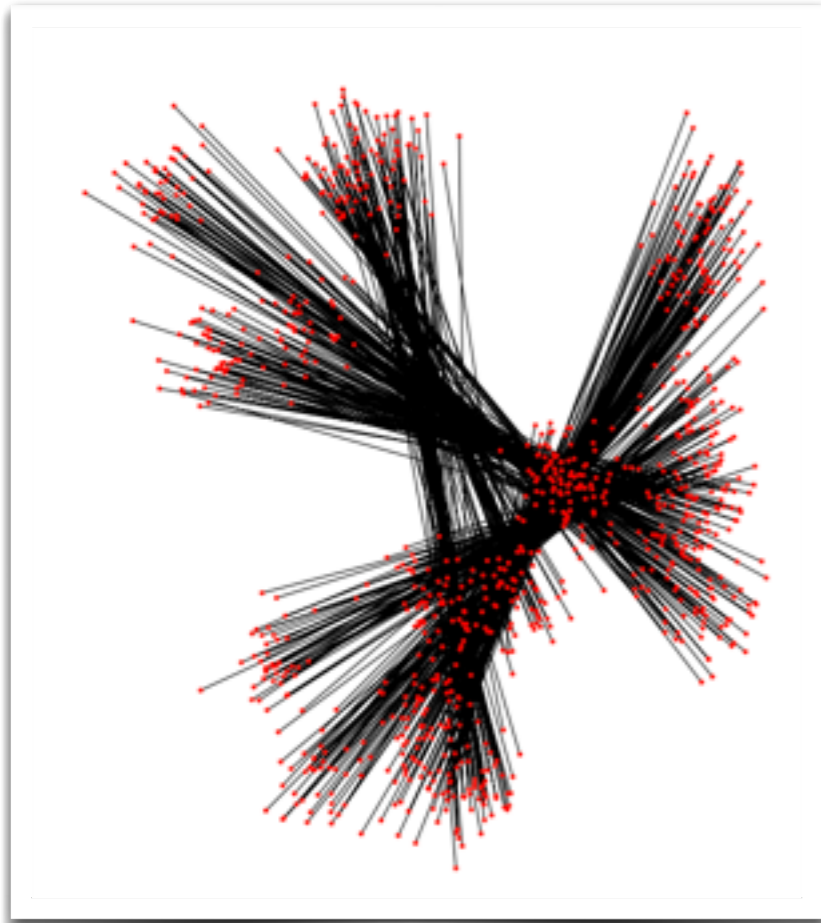


**Problem 7.3:**

**Given:** A set  $P$  of points in  $\mathbb{R}^2$

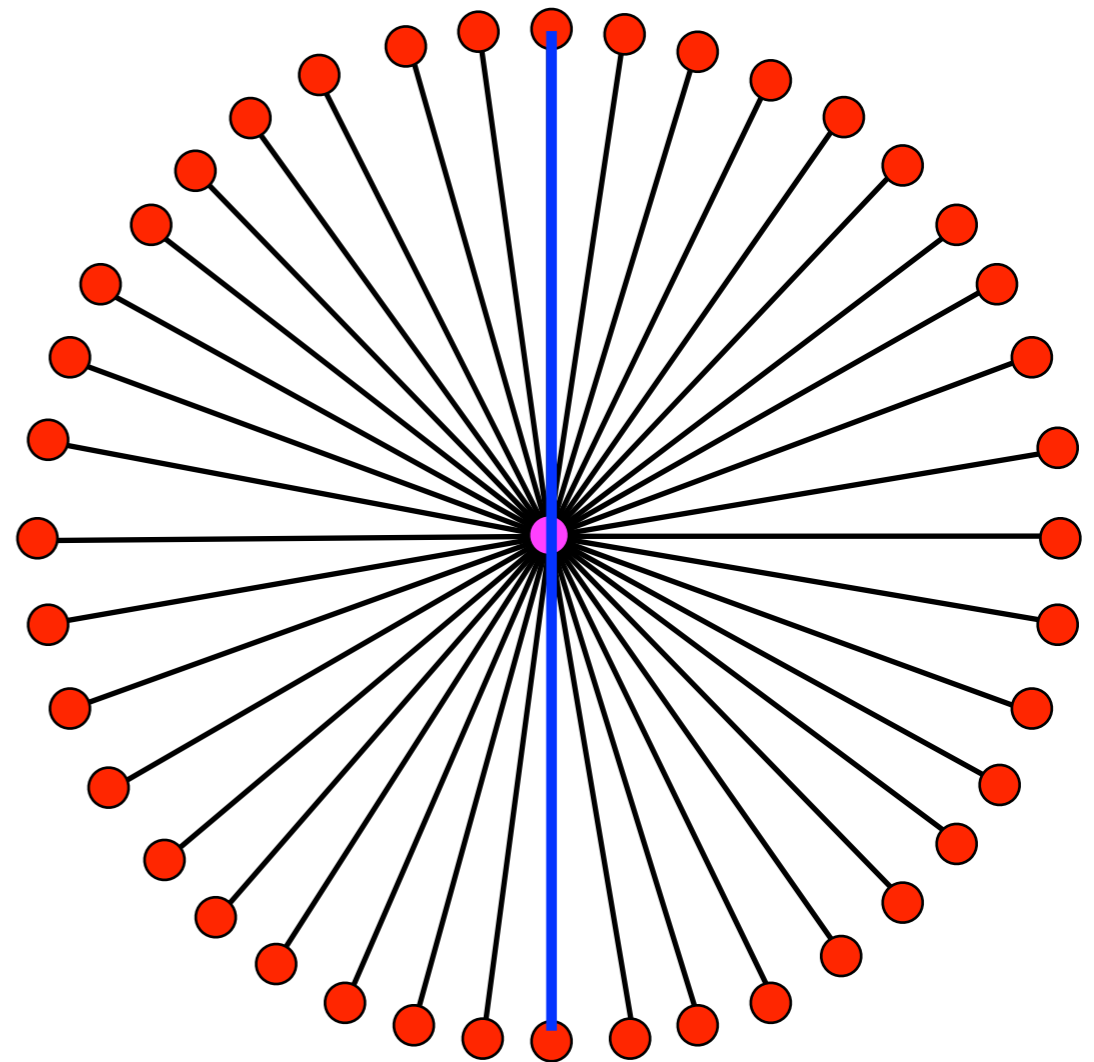
**Wanted:** A maximum-weight perfect matching  
of the given points

$$O(n^4)$$



ts in  $\mathbb{R}^2$   
t minimizes the  
the given points  
 $T + O(n \log n)$

$T$



1. Introduction
2. Longest Tours
3. Stars and Matchings
4. Nonsimple Polygons
5. Optimal Area
6. Turn Cost

# Shape Reconstruction



# Shape Reconstruction

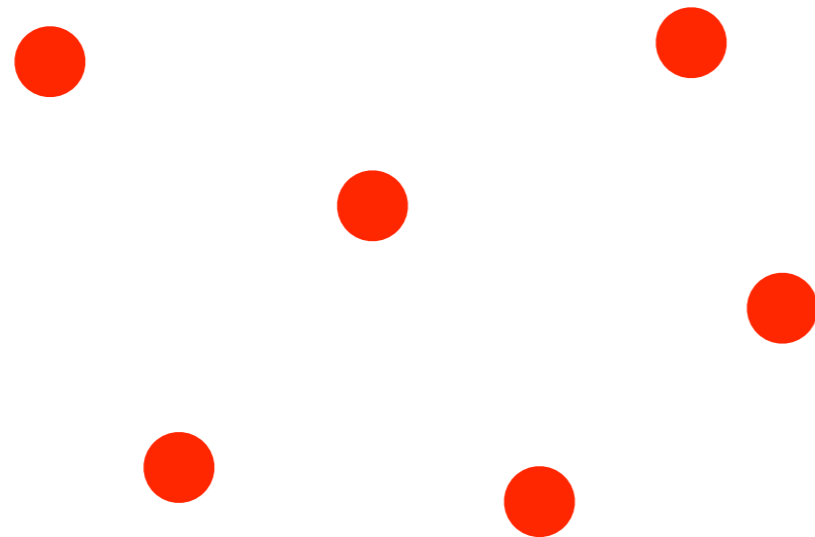
**Given:**

# Shape Reconstruction

**Given:**  $n$  points in the plane sampled from a shape

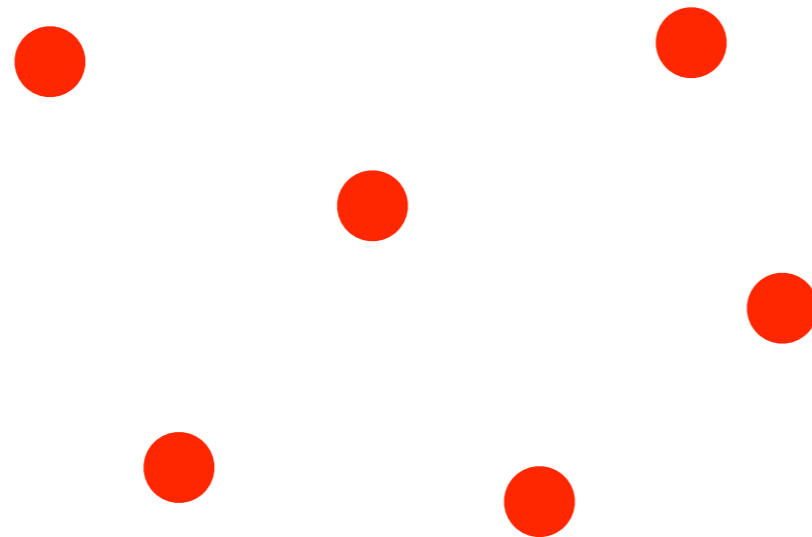
# Shape Reconstruction

**Given:**  $n$  points in the plane sampled from a shape



# Shape Reconstruction

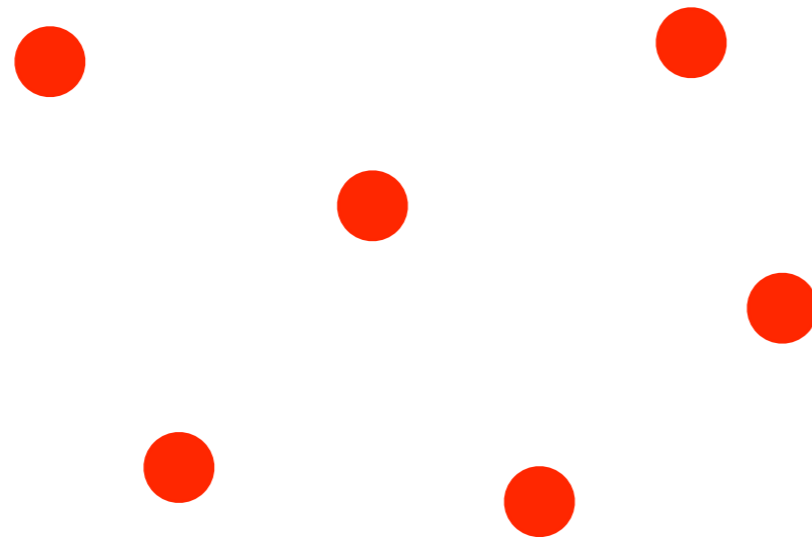
**Given:**  $n$  points in the plane sampled from a shape



**Wanted:**

# Shape Reconstruction

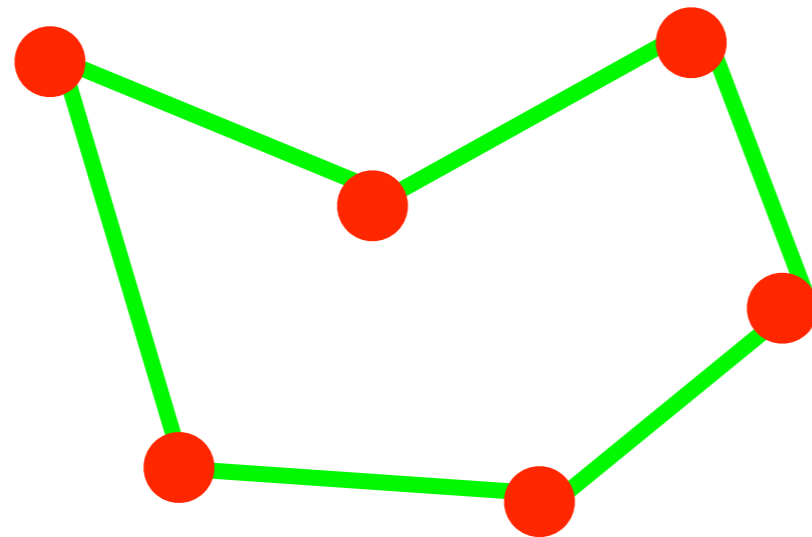
**Given:**  $n$  points in the plane sampled from a shape



**Wanted:** Reconstruct a curve through these points

# Shape Reconstruction

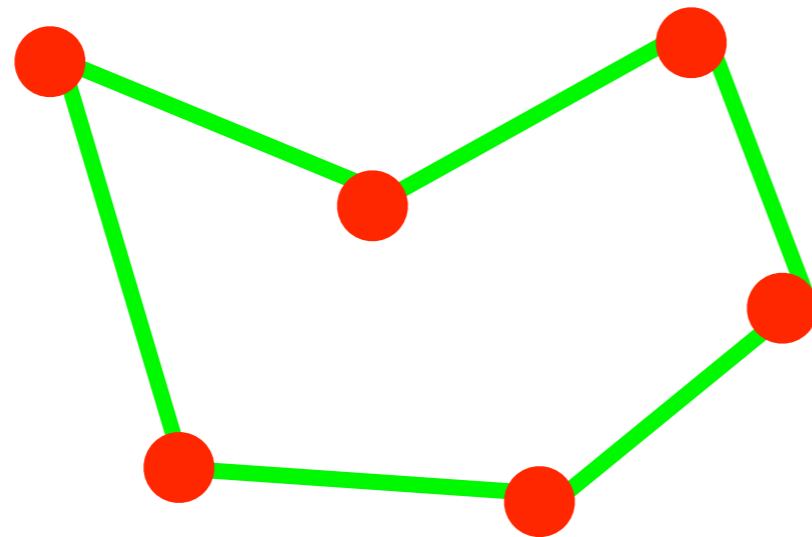
**Given:**  $n$  points in the plane sampled from a shape



**Wanted:** Reconstruct a curve through these points

# Shape Reconstruction

**Given:**  $n$  points in the plane sampled from a shape

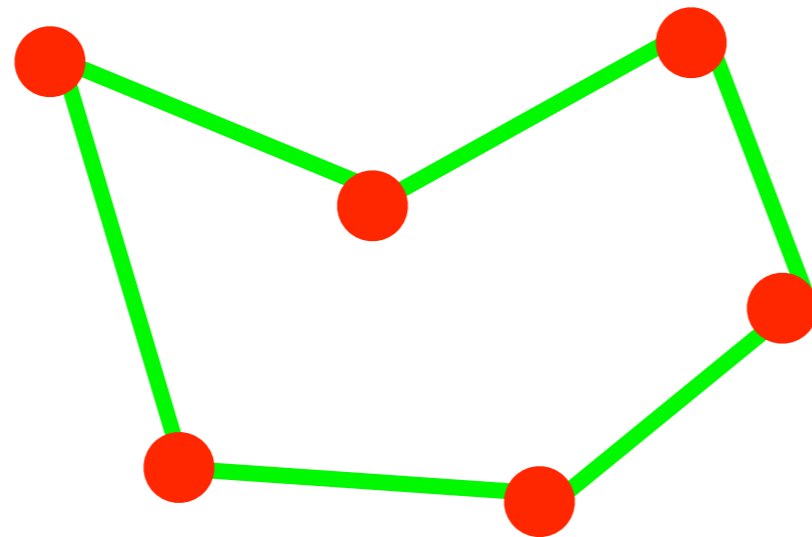


**Wanted:** Reconstruct a curve through these points

**Geometric:**

# Shape Reconstruction

**Given:**  $n$  points in the plane sampled from a shape



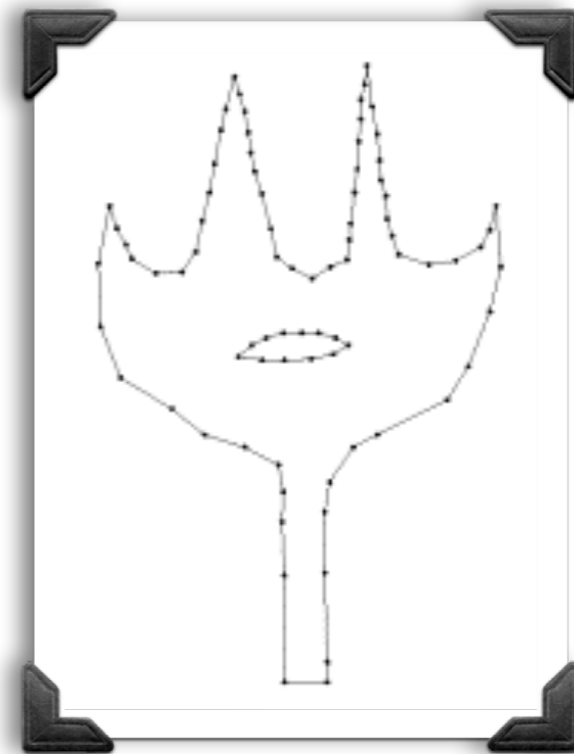
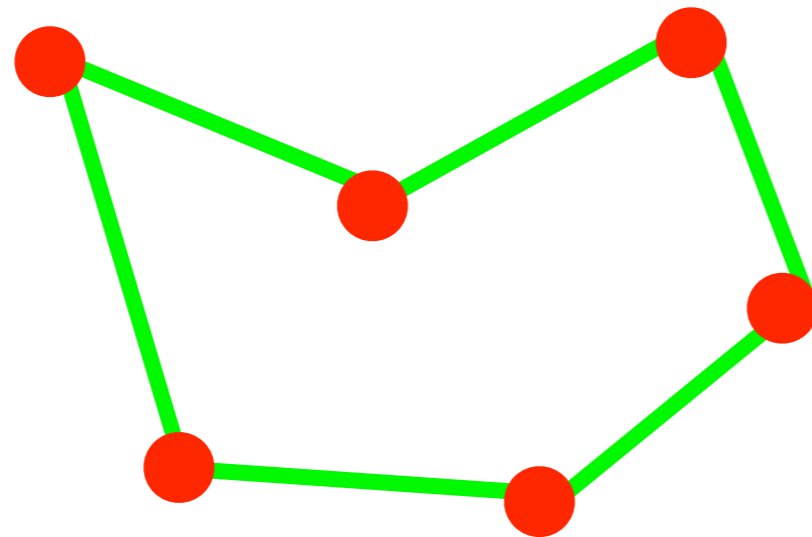
**Wanted:** Reconstruct a curve through these points

**Geometric:** Why a simple curve?!



# Shape Reconstruction

**Given:**  $n$  points in the plane sampled from a shape



**Wanted:** Reconstruct a curve through these points

**Geometric:** Why a simple curve?!

# Minimum Perimeter Polygon (MPP)

## COMPUTING NONSIMPLE POLYGONS OF MINIMUM PERIMETER\*

Sándor P. Fekete,<sup>†</sup> Andreas Haas,<sup>†</sup> Michael Hemmer,<sup>†</sup> Michael Hoffmann,<sup>‡</sup>  
 Irina Kostitsyna,<sup>§</sup> Dominik Krupke,<sup>†</sup> Florian Maurer,<sup>†</sup> Joseph S. B. Mitchell,<sup>¶</sup>  
 Arne Schmidt,<sup>†</sup> Christiane Schmidt,<sup>||</sup> Julian Troegel<sup>†</sup>

**ABSTRACT.** We consider the Minimum Perimeter Polygon Problem (MP3): for a given set  $V$  of points in the plane, find a polygon  $P$  with holes that has vertex set  $V$ , such that the total boundary length is smallest possible. The MP3 can be considered a natural geometric generalization of the Traveling Salesman Problem (TSP), which asks for a *simple* polygon with minimum perimeter. Just like the TSP, the MP3 occurs naturally in the context of curve reconstruction.

Even though the closely related problem of finding a minimum cycle cover is polynomially solvable by matching techniques, we prove how the topological structure of a polygon leads to NP-hardness of the MP3. On the positive side, we provide constant-factor approximation algorithms.

In addition to algorithms with theoretical worst-case guarantess, we provide *practical* methods for computing provably optimal solutions for relatively large instances, based on integer programming. An additional difficulty compared to the TSP is the fact that only a subset of subtour constraints is valid, depending not on combinatorics, but on geometry. We overcome this difficulty by establishing and exploiting geometric properties. This allows us to reliably solve a wide range of benchmark instances with up to 600 vertices within reasonable time on a standard machine. We also show that restricting the set of connections between points to edges of the Delaunay triangulation yields results that are on average within 0.5% of the optimum for large classes of benchmark instances.

### 1 Introduction

Two of the most fundamental structures in Computational Geometry are planar point sets and polygons. In this paper we study a natural algorithmic connection between them. For

\*A preliminary extended abstract [13] appeared in the Proceedings of the 15th Symposium on Experimental and Efficient Algorithms (SEA 2016).

<sup>†</sup>Department of Computer Science, TU Braunschweig, Mühlenpfordtstr. 23, 38106 Braunschweig, Germany. {s.fekete,a.haas,m.hemmer,d.krupke,arne.schmidt,j.troegel}@tu-bs.de

<sup>‡</sup>Department of Computer Science, ETH Zürich, Universitätstrasse 6, 8092 Zürich, Switzerland. hoffmann@inf.ethz.ch

<sup>§</sup>Department of Mathematics and Computer Science, TU Eindhoven, 5600 MB Eindhoven, The Netherlands. i.kostitsyna@tue.nl

<sup>¶</sup>Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794-3600, USA. joseph.mitchell@stonybrook.edu

<sup>||</sup>Department of Science and Technology, Linköping University, SE 60174 Norrköping, Sweden. christiane.schmidt@liu.se

## COMPUTING NONSIMPLE POLYGONS OF MINIMUM PERIMETER\*

Sándor P. Fekete,<sup>†</sup> Andreas Haas,<sup>†</sup> Michael Hemmer,<sup>†</sup> Michael Hoffmann,<sup>‡</sup>  
Irina Kostitsyna,<sup>§</sup> Dominik Krupke,<sup>†</sup> Florian Maurer,<sup>†</sup> Joseph S. B. Mitchell,<sup>¶</sup>  
Arne Schmidt,<sup>†</sup> Christiane Schmidt,<sup>||</sup> Julian Troegel<sup>†</sup>

*Sándor P. Fekete,<sup>†</sup> Andreas Haas,<sup>†</sup> Michael Hemmer,<sup>†</sup> Michael Hoffmann,<sup>‡</sup>  
Irina Kostitsyna,<sup>§</sup> Dominik Krupke,<sup>†</sup> Florian Maurer,<sup>†</sup> Joseph S. B. Mitchell,<sup>¶</sup>  
Arne Schmidt,<sup>†</sup> Christiane Schmidt,<sup>||</sup> Julian Troegel<sup>†</sup>*

imation algorithms.

In addition to algorithms with theoretical worst-case guarantess, we provide *practical* methods for computing provably optimal solutions for relatively large instances, based on integer programming. An additional difficulty compared to the TSP is the fact that only a subset of subtour constraints is valid, depending not on combinatorics, but on geometry. We overcome this difficulty by establishing and exploiting geometric properties. This allows us to reliably solve a wide range of benchmark instances with up to 600 vertices within reasonable time on a standard machine. We also show that restricting the set of connections between points to edges of the Delaunay triangulation yields results that are on average within 0.5% of the optimum for large classes of benchmark instances.

## 1 Introduction

Two of the most fundamental structures in Computational Geometry are planar point sets and polygons. In this paper we study a natural algorithmic connection between them. For

\*A preliminary extended abstract [13] appeared in the Proceedings of the 15th Symposium on Experimental and Efficient Algorithms (SEA 2016).

<sup>†</sup>Department of Computer Science, TU Braunschweig, Mühlenpfordtstr. 23, 38106 Braunschweig, Germany. {s.fekete,a.haas,m.hemmer,d.krupke,arne.schmidt,j.troegel}@tu-bs.de

<sup>‡</sup>Department of Computer Science, ETH Zürich, Universitätstrasse 6, 8092 Zürich, Switzerland. hoffmann@inf.ethz.ch

<sup>§</sup>Department of Mathematics and Computer Science, TU Eindhoven, 5600 MB Eindhoven, The Netherlands. i.kostitsyna@tue.nl

<sup>¶</sup>Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794-3600, USA. joseph.mitchell@stonybrook.edu

<sup>||</sup>Department of Science and Technology, Linköping University, SE 60174 Norrköping, Sweden. christiane.schmidt@liu.se

# Minimum Perimeter Polygon (MPP)

# Minimum Perimeter Polygon (MPP)

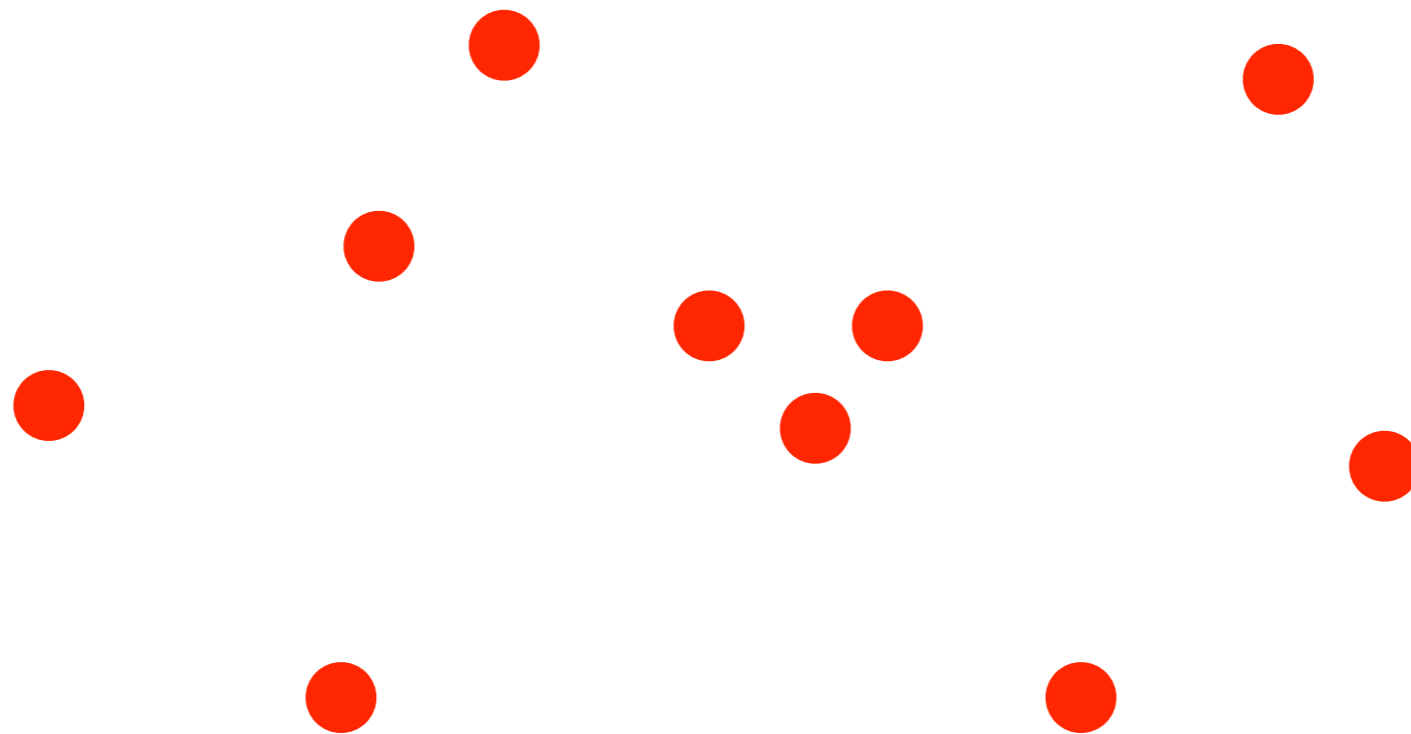
**Given:**

# Minimum Perimeter Polygon (MPP)

**Given:**  $n$  points in the plane

# Minimum Perimeter Polygon (MPP)

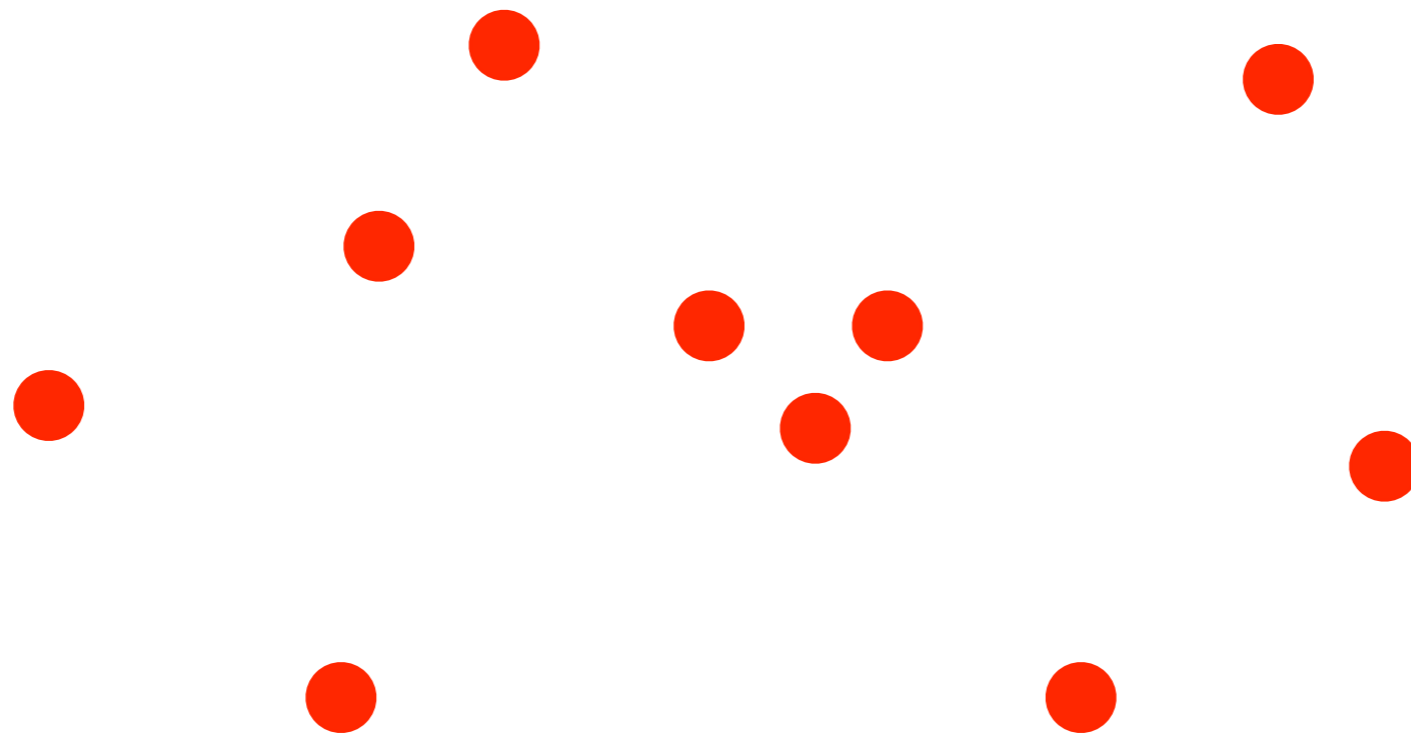
**Given:**  $n$  points in the plane





# Minimum Perimeter Polygon (MPP)

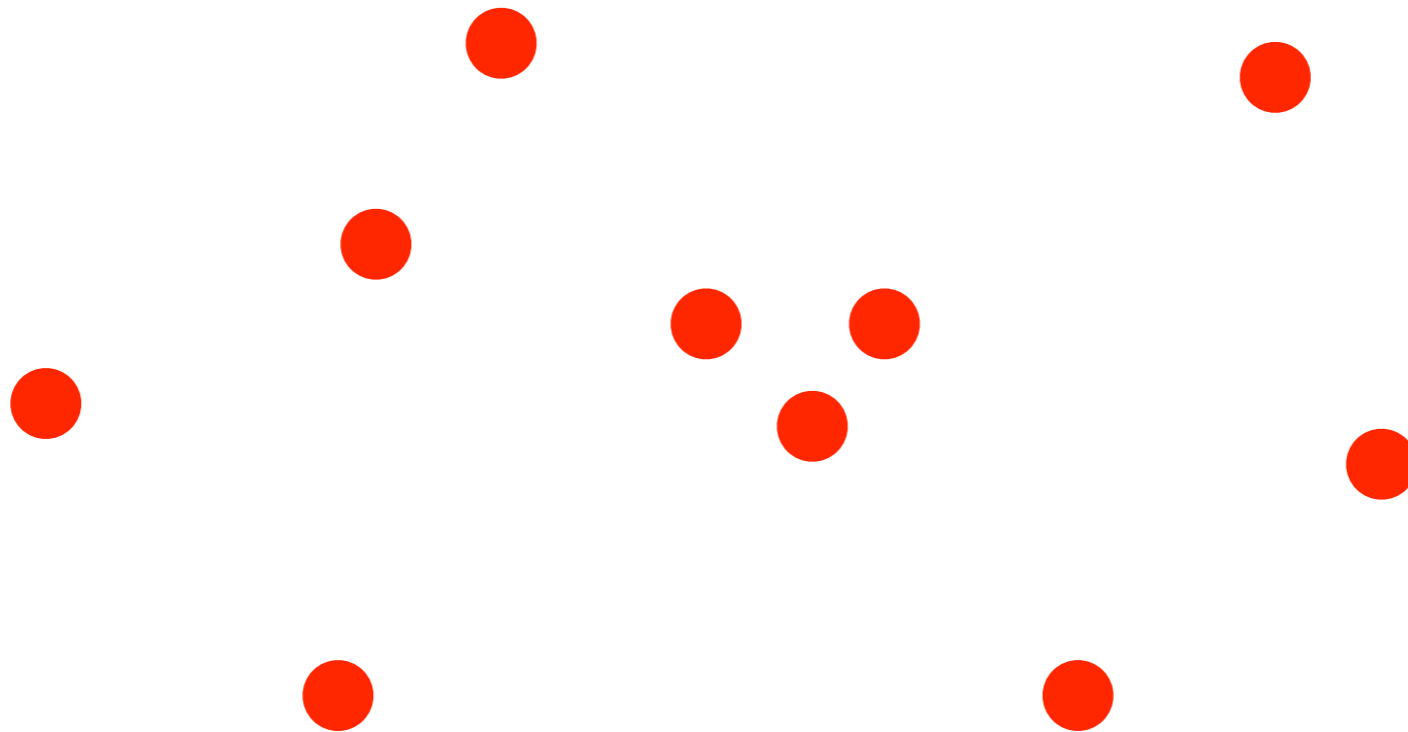
**Given:**  $n$  points in the plane



**Wanted:**

# Minimum Perimeter Polygon (MPP)

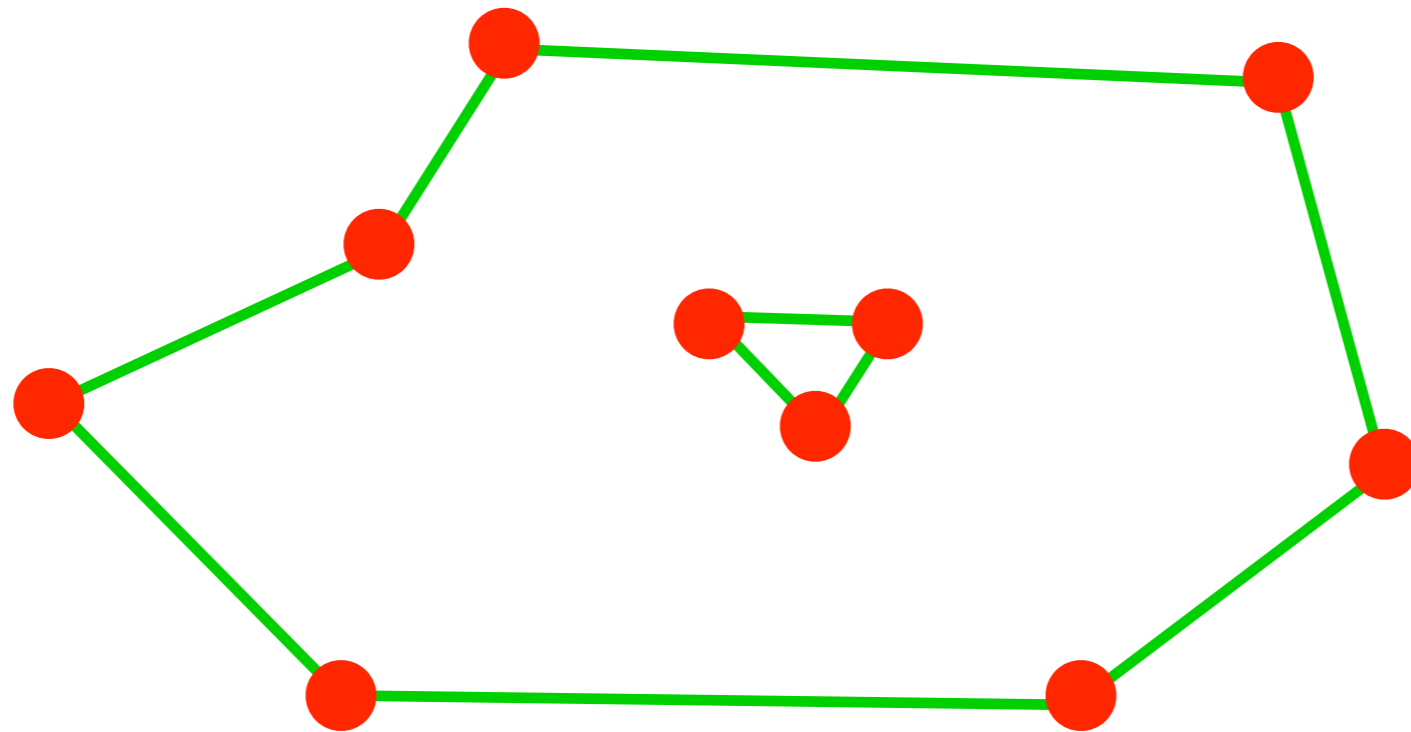
**Given:**  $n$  points in the plane



**Wanted:** A valid polygon of minimum perimeter length

# Minimum Perimeter Polygon (MPP)

**Given:**  $n$  points in the plane



**Wanted:** A valid polygon of minimum perimeter length

# Minimum Perimeter Polygon (MPP)



Figure 1: A Minimum Perimeter Polygon for an instance with 960 vertices.

# An Open Problem Resolved

# An Open Problem Resolved

**Theorem 1.** *The MPP problem is NP-hard.*

# Constant-Factor Approximation

# Constant-Factor Approximation

**Theorem 2.** *There exists a polynomial time 3-approximation for the MPP.*



# 3-Approximation

# 3-Approximation

Idea:

# 3-Approximation

- Idea:**
- Compute outer hull

# 3-Approximation

**Idea:**

- Compute outer hull

$\leq \text{OPT}$

# 3-Approximation

**Idea:**

- Compute outer hull
- Compute 2-factor of interior points

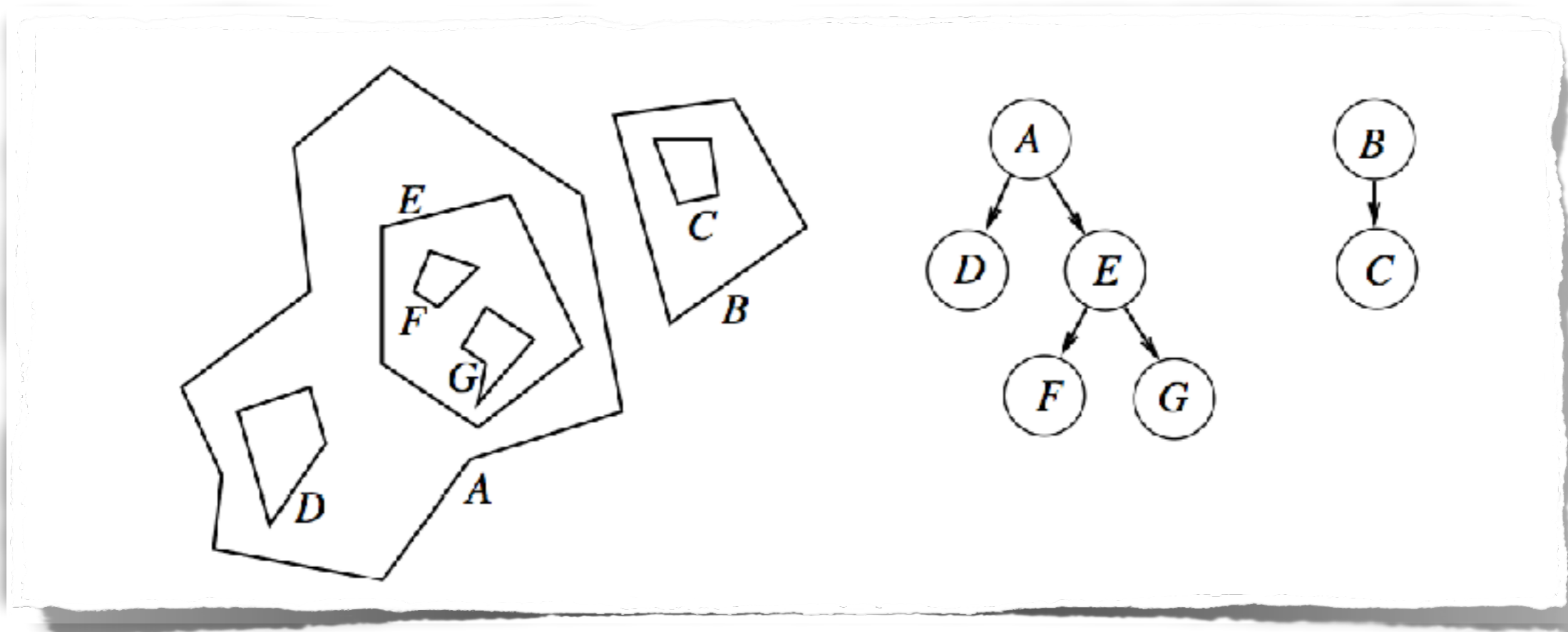
$\leq \text{OPT}$

# 3-Approximation

Idea:

- Compute outer hull
- Compute 2-factor of interior points

$\leq \text{OPT}$



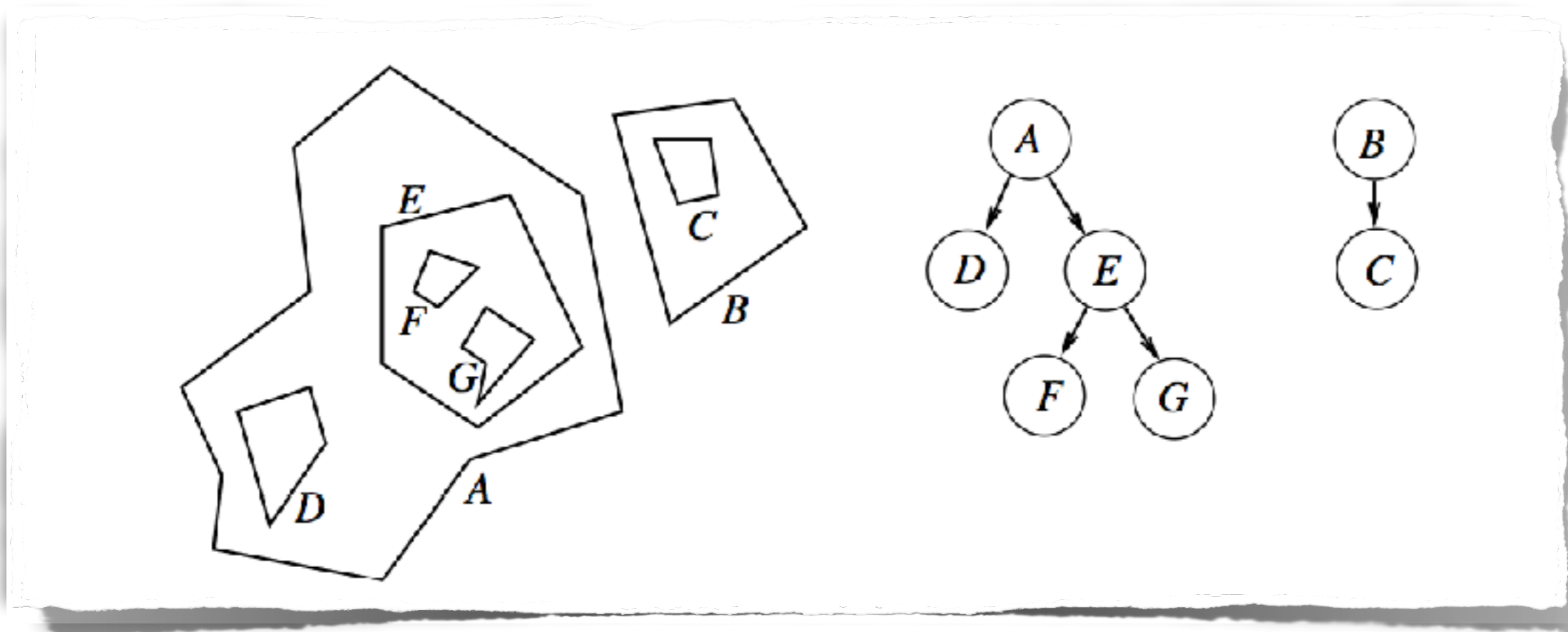
# 3-Approximation

Idea:

- Compute outer hull
- Compute 2-factor of interior points

$\leq \text{OPT}$

$\leq \text{OPT}$



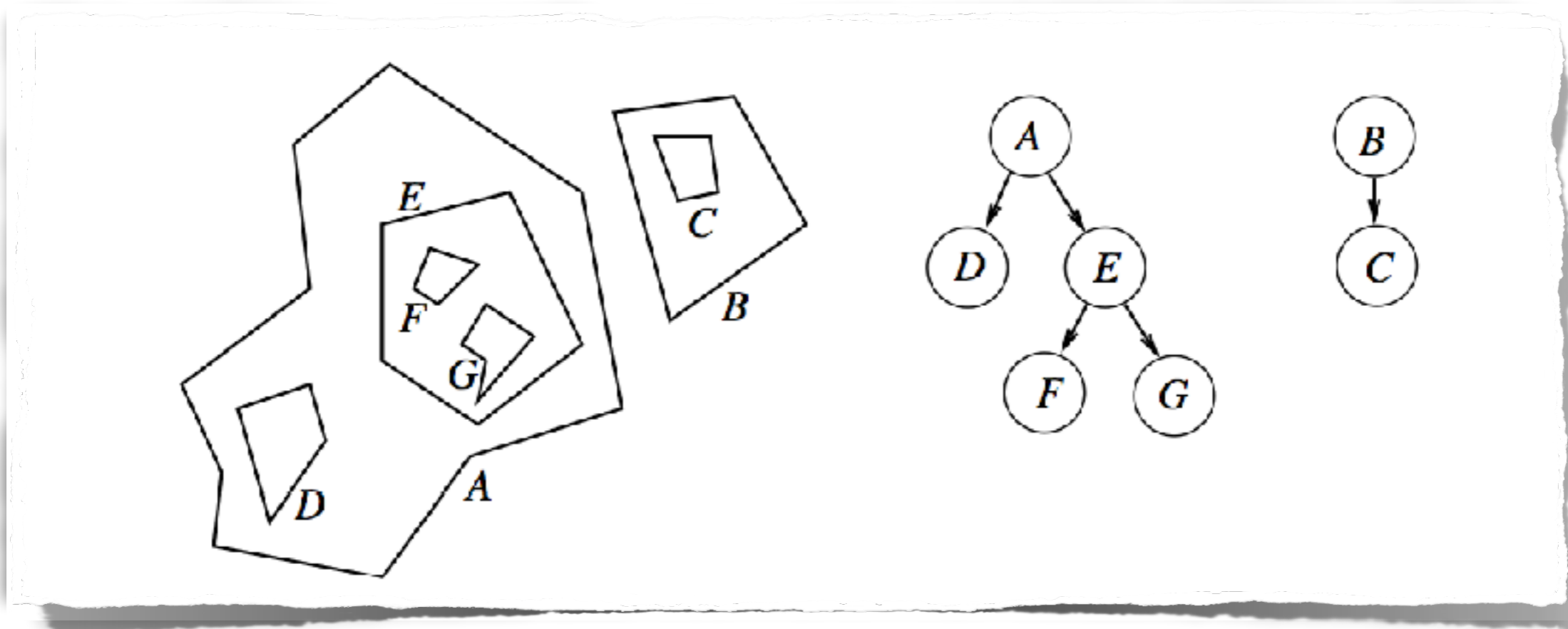
# 3-Approximation

Idea:

- Compute outer hull
- Compute 2-factor of interior points
- Merge cycles

$\leq \text{OPT}$

$\leq \text{OPT}$





# 3-Approximation

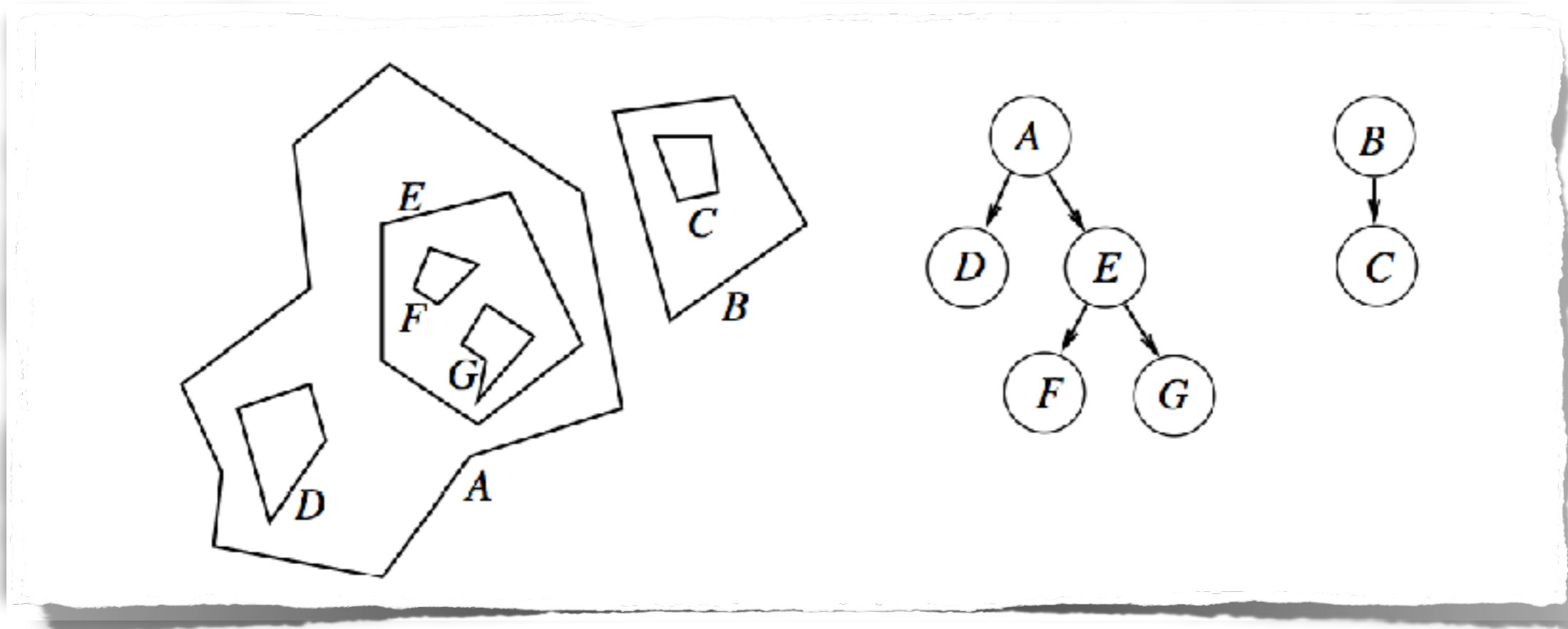
Idea:

- Compute outer hull
- Compute 2-factor of interior points
- Merge cycles

$\leq \text{OPT}$

$\leq \text{OPT}$

$\leq \text{OPT}$



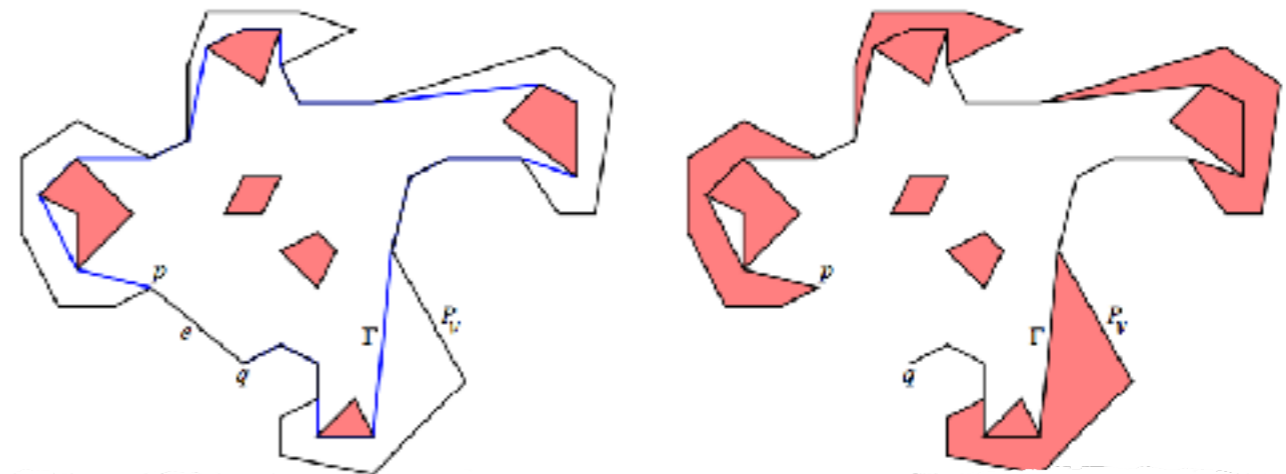
# Technical Details: Merge Cycles (1)

# Technical Details: Merge Cycles (1)

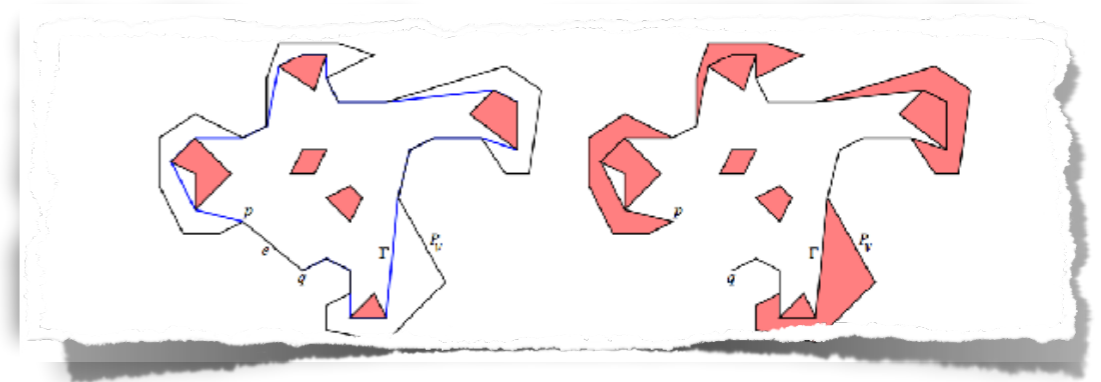
We now describe phase I. Let  $T$  be a nontrivial tree of  $F$ . Associated with  $T$  are a set of cycles, one per node. A node  $v$  of  $T$  that has no outgoing edge of  $T$  (i.e.,  $v$  has no children) is a sink node; it corresponds to a cycle that has no cycle contained within it. Let  $v$  be a node of  $T$  that has at least one child, but no grandchildren. (Such a node must exist in a nontrivial tree  $T$ .) Then,  $v$  corresponds to a cycle (simple polygon)  $P_v$ , within which there is one or more disjoint simple polygonal cycles,  $P_{u_1}, P_{u_2}, \dots, P_{u_k}$ , one for each of the  $k \geq 1$  children of  $v$ . We describe an operation that replaces  $P_v$  with a new weakly simple polygon,  $Q_v$ , whose interior is disjoint from those of  $P_{u_1}, P_{u_2}, \dots, P_{u_k}$ . Let  $e = pq$  ( $p, q \in V$ ) be any edge of  $P_v$ ; assume that  $pq$  is a counterclockwise edge, so that the interior of  $P_v$  lies to the left of the oriented segment  $pq$ . Let  $\Gamma$  be a shortest path within  $P_v$ , from  $p$  to  $q$ , that has all of the polygons  $P_{u_1}, P_{u_2}, \dots, P_{u_k}$  to its right; thus,  $\Gamma$  is a "taut string" path within  $P_v$ , homotopically equivalent to  $\partial P_v$ , from  $p$  to  $q$ . (Such a geodesic path is related to the "relative convex hull" of the polygons  $P_{u_1}, P_{u_2}, \dots, P_{u_k}$  within  $P_v$ , which is the shortest cycle within  $P_v$  that encloses all of the polygons; the difference is that  $\Gamma$  is "anchored" at the endpoints  $p$  and  $q$ .) Note that  $\Gamma$  is a polygonal path whose vertices are either (convex) vertices of the polygons  $P_{u_i}$  or (reflex) vertices of  $P_v$ . Consider the closed polygonal walk that starts at  $p$ , follows the path  $\Gamma$  to  $q$ , then continues counterclockwise around the boundary,  $\partial P_v$ , of  $P_v$  until it returns to  $p$ . This closed polygonal walk is the counterclockwise traversal of a weakly simple polygon,  $Q_v$ , whose interior is disjoint from the interiors of the polygons  $P_{u_1}, P_{u_2}, \dots, P_{u_k}$ . Refer to Figure 8. The length of this closed walk (the counterclockwise traversal of the boundary of  $Q_v$ ) is at most twice the perimeter of  $P_v$ , since the path  $\Gamma$  has length at most that of the counterclockwise boundary  $\partial P_v$ , from  $q$  to  $p$  (since  $\Gamma$  is a homotopically equivalent shortening of this boundary). We consider the boundary of  $P_v$  to be replaced with the cycle around the boundary of  $Q_v$ , and this process has reduced the degree of nesting in  $T$ : node  $v$  that used to have  $k$  children (leaves of  $T$ ) is now replaced by a node  $v'$  corresponding to  $Q_v$ , and  $v'$  and the  $k$  children of  $v$  are now all siblings in the modified tree,  $T'$ . If  $v$  had a parent,  $w$ , in  $T$ , then  $v'$  and the  $k$  children of  $v$  are now children of  $w$ ; if  $v$  had no parent in  $T$  (i.e., it was the root of  $T$ ), then  $T$  has been transformed into a set of  $k+1$  cycles, none of which are nested within another cycle of  $\gamma(U)$ . (Each is within the convex hull  $CH(V)$ , but there is no other surrounding cycle of  $\gamma(U)$ .) We continue this process of transforming a surrounding parent cycle (node  $v$ ) into a sibling cycle (node  $v'$ ), until each tree  $T$  of  $F$  becomes a set of isolated nodes, and finally  $F$  has no edges (there is no nesting).

# Technical Details: Merge Cycles (1)

We now describe phase I. Let  $T$  be a nontrivial tree of  $F$ . Associated with  $T$  are a set of cycles, one per node. A node  $u$  of  $T$  that has no outgoing edge of  $T$  (i.e.,  $u$  has no children) is a sink node; it corresponds to a cycle that has no cycle contained within it. Let  $v$  be a node of  $T$  that has at least one child, but no grandchildren. (Such a node must exist in a nontrivial tree  $T$ .) Then,  $v$  corresponds to a cycle (simple polygon)  $P_v$ , within which there is one or more disjoint simple polygonal cycles,  $P_{u_1}, P_{u_2}, \dots, P_{u_k}$ , one for each of the  $k \geq 1$  children of  $v$ . We describe an operation that replaces  $P_v$  with a new weakly simple polygon,  $Q_v$ , whose interior is disjoint from those of  $P_{u_1}, P_{u_2}, \dots, P_{u_k}$ . Let  $e = pq$  ( $p, q \in V$ ) be any edge of  $P_v$ ; assume that  $pq$  is a counterclockwise edge, so that the interior of  $P_v$  lies to the left of the oriented segment  $pq$ . Let  $\Gamma$  be a path within  $P_v$ , from  $p$  to  $q$ , that has all of the polygons  $P_{u_1}, P_{u_2}, \dots, P_{u_k}$  within its interior.  $\Gamma$  is a "taut string" path within  $P_v$ , homotopically equivalent to the boundary of  $P_v$ . (Such a geodesic path is related to the "relative convex hull" of the polygons  $P_{u_1}, P_{u_2}, \dots, P_{u_k}$  within  $P_v$ , which is the shortest cycle within  $P_v$  that encloses all of the polygons; the difference is that  $\Gamma$  is "anchored" at  $p$  and  $q$ .) Note that  $\Gamma$  is a polygonal path whose vertices are either vertices of the polygons  $P_{u_i}$  or (reflex) vertices of  $P_v$ . Consider the closed path that starts at  $p$ , follows the path  $\Gamma$  to  $q$ , then continues counterclockwise along the boundary,  $\partial P_v$ , of  $P_v$ , until it returns to  $p$ . This closed path is a counterclockwise traversal of a weakly simple polygon,  $Q_v$ , whose interior is disjoint from the interiors of the polygons  $P_{u_1}, P_{u_2}, \dots, P_{u_k}$ . The length of this closed walk (the counterclockwise traversal of the boundary of  $Q_v$ ) is at most twice the perimeter of  $P_v$ , since the path  $\Gamma$  has length at most the perimeter of  $P_v$  (since  $\Gamma$  is a counterclockwise traversal of the boundary of  $P_v$ , from  $q$  to  $p$  (since  $pq$  is a counterclockwise edge of  $P_v$ )). We consider the boundary of  $P_v$  to be replaced with the cycle around the boundary of  $Q_v$ , and this process has reduced the degree of nesting in  $T$ : node  $v$  that used to have  $k$  children (leaves of  $T$ ) is now replaced by a node  $v'$  corresponding to  $Q_v$ , and  $v'$  and the  $k$  children of  $v$  are now all siblings in the modified tree,  $T'$ . If  $v$  had a parent,  $w$ , in  $T$ , then  $v'$  and the  $k$  children of  $v$  are now children of  $w$ ; if  $v$  had no parent in  $T$  (i.e., it was the root of  $T$ ), then  $T$  has been transformed into a set of  $k+1$  cycles, none of which are nested within another cycle of  $\gamma(U)$ . (Each is within the convex hull  $CH(V)$ , but there is no other surrounding cycle of  $\gamma(U)$ .) We continue this process of transforming a surrounding parent cycle (node  $v$ ) into a sibling cycle (node  $v'$ ), until each tree  $T$  of  $F$  becomes a set of isolated nodes, and finally  $F$  has no edges (there is no nesting).



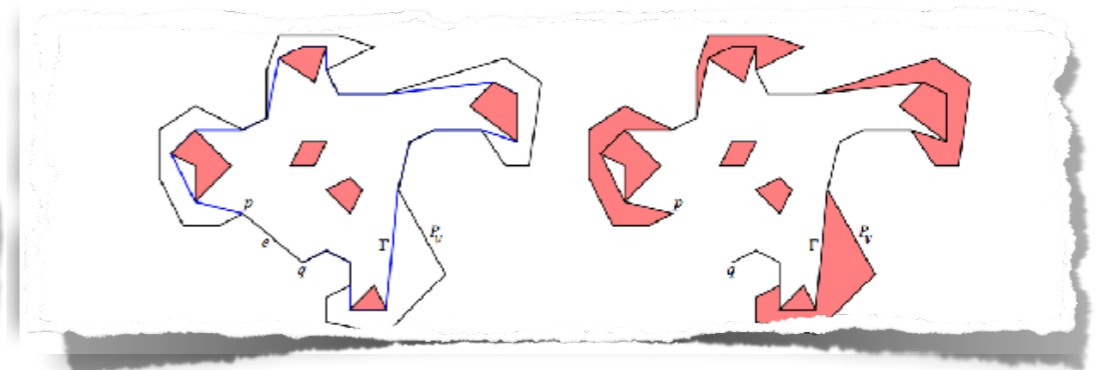
# Technical Details: Merge Cycles (2)



# Technical Details: Merge Cycles (2)

Consider a weakly simple polygon  $Q$ , and let  $p$  be a vertex of  $Q$  which is repeated in the cycle specifying the boundary  $\partial Q$ . This implies that there are four edges of the (counterclockwise) cycle,  $p_0p$ ,  $pp_1$ ,  $p_2p$ , and  $pp_3$ , incident on  $p$ , all of which lie within a halfplane through  $p$  (by local optimality). There are then two subcases: (i)  $p_0, p, p_1$  is a left turn (Figure 9, left); and (ii)  $p_0, p, p_1$  is a right turn (Figure 9, right). In subcase (i),  $p_0p, pp_1$  define a left turn at  $p$  (making  $p$  locally convex for  $Q$ ), and  $p_2p, pp_3$  define a right turn at  $p$  (making  $p$  locally reflex for  $Q$ ). In this case, we replace the pair of edges  $p_0p, pp_1$  with a shorter polygonal chain, namely the “taut” version of this path, from  $p_0$  to  $p_1$ , along a shortest path,  $\beta_{0,1}$ , among the polygons  $Q_i$ , including  $Q$ , treating them as obstacles. The taut path  $\beta_{0,1}$  consists of left turns only, at (locally convex) vertices of polygons  $Q_i$  ( $Q_i \neq Q$ ) or (locally reflex) vertices of  $Q$ , where new pinch points of  $Q$  are created. Refer to Figure 9, left. Case (ii) is treated similarly; see Figure 9, right. Thus, resolving one repeated vertex,  $p$ , of  $Q$  can result in the creation of other repeated vertices of  $Q$ , or repeated vertices where two cycles come together (discussed below). The process is finite, though, since the total length of all cycles strictly decreases with each operation; in fact, there can be only a polynomial number of such adjustments, since each triple  $(p_0, p, p_1)$ , is resolved at most once.

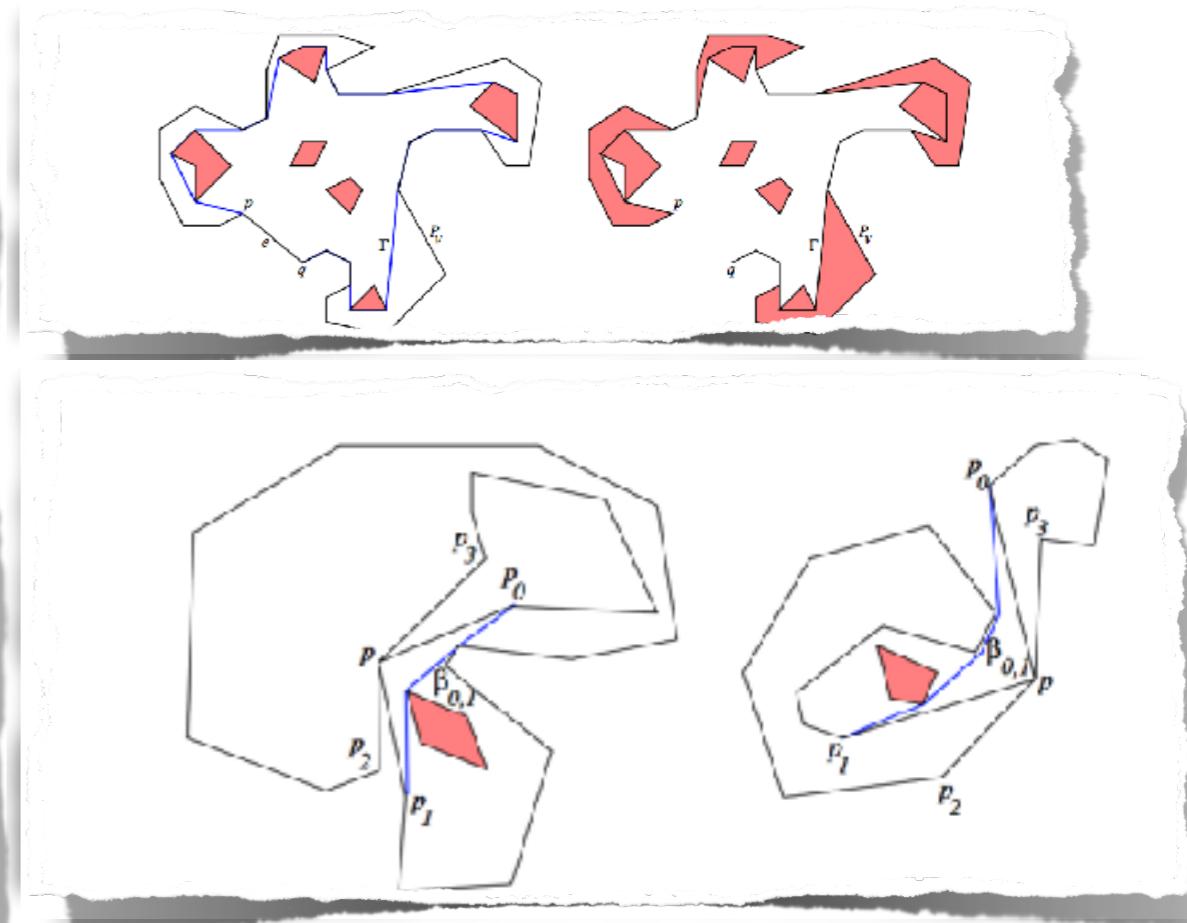
Now consider a vertex  $p$  that appears once as a reflex vertex in  $Q_1$  (with incident ccw edges  $p_0p$  and  $pp_1$ ) and once as a convex vertex in  $Q_2$  (with incident ccw edges  $p_2p$  and  $pp_3$ ). (Because cycles resulting after phase 1 are locally shortest,  $p$  must be reflex in one cycle and convex in the other.) Our local operation in this case results in a merging of the two cycles  $Q_1$  and  $Q_2$  into a single cycle, replacing edges  $p_0p$  (of  $Q_1$ ) and  $pp_3$  (of  $Q_2$ ) with the taut shortest path,  $\beta_{0,3}$ . As in the process described above, this replacement can result in new repeated vertices, as the merged cycle may come into contact with other cycles, or with itself.



# Technical Details: Merge Cycles (2)

Consider a weakly simple polygon  $Q$ , and let  $p$  be a vertex of  $Q$  which is repeated in the cycle specifying the boundary  $\partial Q$ . This implies that there are four edges of the (counterclockwise) cycle,  $p_0p$ ,  $pp_1$ ,  $p_2p$ , and  $pp_3$ , incident on  $p$ , all of which lie within a halfplane through  $p$  (by local optimality). There are then two subcases: (i)  $p_0, p, p_1$  is a left turn (Figure 9, left); and (ii)  $p_0, p, p_1$  is a right turn (Figure 9, right). In subcase (i),  $p_0p, pp_1$  define a left turn at  $p$  (making  $p$  locally convex for  $Q$ ), and  $p_2p, pp_3$  define a right turn at  $p$  (making  $p$  locally reflex for  $Q$ ). In this case, we replace the pair of edges  $p_0p, pp_1$  with a shorter polygonal chain, namely the “taut” version of this path, from  $p_0$  to  $p_1$ , along a shortest path,  $\beta_{0,1}$ , among the polygons  $Q_i$ , including  $Q$ , treating them as obstacles. The taut path  $\beta_{0,1}$  consists of left turns only, at (locally convex) vertices of polygons  $Q_i$  ( $Q_i \neq Q$ ) or (locally reflex) vertices of  $Q$ , where new pinch points of  $Q$  are created. Refer to Figure 9, left. Case (ii) is treated similarly; see Figure 9, right. Thus, resolving one repeated vertex,  $p$ , of  $Q$  can result in the creation of other repeated vertices of  $Q$ , or repeated vertices where two cycles come together (discussed below). The process is finite, though, since the total length of all cycles strictly decreases with each operation; in fact, there can be only a polynomial number of such adjustments, since each triple  $(p_0, p, p_1)$ , is resolved at most once.

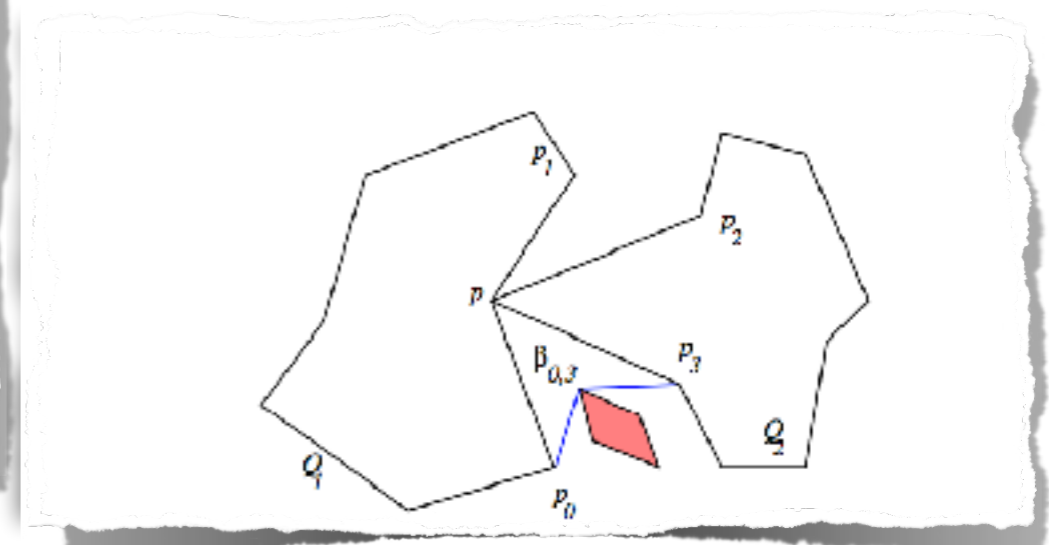
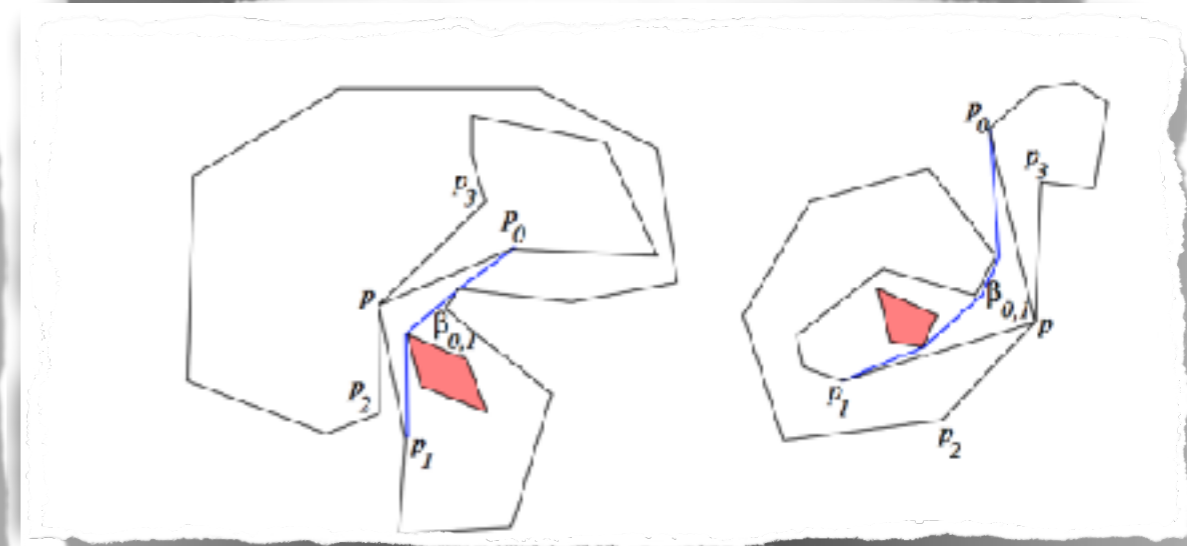
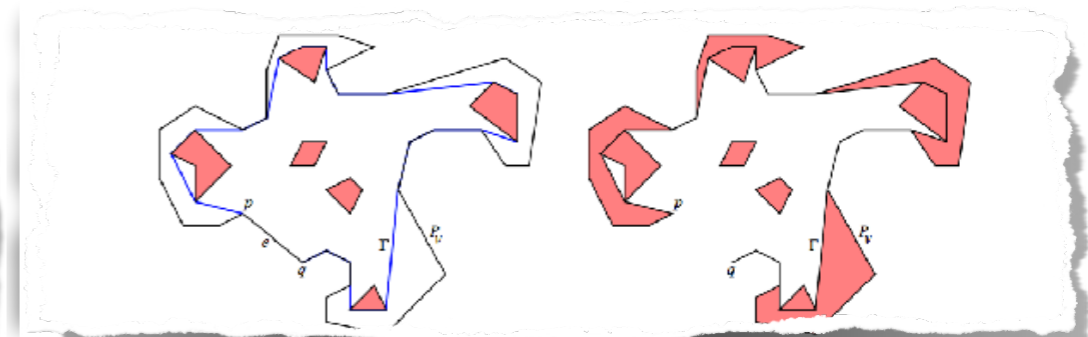
Now consider a vertex  $p$  that appears once as a reflex vertex in  $Q_1$  (with incident ccw edges  $p_0p$  and  $pp_1$ ) and once as a convex vertex in  $Q_2$  (with incident ccw edges  $p_2p$  and  $pp_3$ ). (Because cycles resulting after phase 1 are locally shortest,  $p$  must be reflex in one cycle and convex in the other.) Our local operation in this case results in a merging of the two cycles  $Q_1$  and  $Q_2$  into a single cycle, replacing edges  $p_0p$  (of  $Q_1$ ) and  $pp_3$  (of  $Q_2$ ) with the taut shortest path,  $\beta_{0,3}$ . As in the process described above, this replacement can result in new repeated vertices, as the merged cycle may come into contact with other cycles, or with itself.



# Technical Details: Merge Cycles (2)

Consider a weakly simple polygon  $Q$ , and let  $p$  be a vertex of  $Q$  which is repeated in the cycle specifying the boundary  $\partial Q$ . This implies that there are four edges of the (counterclockwise) cycle,  $p_0p$ ,  $pp_1$ ,  $p_2p$ , and  $pp_3$ , incident on  $p$ , all of which lie within a halfplane through  $p$  (by local optimality). There are then two subcases: (i)  $p_0, p, p_1$  is a left turn (Figure 9, left); and (ii)  $p_0, p, p_1$  is a right turn (Figure 9, right). In subcase (i),  $p_0p, pp_1$  define a left turn at  $p$  (making  $p$  locally convex for  $Q$ ), and  $p_2p, pp_3$  define a right turn at  $p$  (making  $p$  locally reflex for  $Q$ ). In this case, we replace the pair of edges  $p_0p, pp_1$  with a shorter polygonal chain, namely the “taut” version of this path, from  $p_0$  to  $p_1$ , along a shortest path,  $\beta_{0,1}$ , among the polygons  $Q_i$ , including  $Q$ , treating them as obstacles. The taut path  $\beta_{0,1}$  consists of left turns only, at (locally convex) vertices of polygons  $Q_i$  ( $Q_i \neq Q$ ) or (locally reflex) vertices of  $Q$ , where new pinch points of  $Q$  are created. Refer to Figure 9, left. Case (ii) is treated similarly; see Figure 9, right. Thus, resolving one repeated vertex,  $p$ , of  $Q$  can result in the creation of other repeated vertices of  $Q$ , or repeated vertices where two cycles come together (discussed below). The process is finite, though, since the total length of all cycles strictly decreases with each operation; in fact, there can be only a polynomial number of such adjustments, since each triple  $(p_0, p, p_1)$ , is resolved at most once.

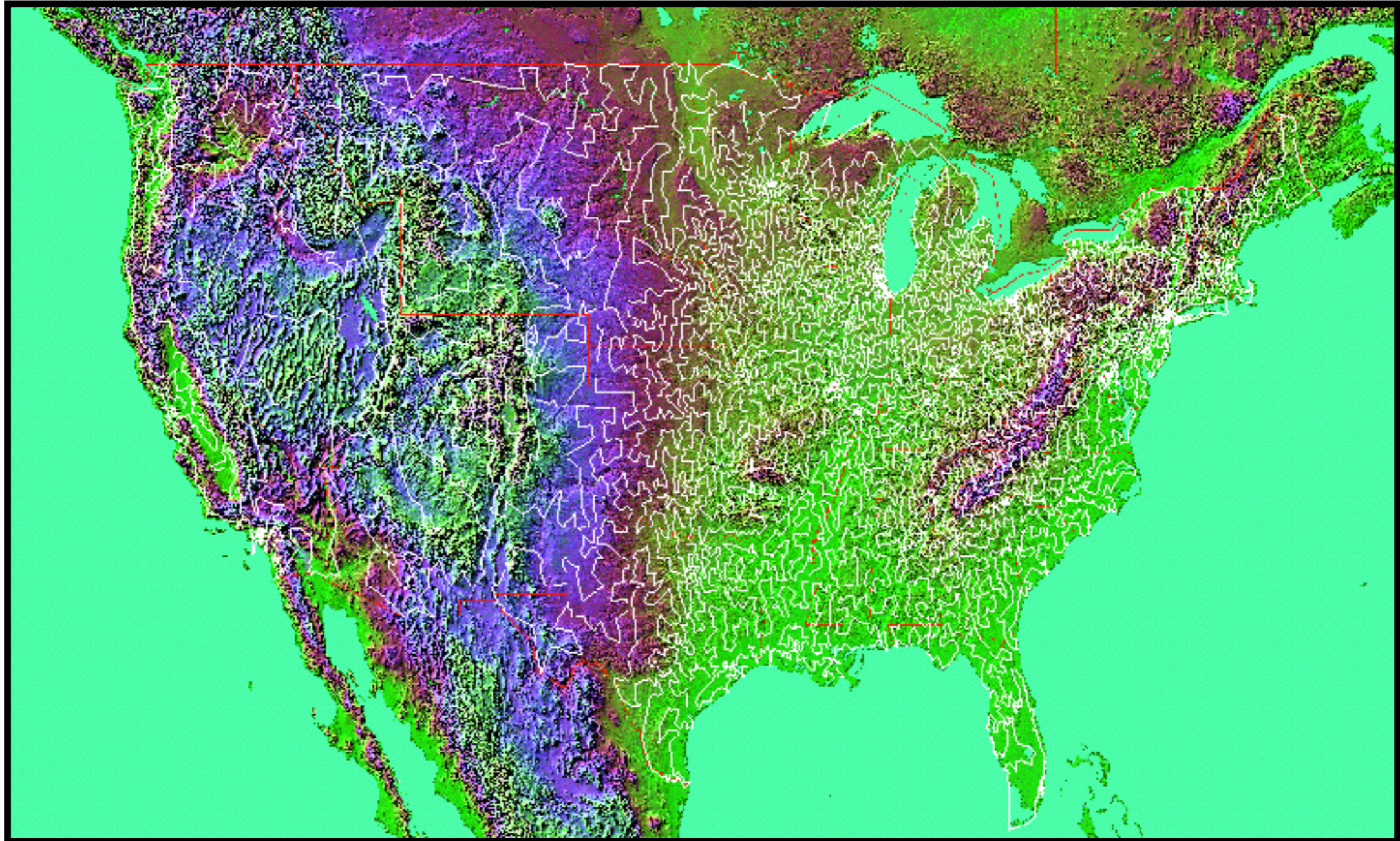
Now consider a vertex  $p$  that appears once as a reflex vertex in  $Q_1$  (with incident ccw edges  $p_0p$  and  $pp_1$ ) and once as a convex vertex in  $Q_2$  (with incident ccw edges  $p_2p$  and  $pp_3$ ). (Because cycles resulting after phase 1 are locally shortest,  $p$  must be reflex in one cycle and convex in the other.) Our local operation in this case results in a merging of the two cycles  $Q_1$  and  $Q_2$  into a single cycle, replacing edges  $p_0p$  (of  $Q_1$ ) and  $pp_3$  (of  $Q_2$ ) with the taut shortest path,  $\beta_{0,3}$ . As in the process described above, this replacement can result in new repeated vertices, as the merged cycle may come into contact with other cycles, or with itself.





# Overall Goals

# Overall Goals



# Overall Goals

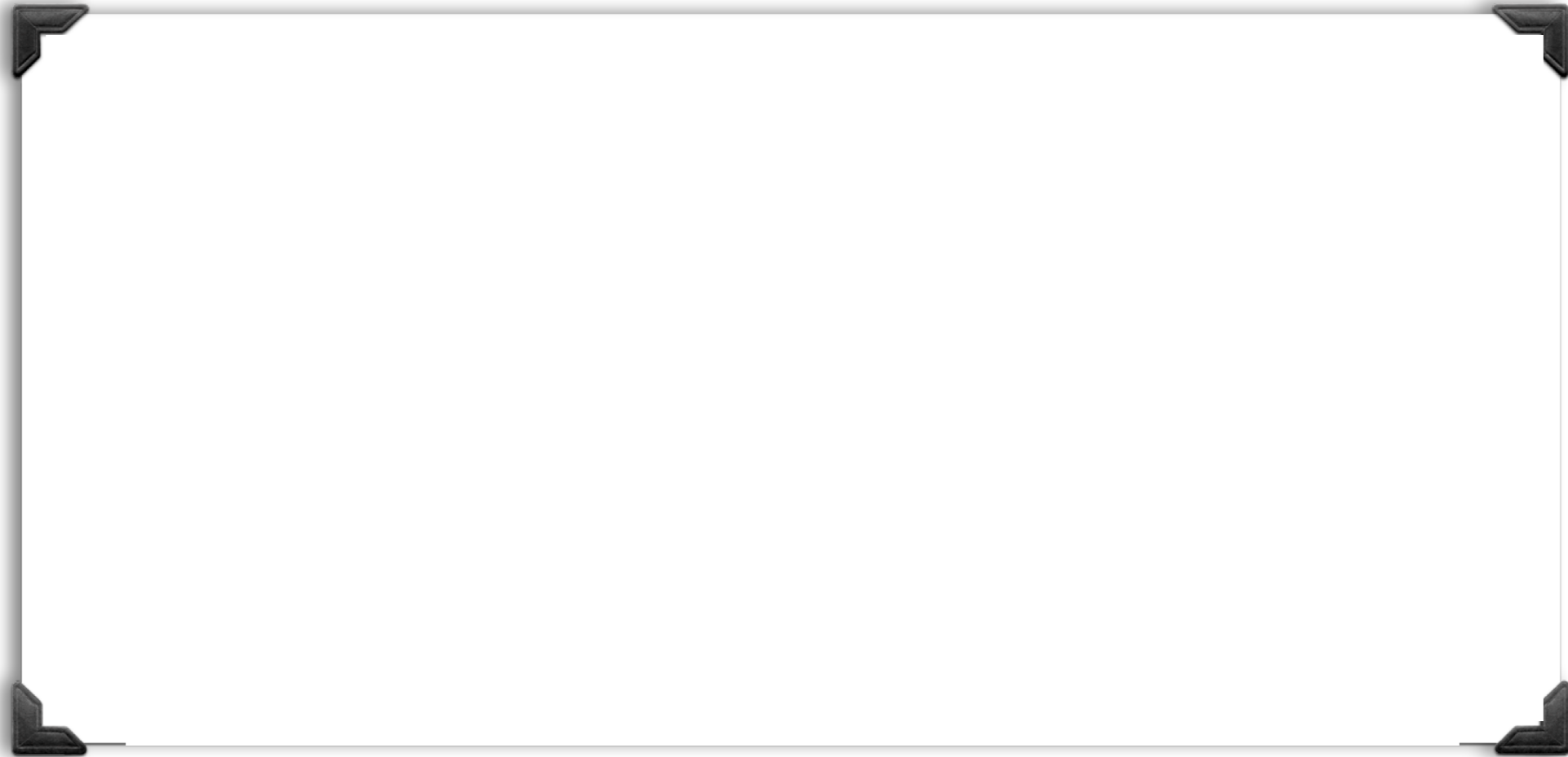
# Overall Goals



meets



# Subtour Elimination (TSP)



# Subtour Elimination (TSP)

$$x_e \in \{0, 1\}. \quad (4)$$

# Subtour Elimination (TSP)

$$\min \sum_{e \in E} x_e c_e. \quad (1)$$

$$x_e \in \{0, 1\}. \quad (4)$$

# Subtour Elimination (TSP)

$$\min \sum_{e \in E} x_e c_e. \quad (1)$$

$$\forall v \in V : \sum_{e \in \delta(v)} x_e = 2, \quad (2)$$

$$x_e \in \{0, 1\}. \quad (4)$$



# Subtour Elimination (TSP)

$$\min \sum_{e \in E} x_e c_e. \quad (1)$$

$$\forall v \in V : \sum_{e \in \delta(v)} x_e = 2, \quad (2)$$

$$\forall C \in \mathcal{C} : \sum_{e \in C} x_e \leq |C| - 1, \quad (3)$$

$$x_e \in \{0, 1\}. \quad (4)$$

# Subtour Elimination (TSP)

$$\min \sum_{e \in E} x_e c_e. \quad (1)$$

$$\forall v \in V : \sum_{e \in \delta(v)} x_e = 2, \quad (2)$$

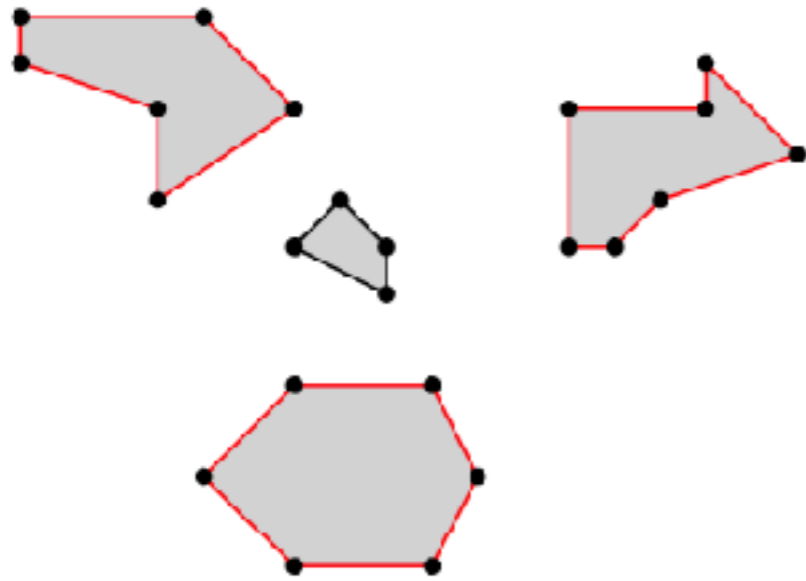
$$\forall C \in \mathcal{C} : \sum_{e \in C} x_e \leq |C| - 1, \quad (3)$$

$$x_e \in \{0, 1\}. \quad (4)$$

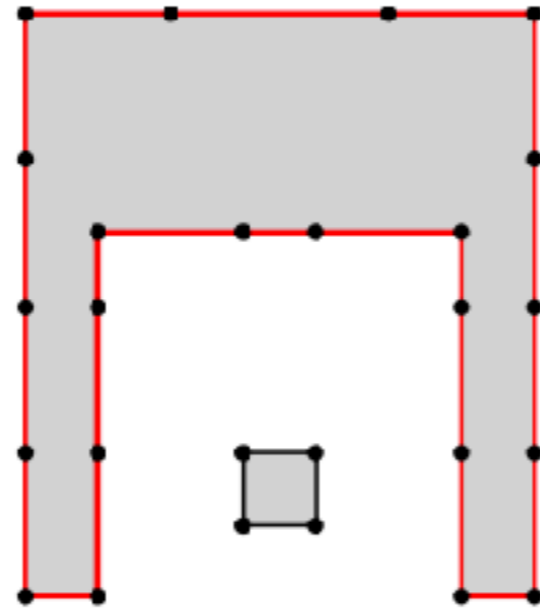
**Problem: Which cycles are illegal for MPP?**

# Illegal Cycles

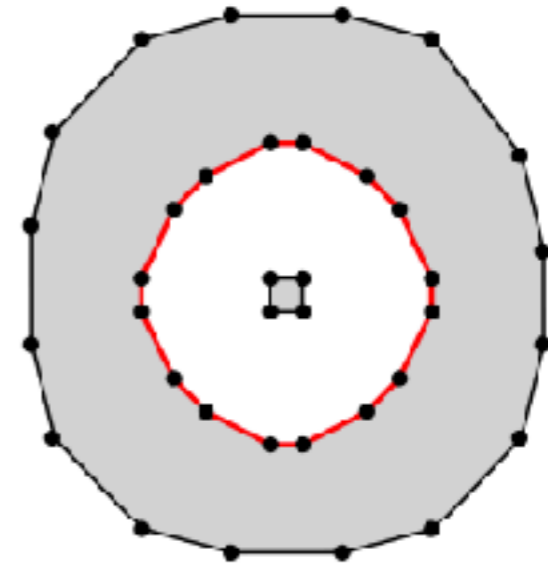
# Illegal Cycles



(a) Invalid cycle of type 1

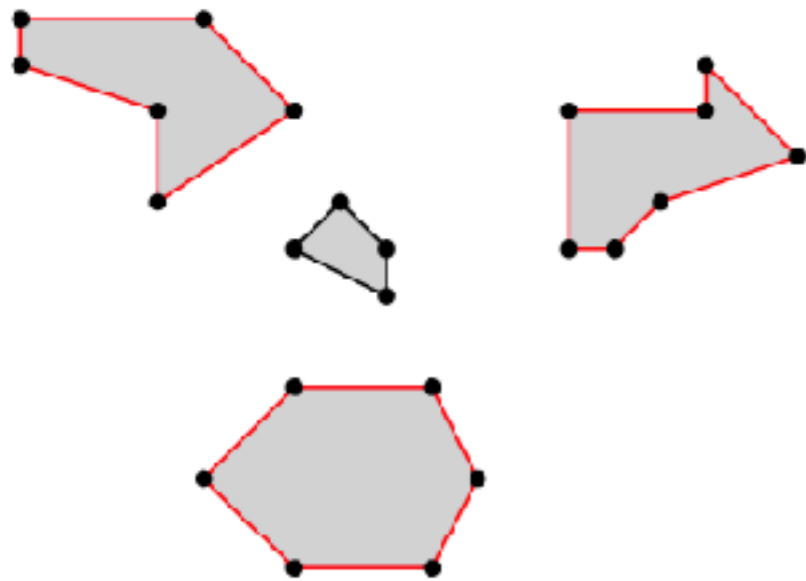


(b) Invalid cycle of type 2

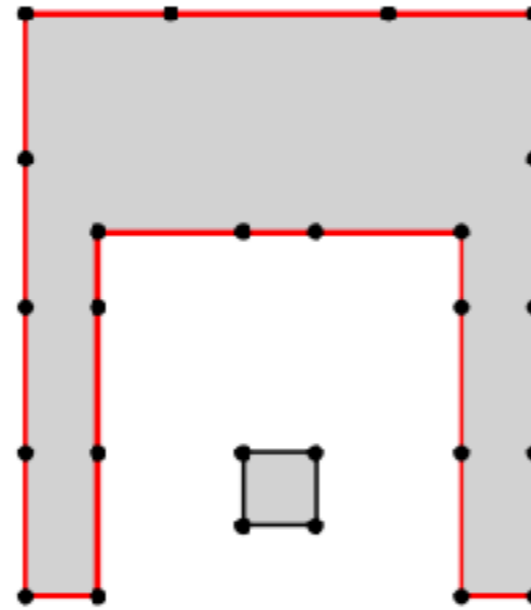


(c) Invalid cycle of type 3

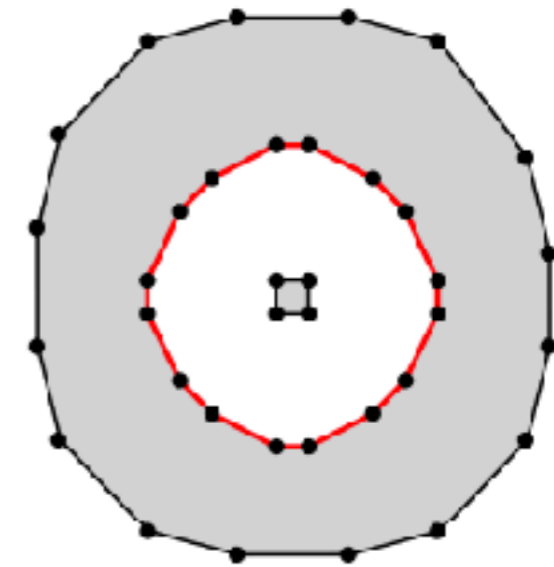
# Illegal Cycles



(a) Invalid cycle of type 1



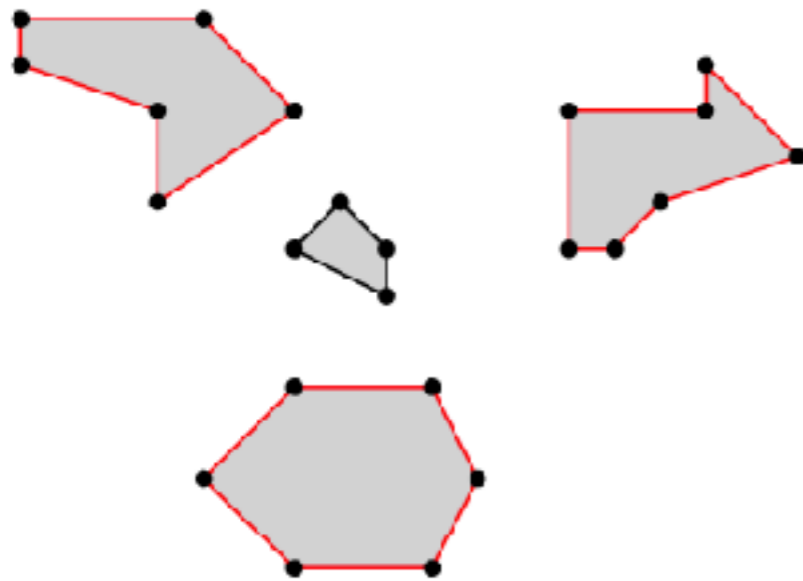
(b) Invalid cycle of type 2



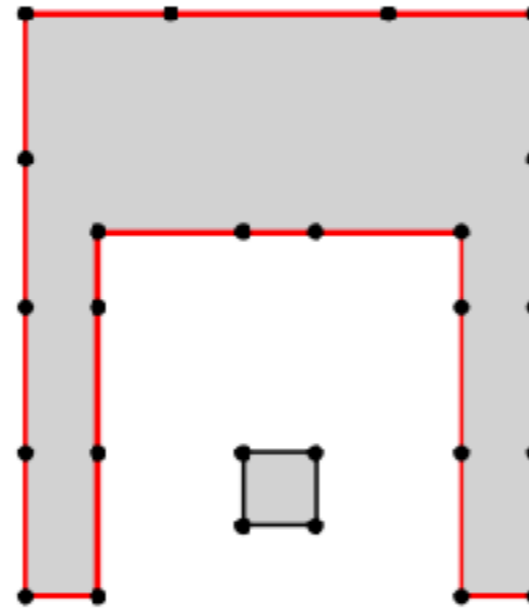
(c) Invalid cycle of type 3

1. at least one and at most  $|CH| - 1$  convex hull points. (See Figure 11(a))

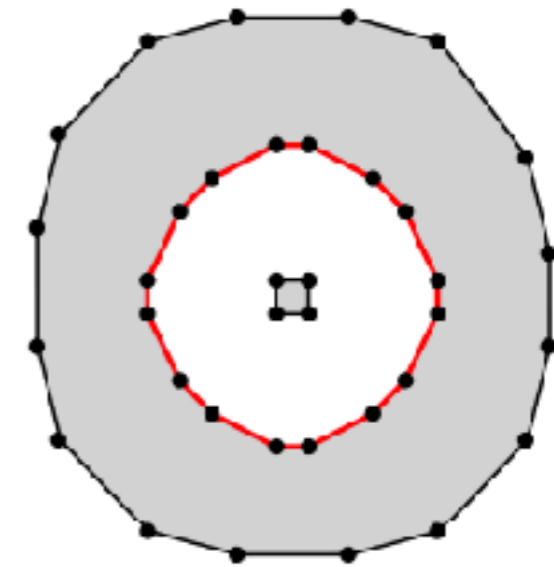
# Illegal Cycles



(a) Invalid cycle of type 1



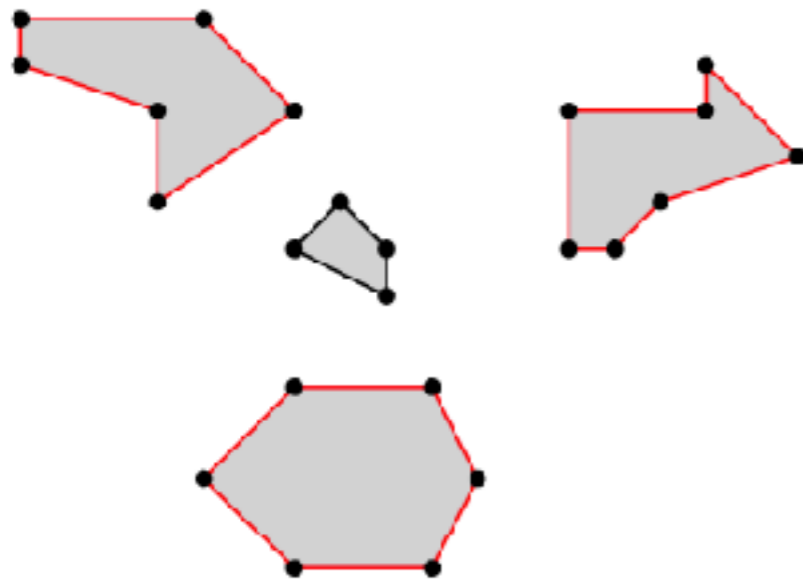
(b) Invalid cycle of type 2



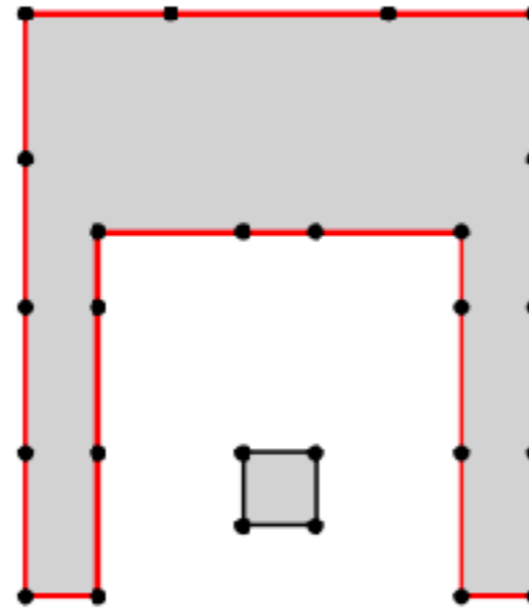
(c) Invalid cycle of type 3

1. at least one and at most  $|CH| - 1$  convex hull points. (See Figure 11(a))
2. all convex hull points but does not enclose all other points. (See Figure 11(b))

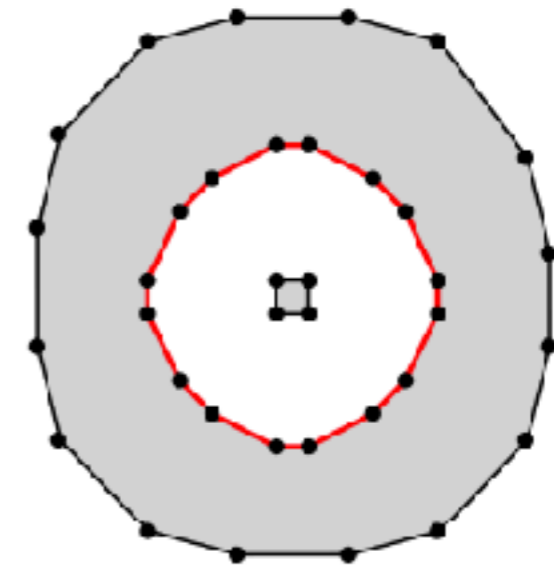
# Illegal Cycles



(a) Invalid cycle of type 1



(b) Invalid cycle of type 2



(c) Invalid cycle of type 3

1. at least one and at most  $|CH| - 1$  convex hull points. (See Figure 11(a))
2. all convex hull points but does not enclose all other points. (See Figure 11(b))
3. no convex hull point but encloses other points. (See Figure 11(c))

# Eliminating Cycles May Be Inefficient (1)



# Eliminating Cycles May Be Inefficient (1)

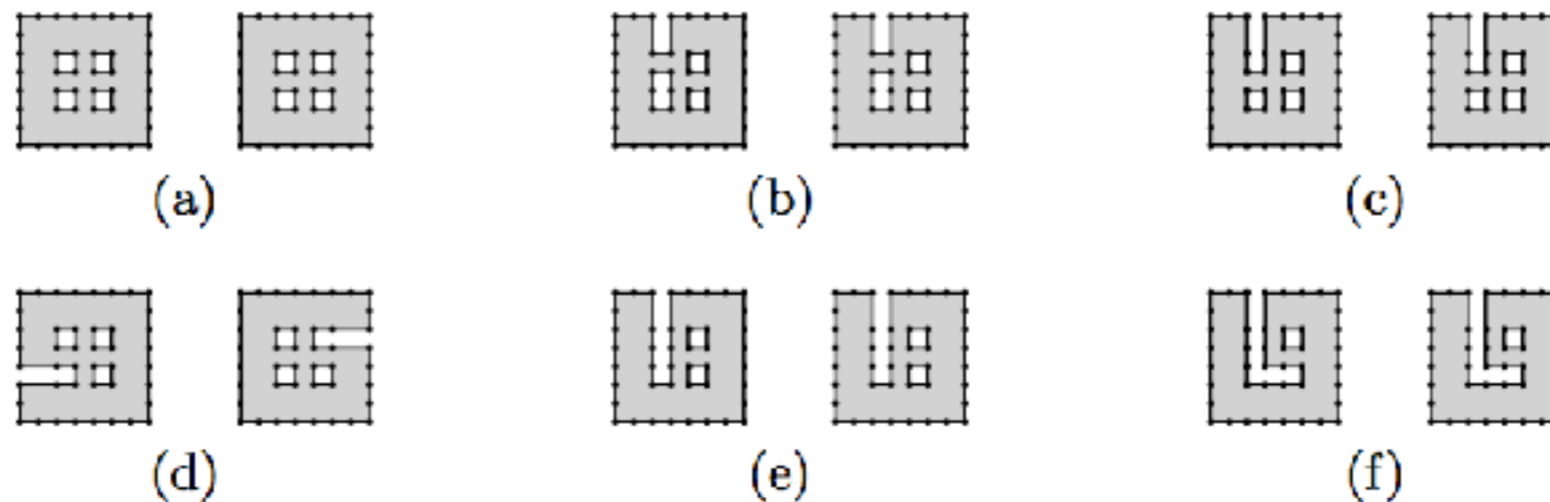
For an invalid cycle with property 1, we use the equivalent cut constraint

$$\forall C \in \mathcal{C}_1 : \sum_{e \in \delta(C)} x_e \geq 2. \quad (5)$$

# Eliminating Cycles May Be Inefficient (1)

For an invalid cycle with property 1, we use the equivalent cut constraint

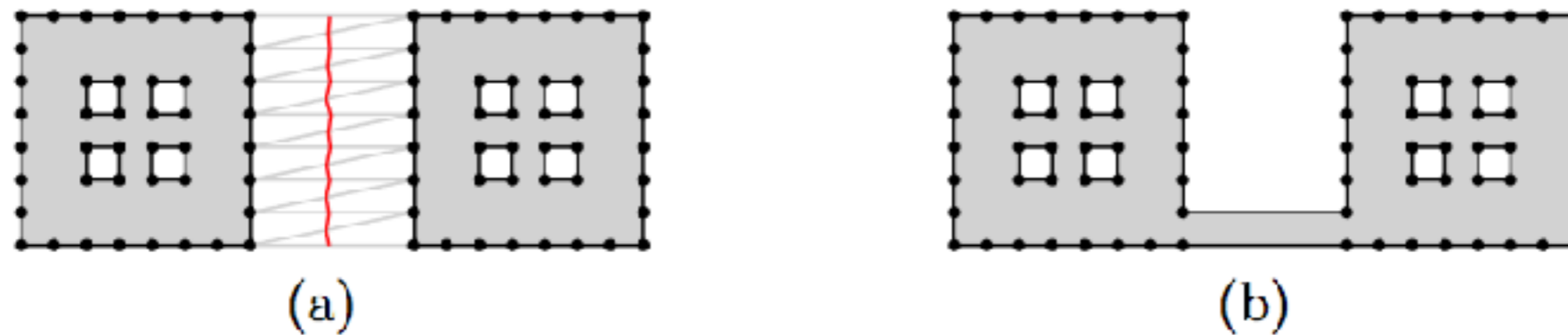
$$\forall C \in \mathcal{C}_1 : \sum_{e \in \delta(C)} x_e \geq 2. \quad (5)$$



**Figure 12.** (a) - (f) show consecutive iterations when trying to solve an instance using only constraint (5).

# More Efficient: Glue Cuts

# More Efficient: Glue Cuts



**Figure 15.** Solving instance from Figure 12 with a glue cut (red). (a) The red curve needs to be crossed at least twice; it is found using the Delaunay Triangulation (grey). (b) The first iteration after using the glue cut.

# TSPLIB Instances (Results)

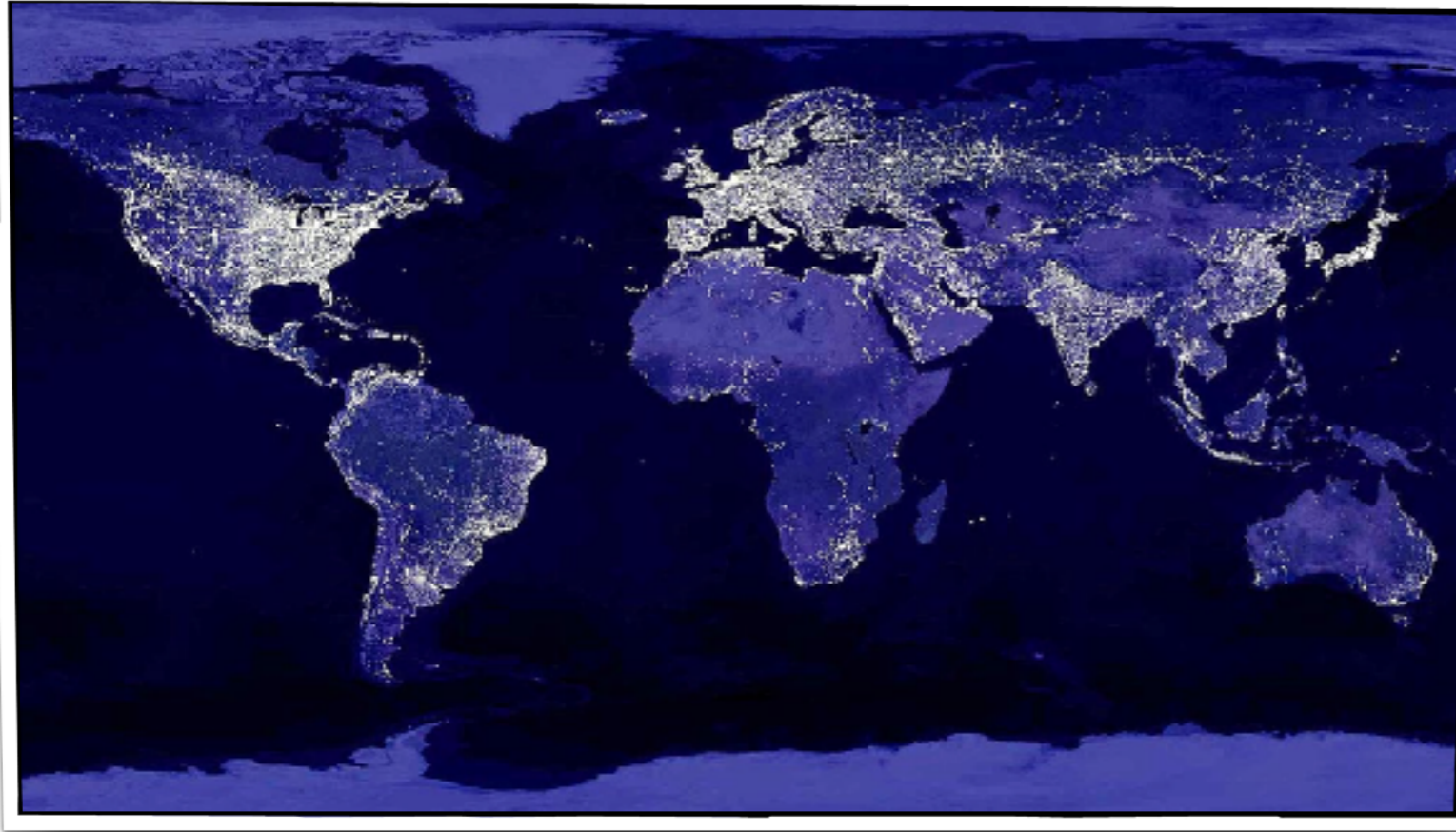
# TSPLIB Instances (Results)

	BasicIP	+JS+DC +TC+HIHC	+JS+TC +HIHC	+JS+DC +HIHC	+JS+DC +TC	+DC+TC +HIHC
burma14	20	22	17	19	26	19
ulysses16	48	42	35	43	32	42
ulysses22	50	34	55	31	32	61
att48	180	58	72	62	57	129
eil51	74	82	72	78	81	99
berlin52	43	38	37	37	38	51
st70	-	329	324	-	348	414
eil76	714	144	105	530	148	239
pr76	-	711	711	-	731	1238
gr96	376	388	349	10982	384	367
rat99	922	480	485	464	513	1190
kroA100	-	961	689	-	950	1294
kroB100	-	1470	2623	-	1489	2285
kroC100	-	470	431	-	465	577
kroD100	4673	509	451	4334	514	835
kroE100	-	273	273	-	272	574
rd100	-	894	756	-	890	2861
eil101	-	575	445	-	527	1090
lin105	-	390	359	-	412	931
pr107	550	401	272	346	513	923
pr124	495	348	264	322	355	940
bier127	439	288	270	267	276	476
ch130	-	1758	1802	-	1594	2853
pr136	1505	964	1029	992	950	3001
gr137	-	1262	1361	-	1252	1724
pr144	6276	1028	2926	985	1030	2012
ch150	-	4938	5167	-	5867	7997
kroA150	-	3427	5615	-	3327	7474
kroB150	-	2993	2396	-	2943	5265
pr152	13285	2161	1619	10978	2151	19479
u159	13285	1424	1262	5339	1410	2513
rat195	106030	16188	19780	77216	16117	27580

	BasicIP	+JS+DC +TC+HIHC	+JS+TC +HIHC	+JS+DC +HIHC	+JS+DC +TC	+DC+TC +HIHC
d198	-	19329	155550	-	19398	41118
kroA200	-	26360	13093	-	26389	11844
kroB200	-	5492	6239	-	5525	15238
gr202	-	4975	7512	-	4304	9670
ls225	18902	7746	9750	7595	7603	60167
tsp225	91423	11600	9741	28756	11531	44297
pr226	-	8498	2800	-	7204	18848
gr229	-	5462	26478	-	10153	25674
gil262	-	23000	22146	-	-	72772
pr264	24690	6537	-	6719	6549	23641
a280	22023	3601	3857	3980	3619	12983
pr299	-	16251	355323	-	16173	85789
lin318	-	23863	1511219	-	24035	75312
linhp318	-	23107	1313680	-	23064	79352
rd400	-	111128	92995	-	-	302363
fl417	-	198013	-	-	215210	825808
gr431	-	56716	173609	-	78133	265416
pr439	-	46685	36592	-	48231	273873
pcb442	-	1356796	-	-	-	-
d493	-	359072	-	-	-	837229
att532	-	217679	256394	-	218665	817096
ali535	-	93771	427800	-	91828	323104
u574	-	371523	199114	-	-	1010276
rat575	-	417494	191198	-	580320	934988
p654	-	864066	-	-	-	-
d657	-	455378	253374	-	646148	1352747
gr666	-	366157	-	-	670818	-

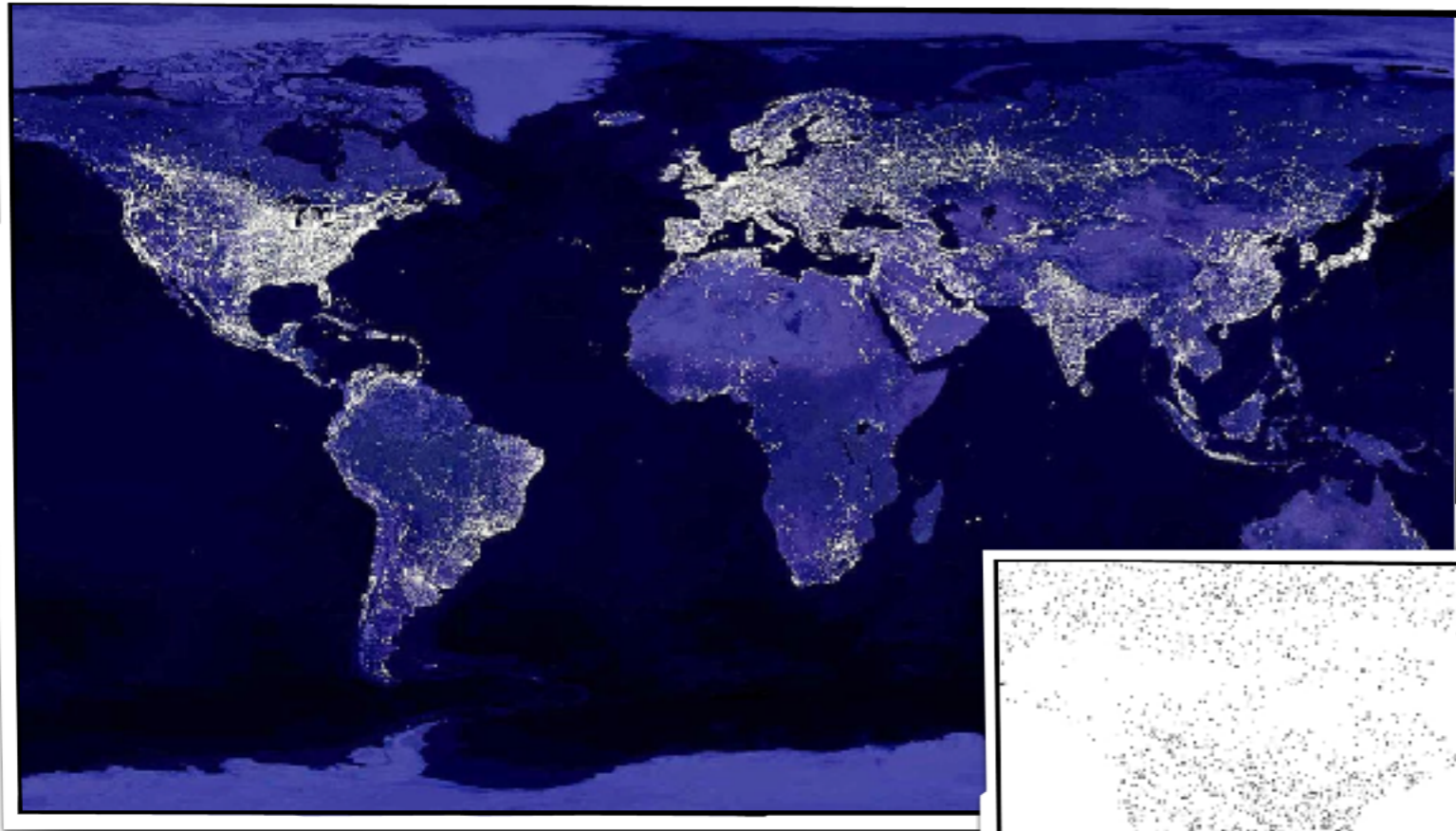
# Other Instances

# Other Instances



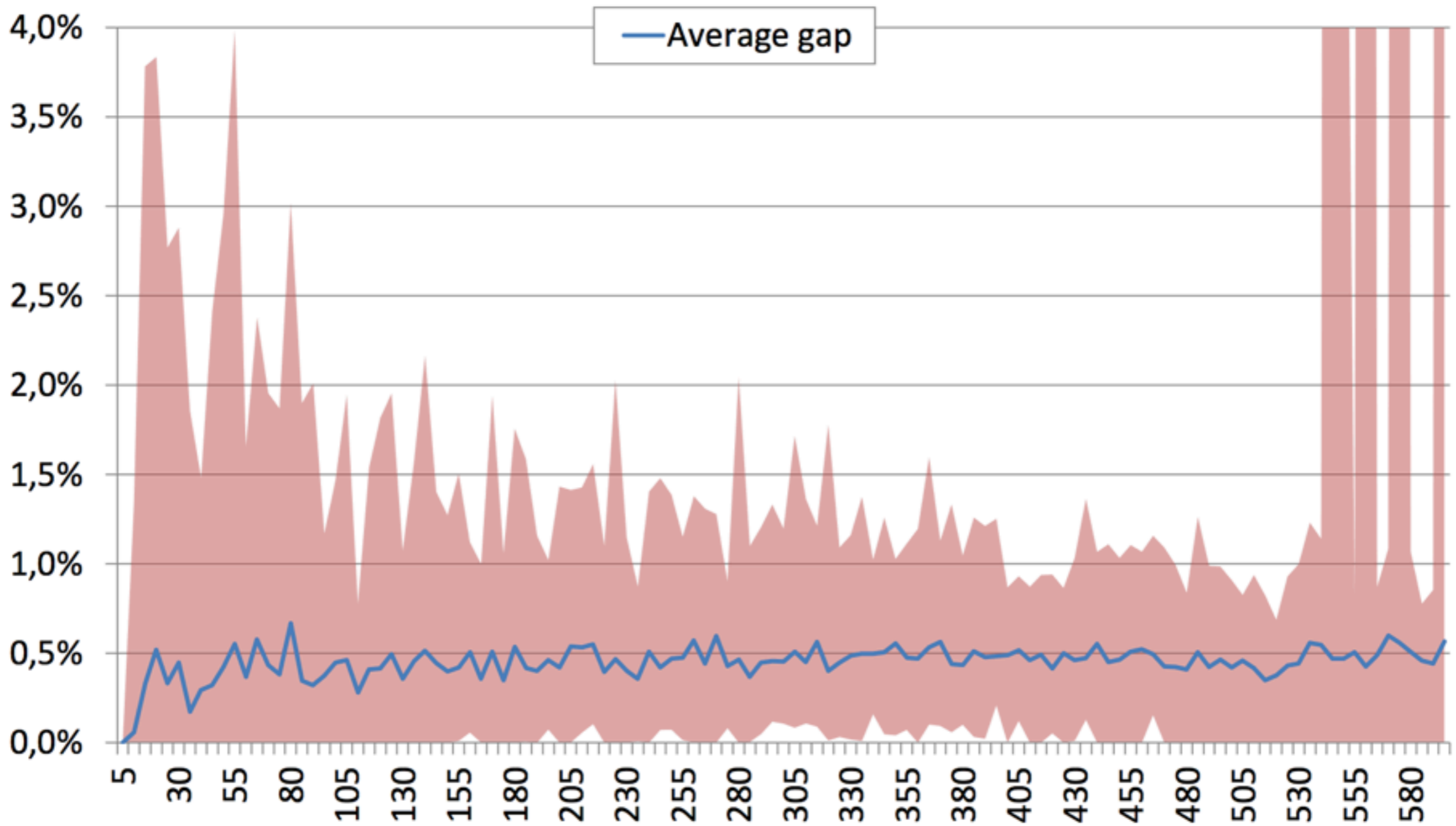


# Other Instances



# A Sparser Version: Delaunay Edges

# A Sparser Version: Delaunay Edges



- 1. Introduction**
- 2. Longest Tours**
- 3. Stars and Matchings**
- 4. Nonsimple Polygons**
- 5. Optimal Area**
- 6. Turn Cost**

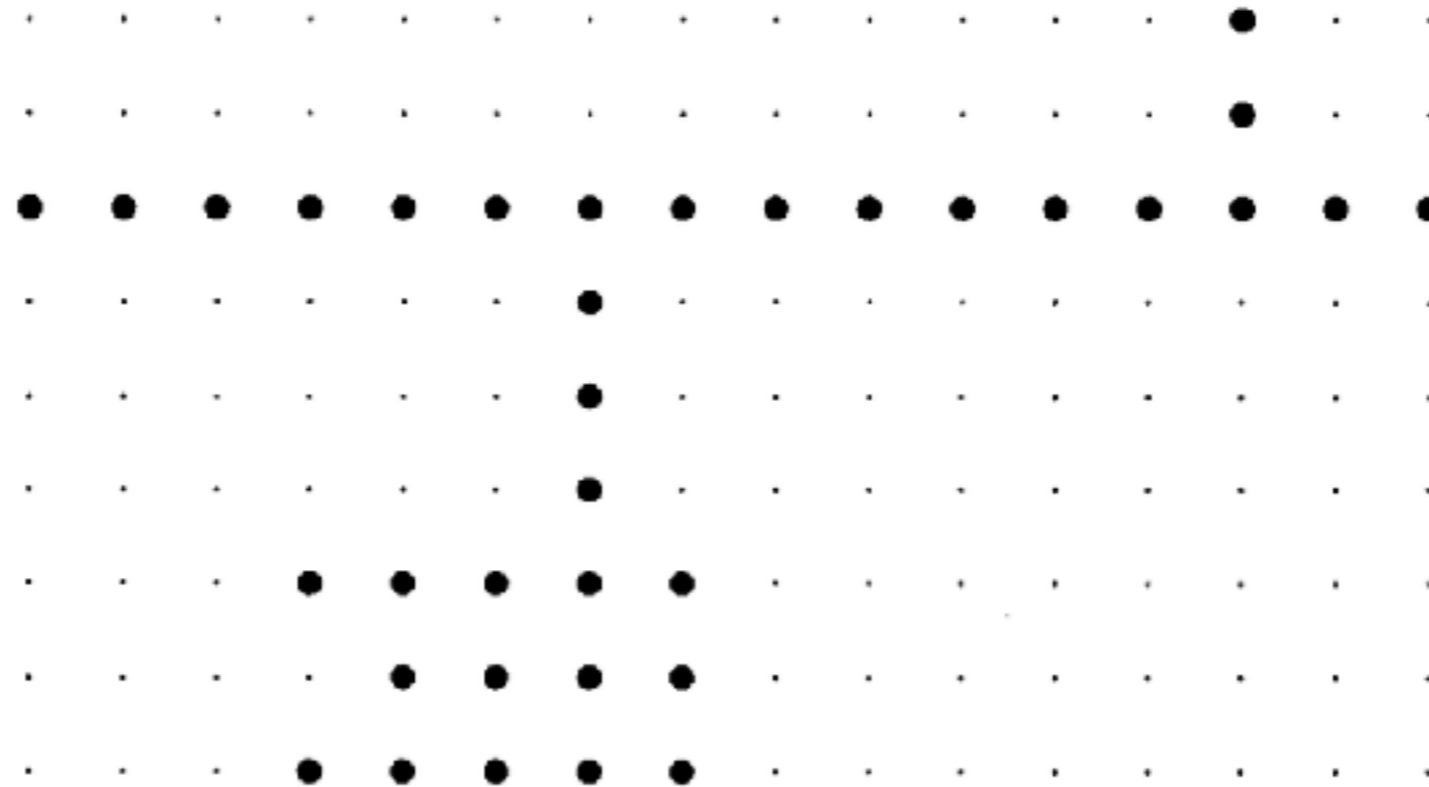
# Optimal Polygonizations

# Optimal Polygonizations

**Given:** A planar point set  $P$

# Optimal Polygonizations

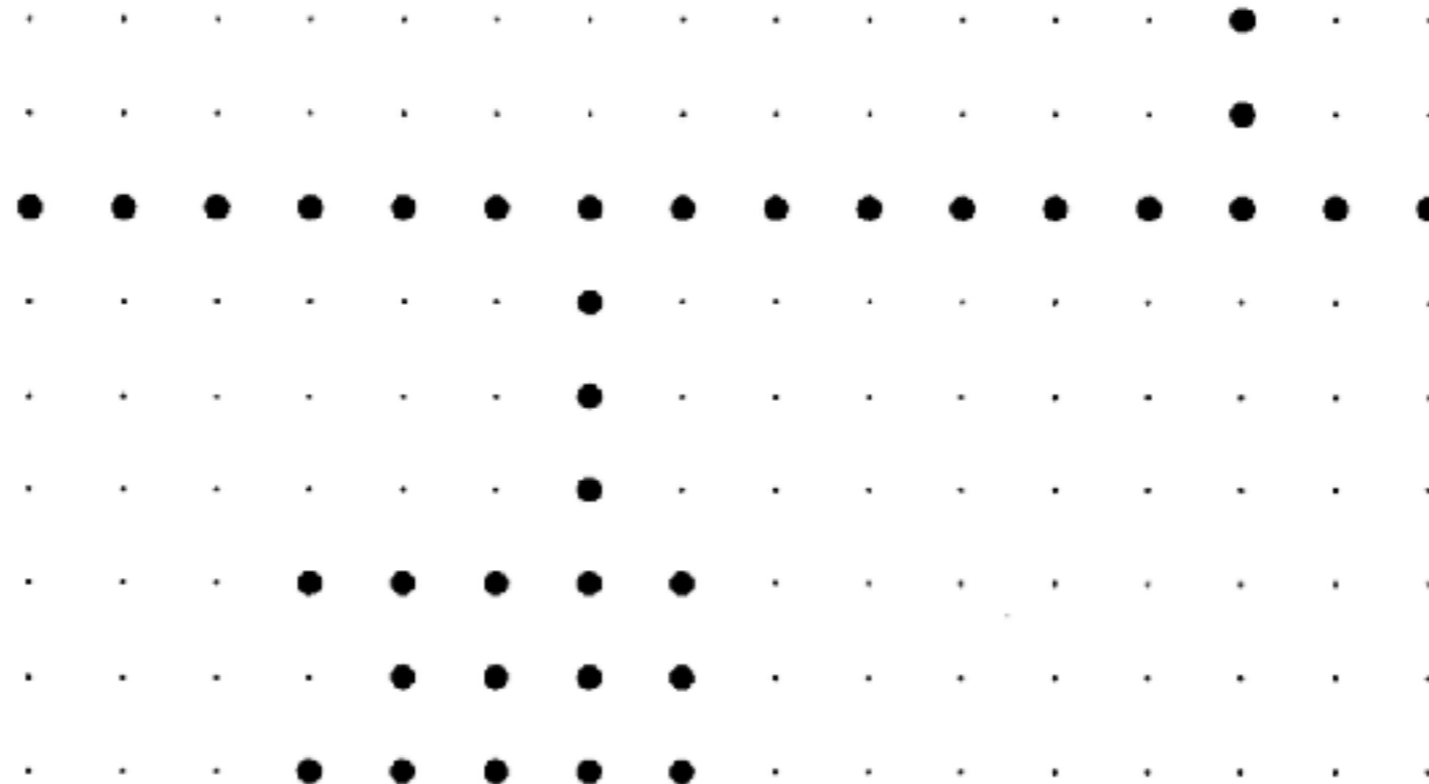
**Given:** A planar point set  $P$



# Optimal Polygonizations

**Given:** A planar point set  $P$

**Wanted:** A simple polygon with vertex set  $P$





# Optimal Polygonizations

**Given:** A planar point set  $P$

**Wanted:** A simple polygon with vertex set  $P$



# Optimal Polygonizations

**Given:** A planar point set  $P$

**Wanted:** A simple polygon with vertex set  $P$



**Objective:**

# Optimal Polygonizations

**Given:** A planar point set  $P$

**Wanted:** A simple polygon with vertex set  $P$



**Objective:** Minimize Perimeter

# Optimal Polygonizations

**Given:** A planar point set  $P$

**Wanted:** A simple polygon with vertex set  $P$



**Objective:** Minimize Perimeter



Traveling Salesman Problem

# Optimal Polygonizations

**Given:** A planar point set  $P$

**Wanted:** A simple polygon with vertex set  $P$



**Objective:**

# Optimal Polygonizations

**Given:** A planar point set  $P$

**Wanted:** A simple polygon with vertex set  $P$



**Objective:** Minimize Area!



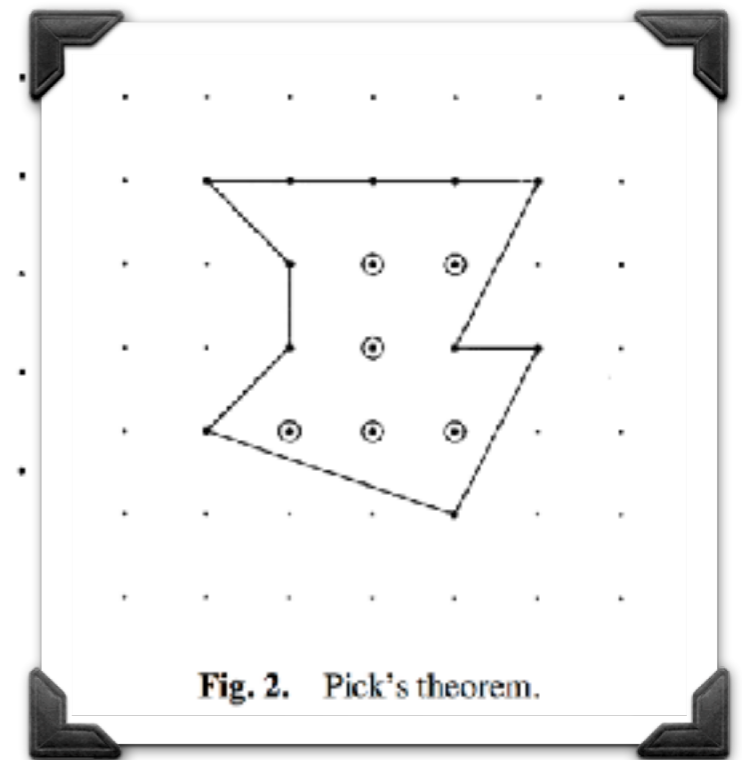
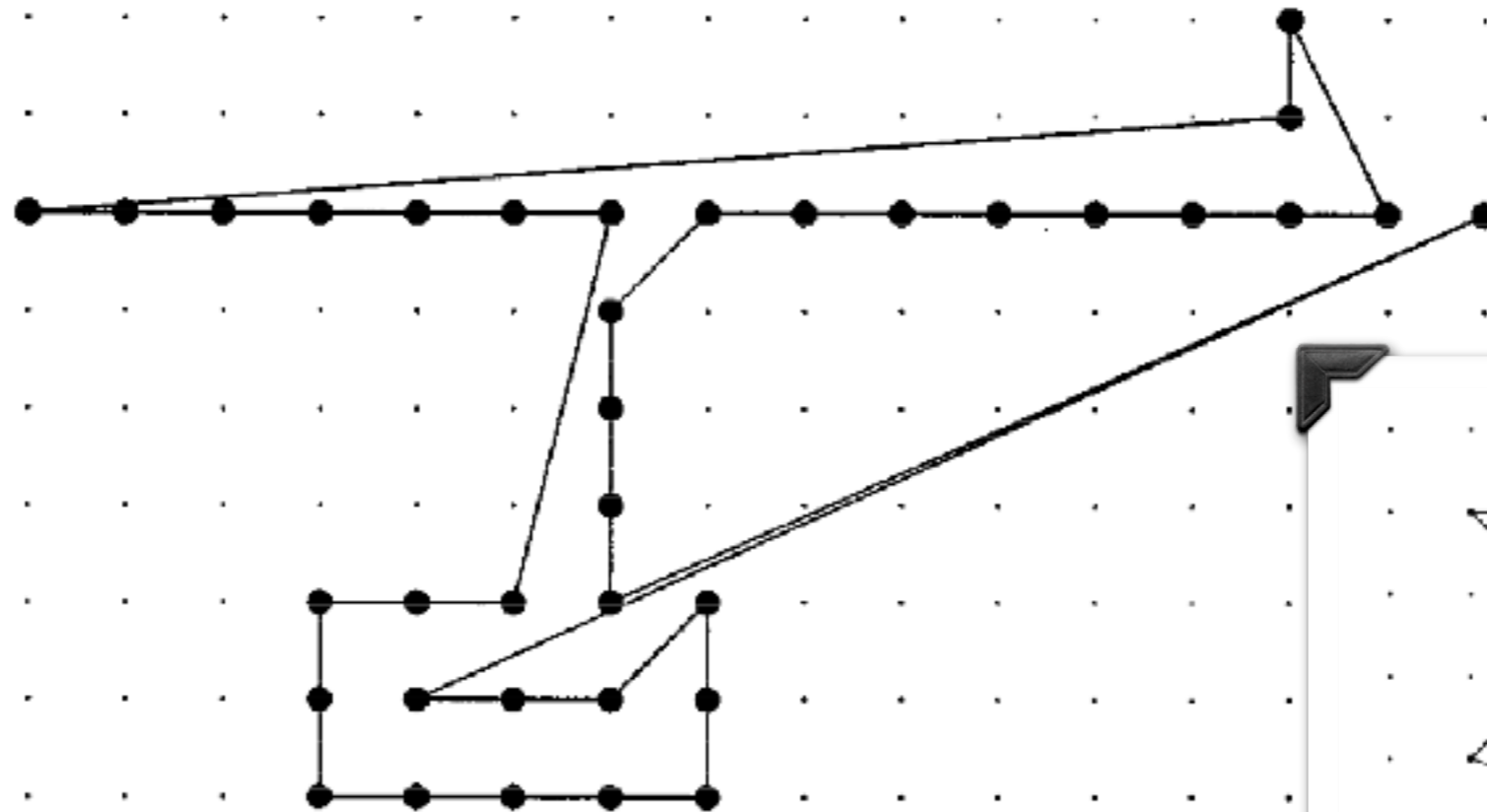




# Optimal Polygonizations

**Given:** A planar point set  $P$

**Wanted:** A simple polygon with vertex set  $P$



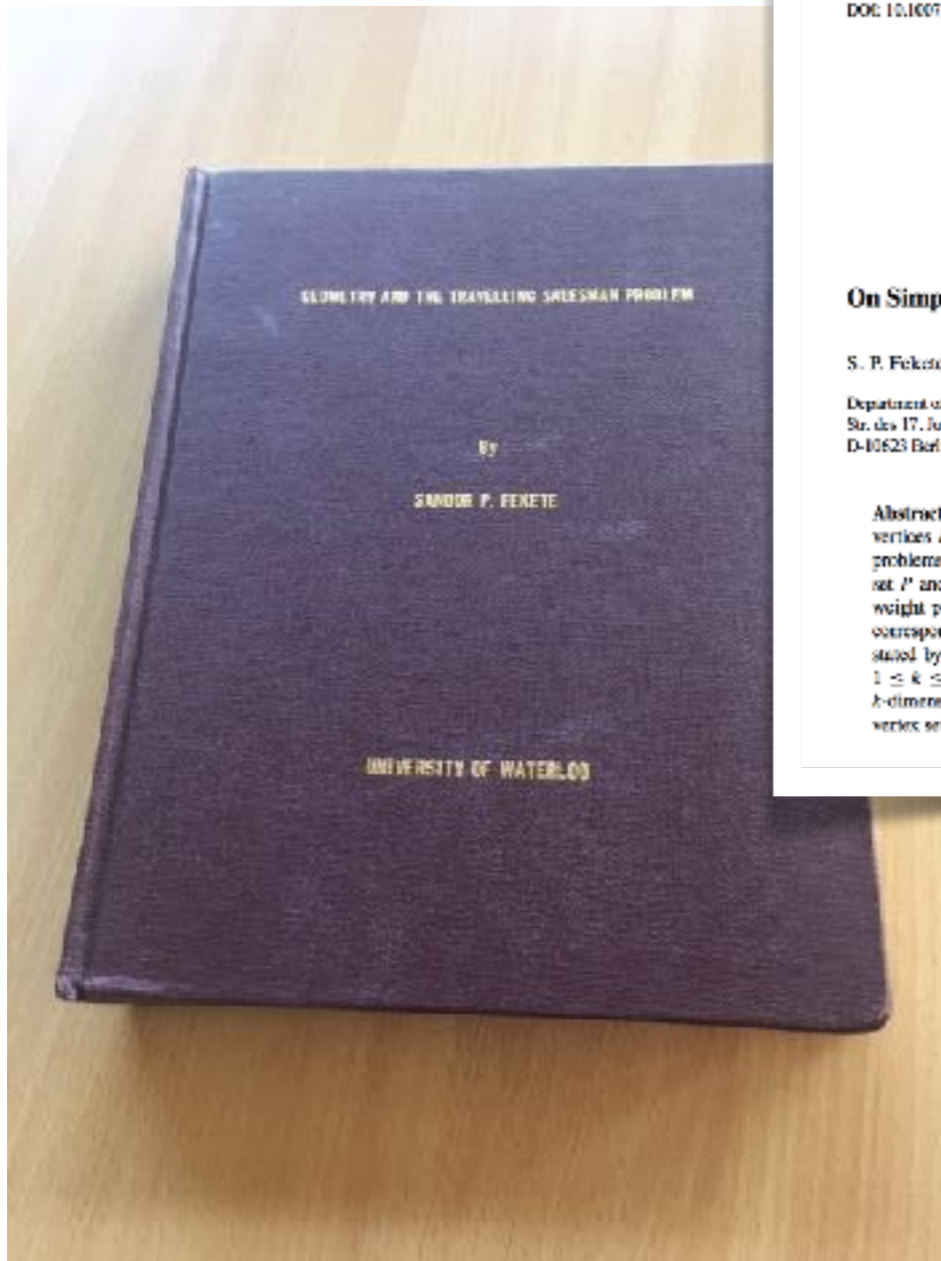
**Objective:** Minimize Area!

# Some History

# Some History



# Some History



Discrete Comput Geom 23:73–110 (2000)  
DOI: 10.1007/s00454910005

Discrete & Computational  
**Geometry**  
© 2000 Springer-Verlag New York, Inc.

### On Simple Polygonizations with Optimal Area\*

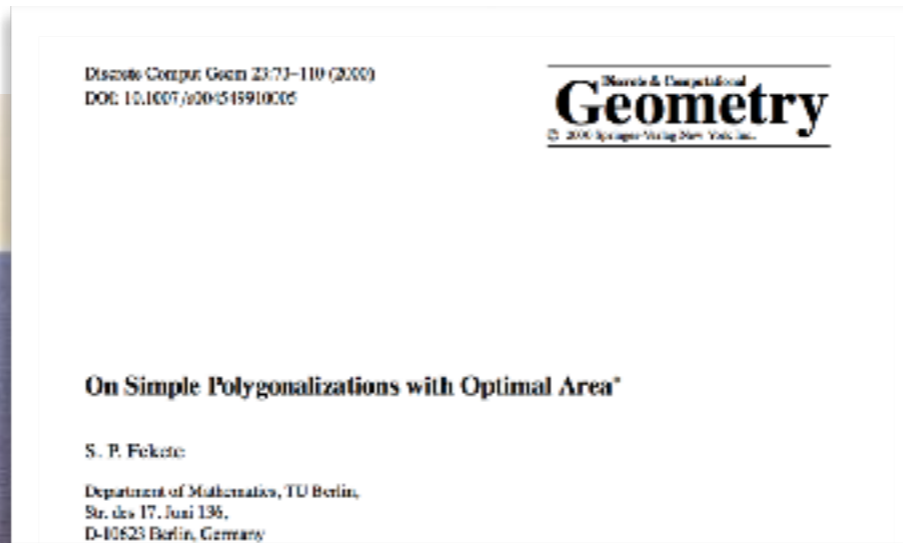
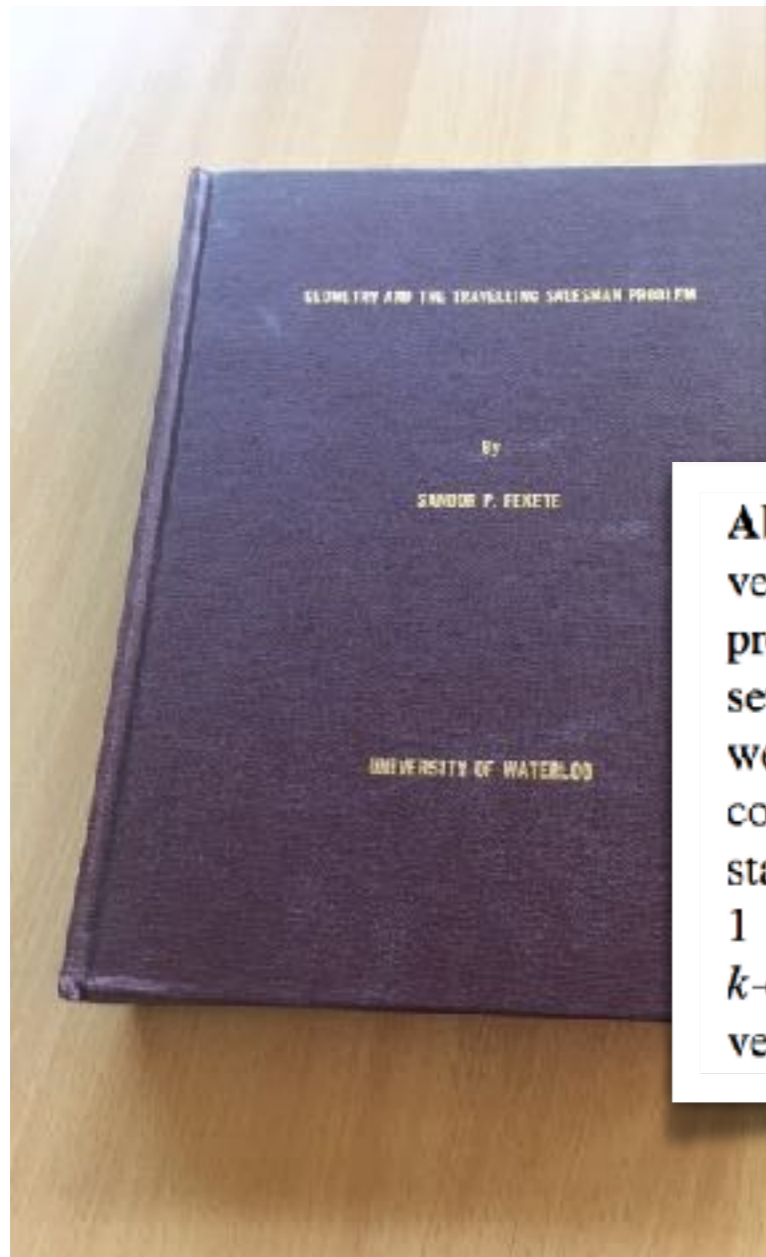
S. P. Fekete

Department of Mathematics, TU Berlin,  
Str. des 17. Juni 136,  
D-10623 Berlin, Germany

**Abstract.** We discuss the problem of finding a simple polygonization for a given set of vertices  $P$  that has optimal area. We show that these problems are very closely related to problems of optimizing the number of points from a set  $Q$  in a simple polygon with vertex set  $P$  and prove that it is NP-complete to find a minimum weight polygon or a maximum weight polygon for a given vertex set, resulting in a proof of NP-completeness for the corresponding area optimization problems. This answers a generalization of a question stated by Suri in 1989. Finally, we turn to higher dimensions, where we prove that, for  $1 \leq k \leq d$ ,  $2 \leq d$ , it is NP-hard to determine the smallest possible total volume of the  $k$ -dimensional faces of a  $d$ -dimensional simple nondegenerate polyhedron with a given vertex set, answering a generalization of a question stated by O'Rourke in 1990.

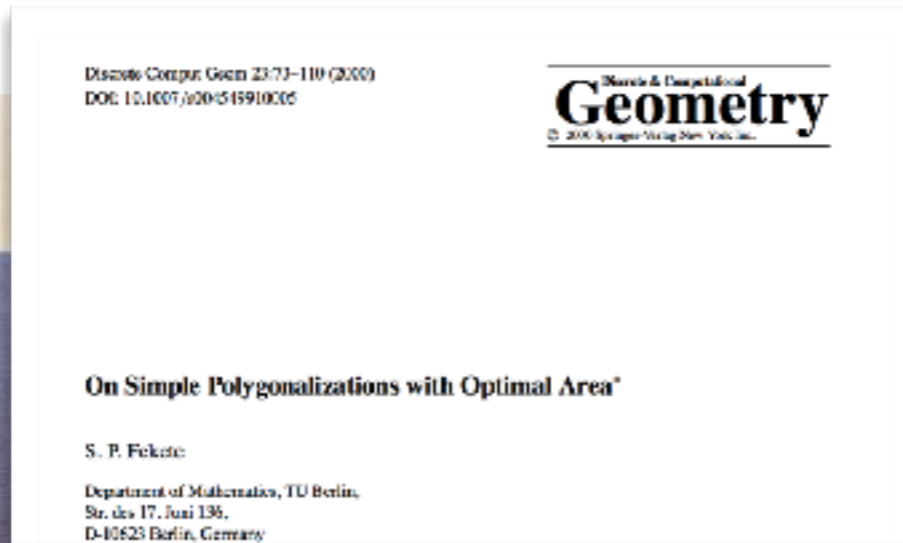
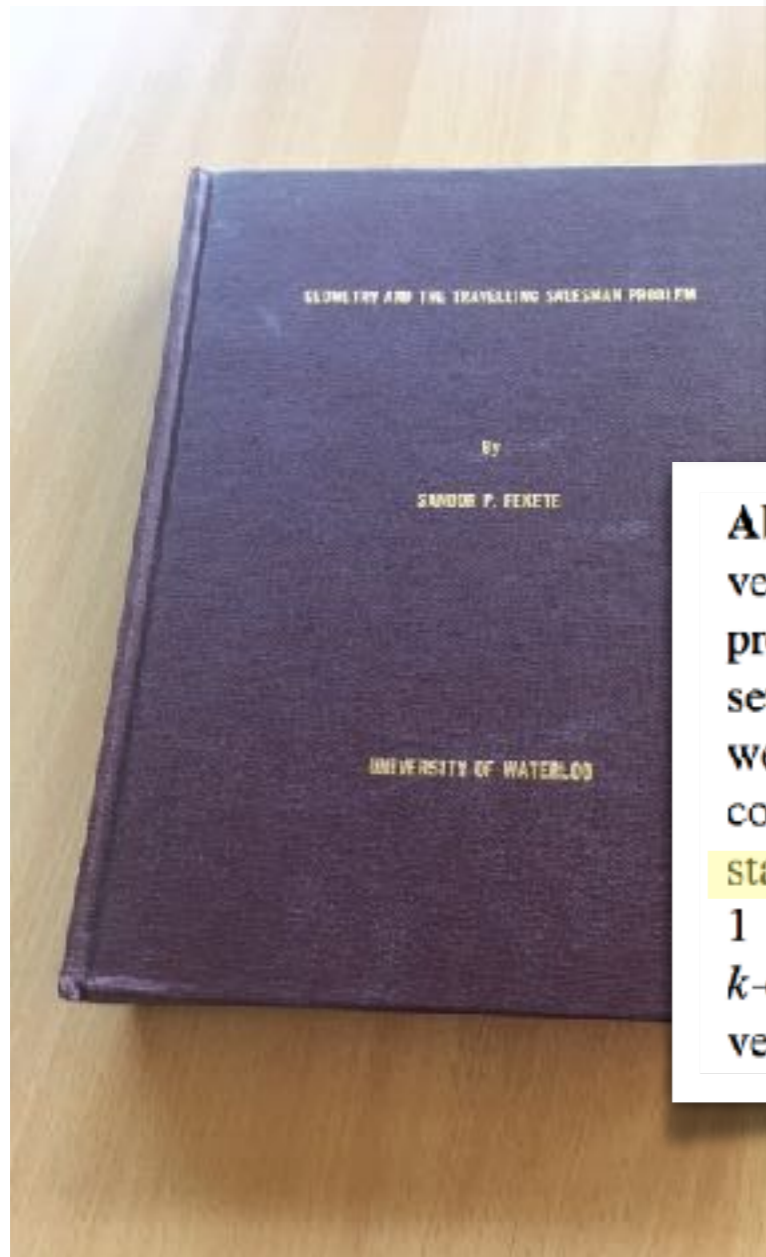


# Some History



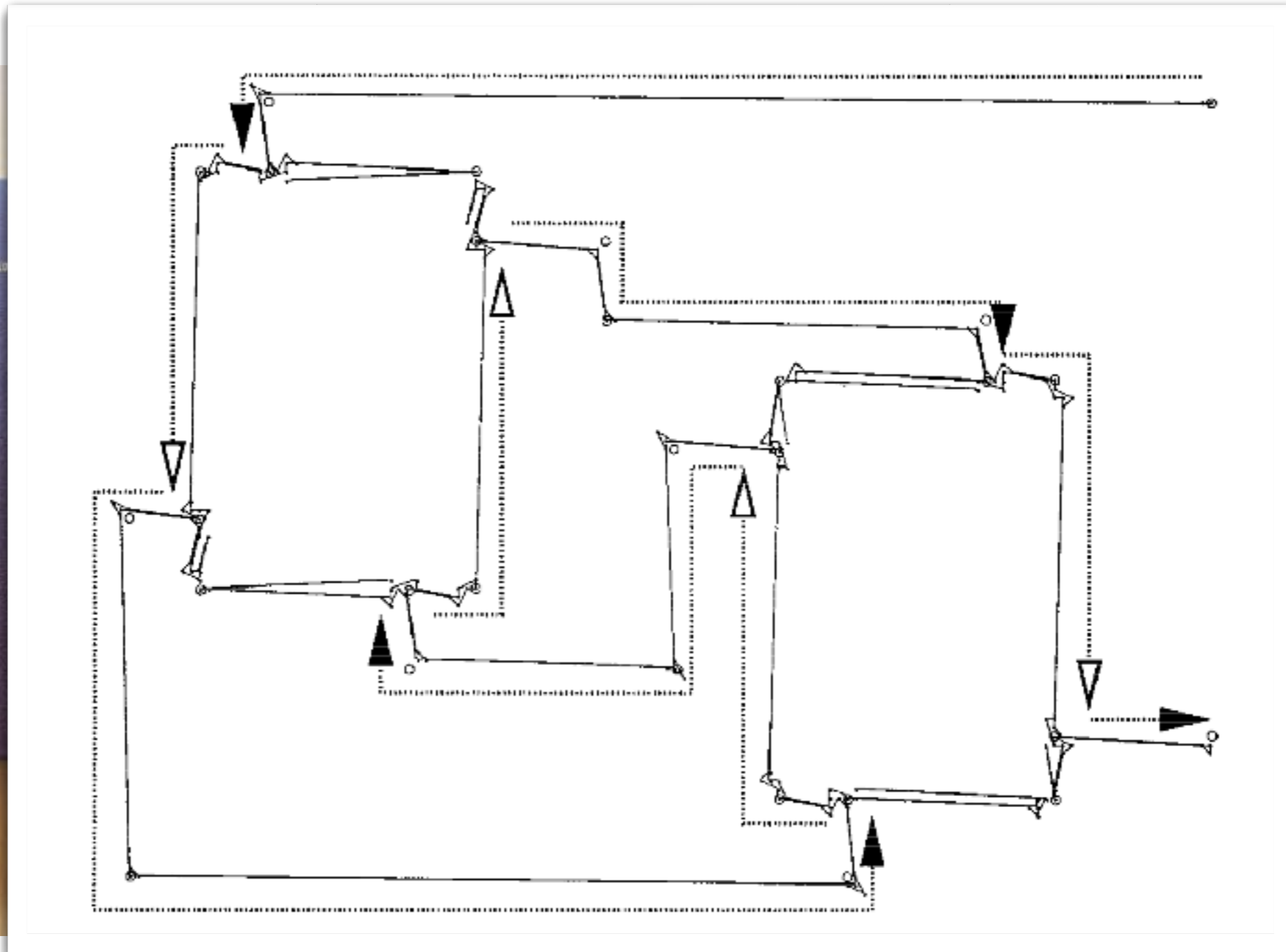
**Abstract.** We discuss the problem of finding a simple polygonalization for a given set of vertices  $P$  that has optimal area. We show that these problems are very closely related to problems of optimizing the number of points from a set  $Q$  in a simple polygon with vertex set  $P$  and prove that it is NP-complete to find a minimum weight polygon or a maximum weight polygon for a given vertex set, resulting in a proof of NP-completeness for the corresponding area optimization problems. This answers a generalization of a question stated by Suri in 1989. Finally, we turn to higher dimensions, where we prove that, for  $1 \leq k \leq d$ ,  $2 \leq d$ , it is NP-hard to determine the smallest possible total volume of the  $k$ -dimensional faces of a  $d$ -dimensional simple nondegenerate polyhedron with a given vertex set, answering a generalization of a question stated by O'Rourke in 1980.

# Some History



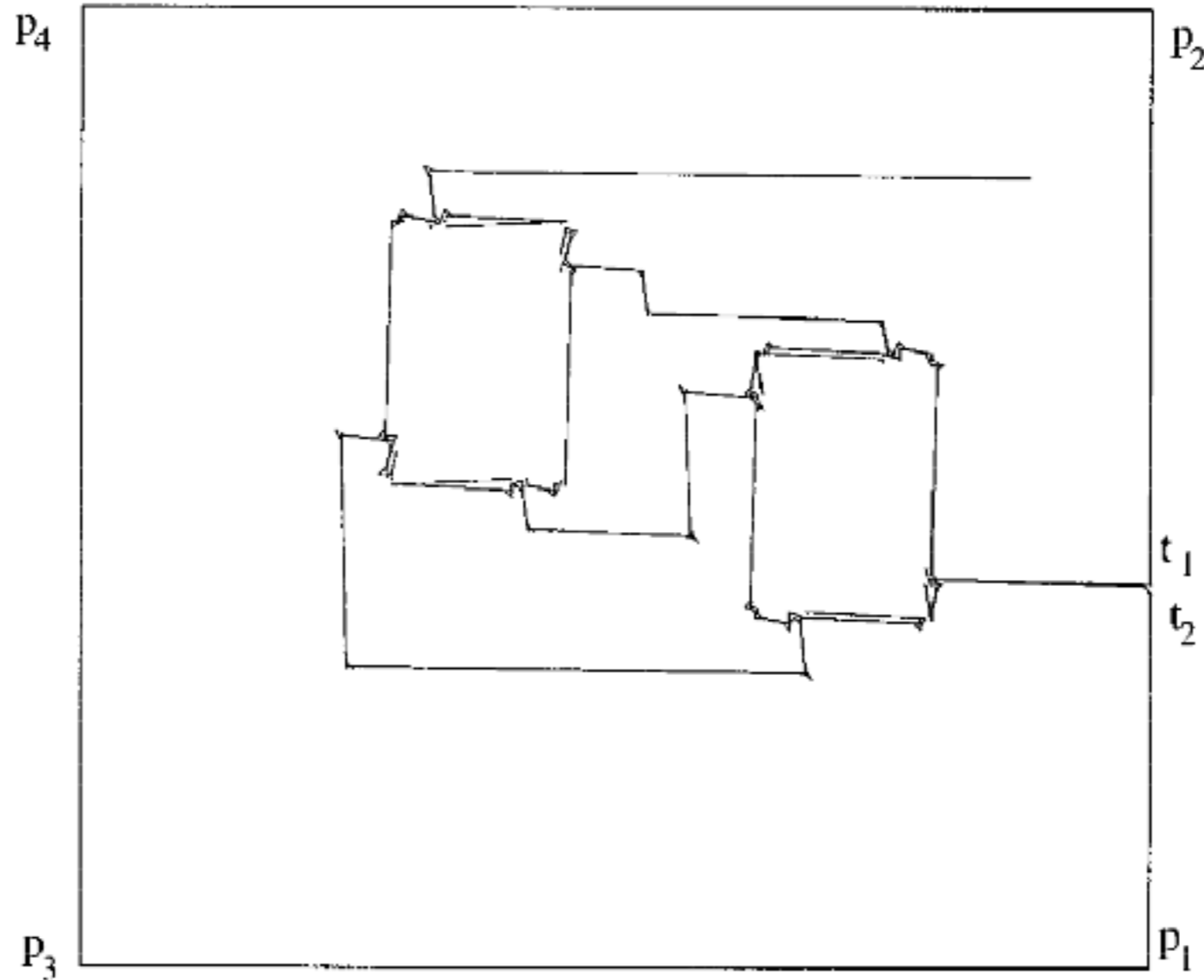
**Abstract.** We discuss the problem of finding a simple polygonalization for a given set of vertices  $P$  that has optimal area. We show that these problems are very closely related to problems of optimizing the number of points from a set  $Q$  in a simple polygon with vertex set  $P$  and prove that it is NP-complete to find a minimum weight polygon or a maximum weight polygon for a given vertex set, resulting in a proof of NP-completeness for the corresponding area optimization problems. This answers a generalization of a question stated by Suri in 1989. Finally, we turn to higher dimensions, where we prove that, for  $1 \leq k \leq d$ ,  $2 \leq d$ , it is NP-hard to determine the smallest possible total volume of the  $k$ -dimensional faces of a  $d$ -dimensional simple nondegenerate polyhedron with a given vertex set, answering a generalization of a question stated by O'Rourke in 1980.

# Some History



a given set of  
ely related to  
n with vertex  
a maximum  
eness for the  
of a question  
rove that, for  
olume of the  
with a given  
80.

# Some History

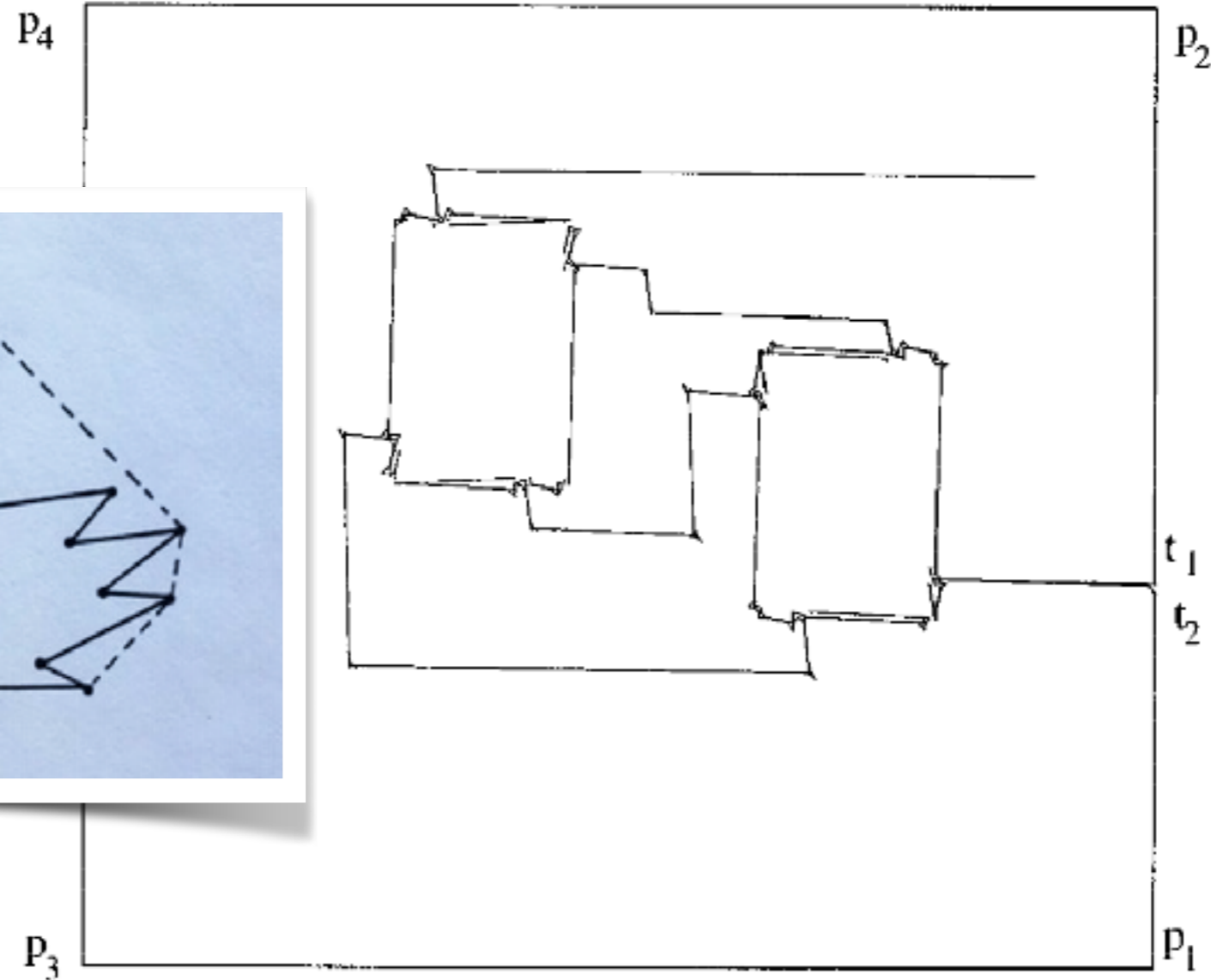
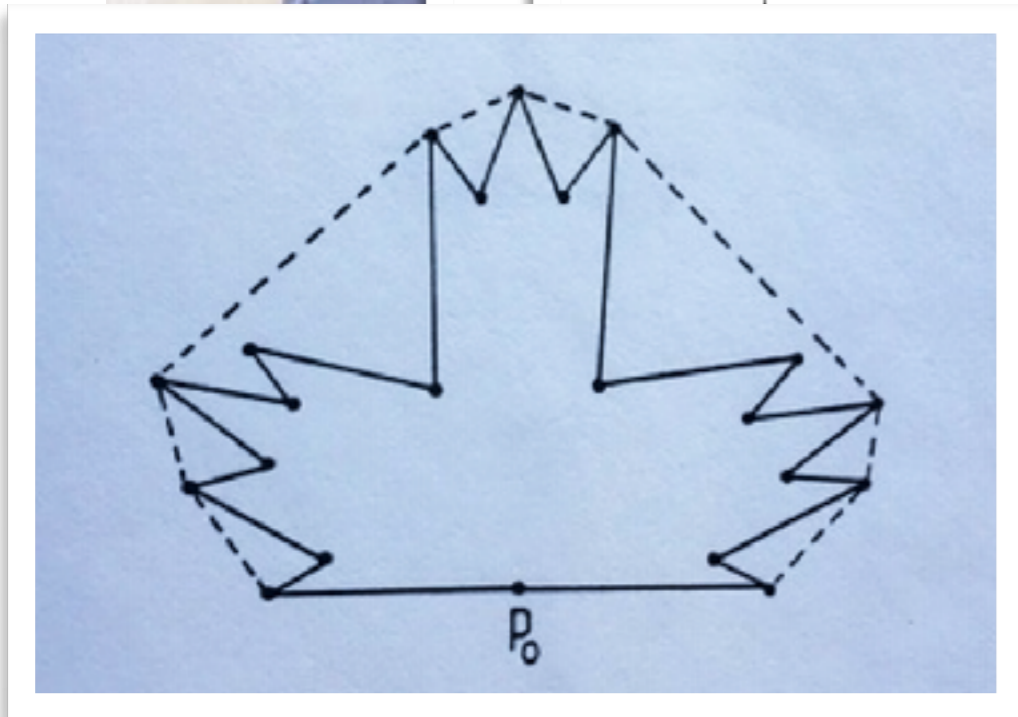


**Fig. 24.** GRID-FULL solves GRID-EMPTY: turning a small polygon inside out.

a given set of  
ely related to  
n with vertex  
a maximum  
ness for the  
of a question  
rove that, for  
olume of the  
with a given  
80.



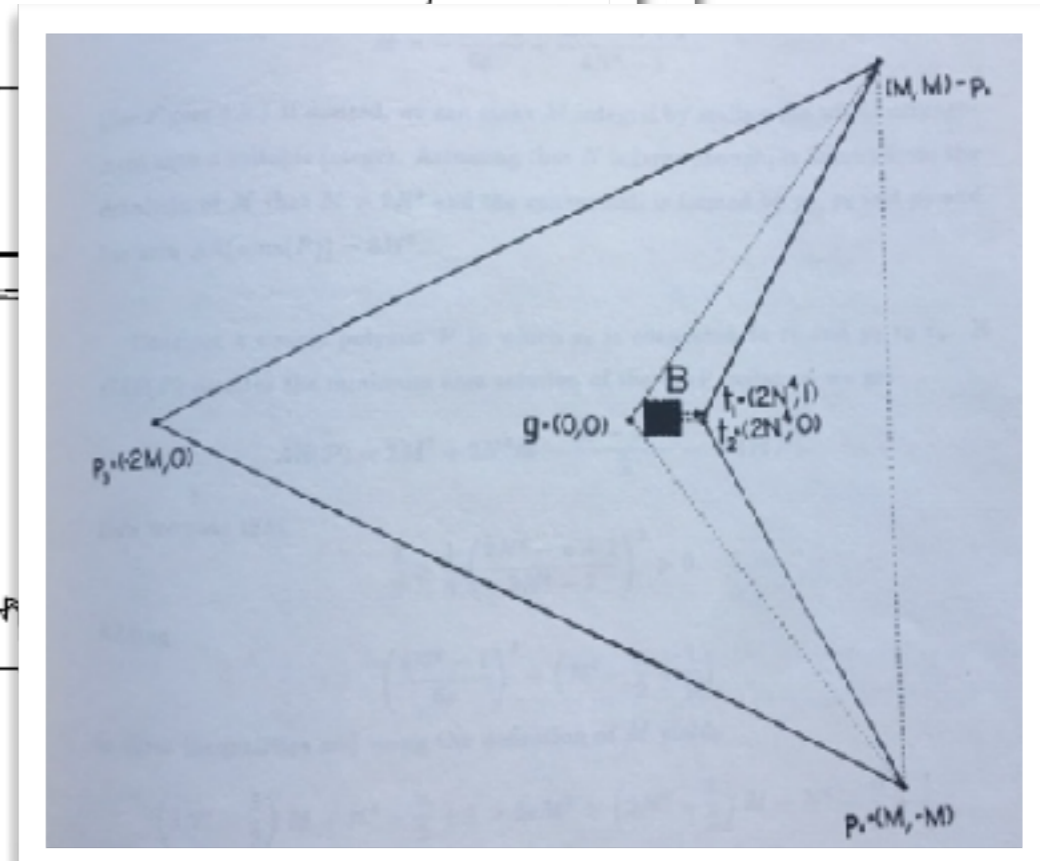
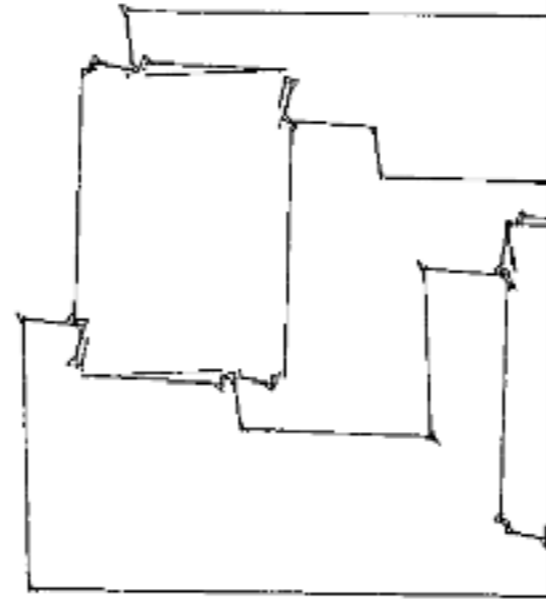
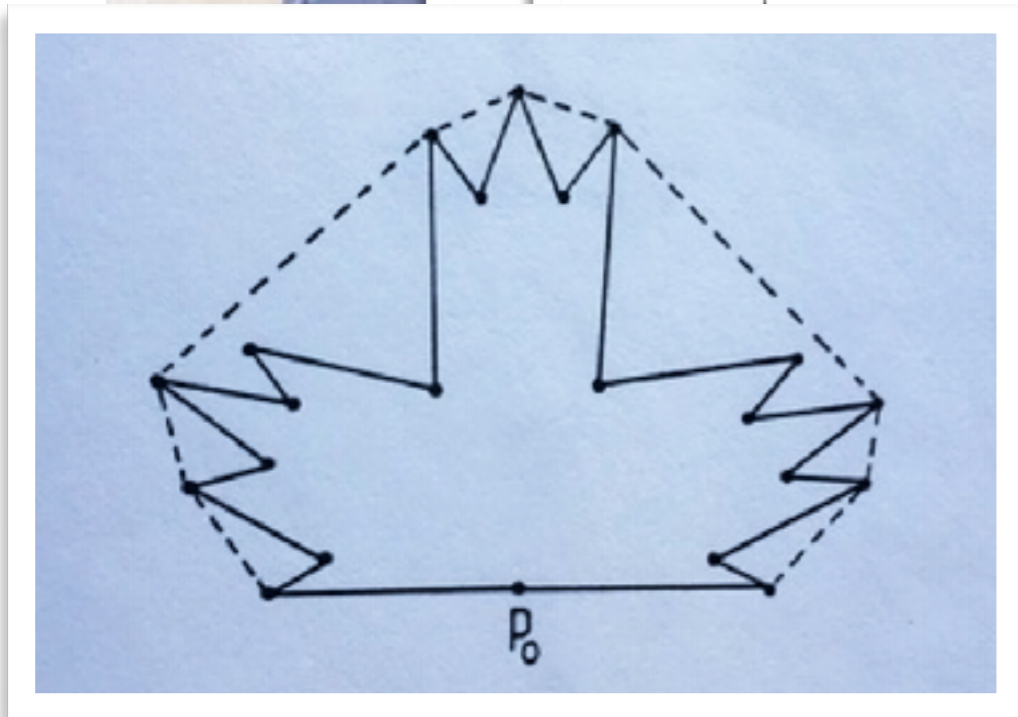
# Some History



**Fig. 24.** GRID-FULL solves GRID-EMPTY: turning a small polygon inside out.

a given set of  
ely related to  
n with vertex  
a maximum  
ness for the  
of a question  
rove that, for  
olume of the  
with a given  
80.

# Some History



$P_4$   $P_2$

$P_3$   $P_1$

with a given  
80.

**Fig. 24.** GRID-FULL solves GRID-EMPTY: turning a small polygon inside out.

# Some Recent Progress

# Some Recent Progress



Technische  
Universität  
Braunschweig

Bachelor's Thesis

## New algorithmic approaches for area-optimal polygons

Michael Perk

June 29, 2018

Institute of Operating Systems and Computer Networks  
Prof. Dr. Sándor Fekete

Supervisors:  
Arne Schmidt, M. Sc.  
Andreas Haas, M. Sc.  
Phillip Keldenich, M. Sc.

# Some Recent Progress



Technische  
Universität  
Braunschweig

Bachelor's Thesis

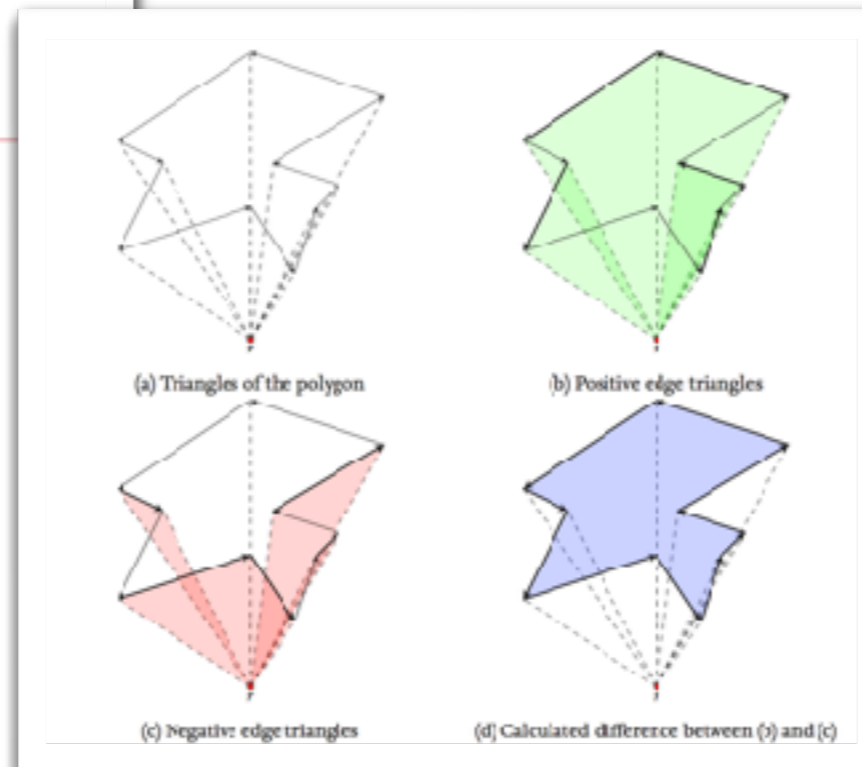
## New algorithmic approaches for area-optimal polygons

Michael Perk

June 29, 2018

Institute of Operating Systems and Computer Networks  
Prof. Dr. Sándor Fekete

Supervisors:  
Arne Schmidt, M. Sc.  
Andreas Haas, M. Sc.  
Phillip Keldenich, M. Sc.



Technische  
Universität  
Braunschweig

# Some Recent Progress



Technische  
Universität  
Braunschweig

Bachelor's Thesis

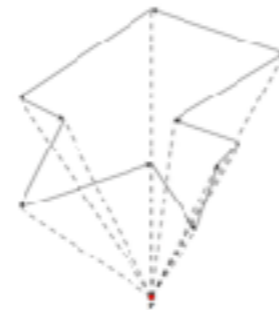
## New algorithmic approaches for area-optimal polygons

Michael Perk

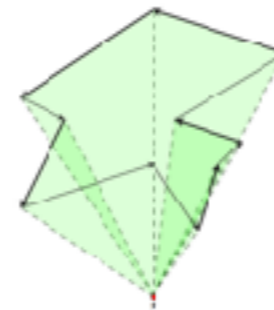
June 29, 2018

Institute of Operating Systems and Computer Networks  
Prof. Dr. Sándor Fekete

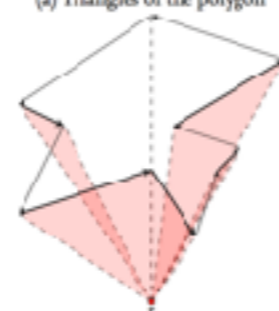
Supervisors:  
Arne Schmidt, M. Sc.  
Andreas Haas, M. Sc.  
Phillip Keldenich, M. Sc.



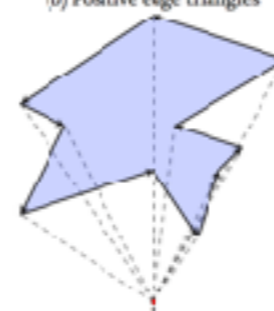
(a) Triangles of the polygon



(b) Positive edge triangles



(c) Negative edge triangles



(d) Calculated difference between (b) and (c)

$$\{\min, \max\} \sum_{e^+ \in E^+} z_{e^+} \cdot f_e - \sum_{e^- \in E^-} z_{e^-} \cdot f_e$$

$$\forall s_i \in S: \sum_{(j,i) \in \delta^+(s_i)} z_{ij} = 1$$

$$\forall s_i \in S: \sum_{(i,j) \in \delta^-(s_i)} z_{ij} = 1$$

$$\forall e = \{i, j\} \in E: z_{ij} + z_{ji} \leq 1$$

$$\forall \text{intersecting } \{i, j\}, \{k, l\} \in E: z_{ij} + z_{ji} + z_{kl} + z_{lk} \leq 1$$

$$(\forall \text{slabs } D)(\forall m = 1, \dots, |D|): \sum_{i=1}^m z_{e_i^+} - z_{e_i^-}$$

$$\forall \emptyset \neq D \subseteq S: \sum_{(k,l) \in \delta^-(D)} z_{kl} \geq 1$$

$$\sum_{(k,l) \in \delta^+(D)} z_{kl} \geq 1$$

$$\forall e \in E^+ \cup E^-: e \in \{0, 1\}$$

# Some Recent Progress



Technische  
Universität  
Braunschweig

Bachelor's Thesis

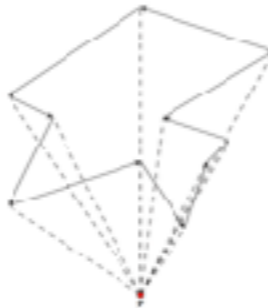
## New algorithmic approaches for area-optimal polygons

Michael Perk

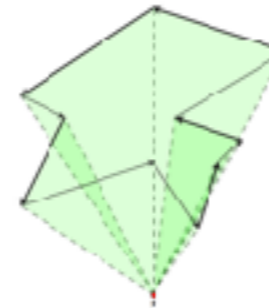
June 29, 2018

Institute of Operating Systems and Computer Networks  
Prof. Dr. Sándor Fekete

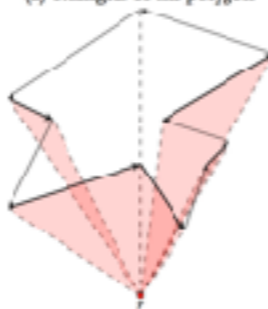
Supervisors:  
Arne Schmidt, M. Sc.  
Andreas Haas, M. Sc.  
Phillip Keldenich, M. Sc.



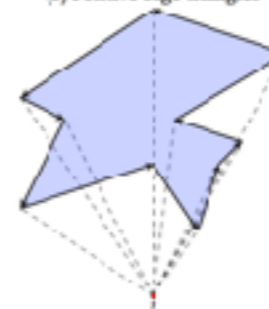
(a) Triangles of the polygon



(b) Positive edge triangles



(c) Negative edge triangles



(d) Calculated difference between (b) and (c)

$$\{\min, \max\} \sum_{e^+ \in E^+} z_{e^+} \cdot f_e - \sum_{e^- \in E^-} z_{e^-} \cdot f_e$$

$$\forall s_i \in S: \sum_{(j,i) \in S^+(s_i)} z_{ij} = 1$$

$$\forall s_i \in S: \sum_{(i,j) \in S^-(s_i)} z_{ij} = 1$$

$$\forall e = \{i, j\} \in E: z_{ij} + z_{ji} \leq 1$$

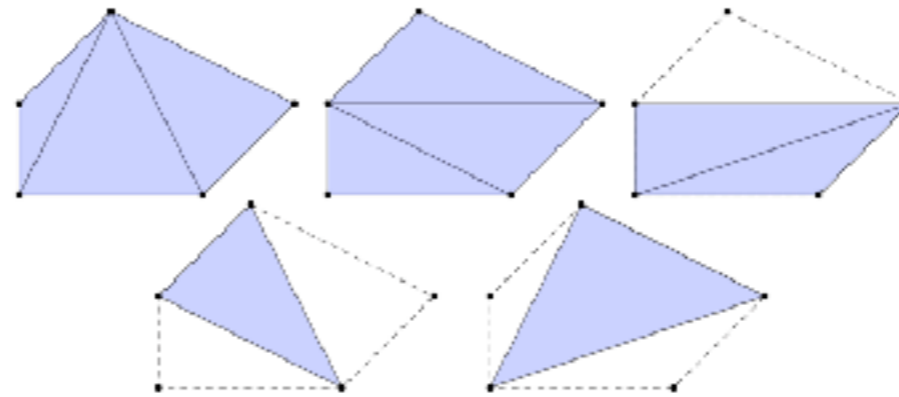
$$\text{Nonintersecting } \{(i, j), \{k, l\} \in E: z_{ij} + z_{ji} + z_{kl} + z_{lk} \leq 1$$

$$(\forall \text{slabs } D)(\forall m = 1, \dots, |D|): \sum_{i=1}^m z_{e_i^+} - z_{e_i^-}$$

$$\forall \emptyset \neq D \subseteq S: \sum_{(k,l) \in D^-} z_{kl} \geq 1$$

$$\sum_{(k,l) \in D^+} z_{kl} \geq 1$$

$$\forall e \in E^+ \cup E^-: e \in \{0, 1\}$$



Technische  
Universität  
Braunschweig

# Some Recent Progress



Technische  
Universität  
Braunschweig

Bachelor's Thesis

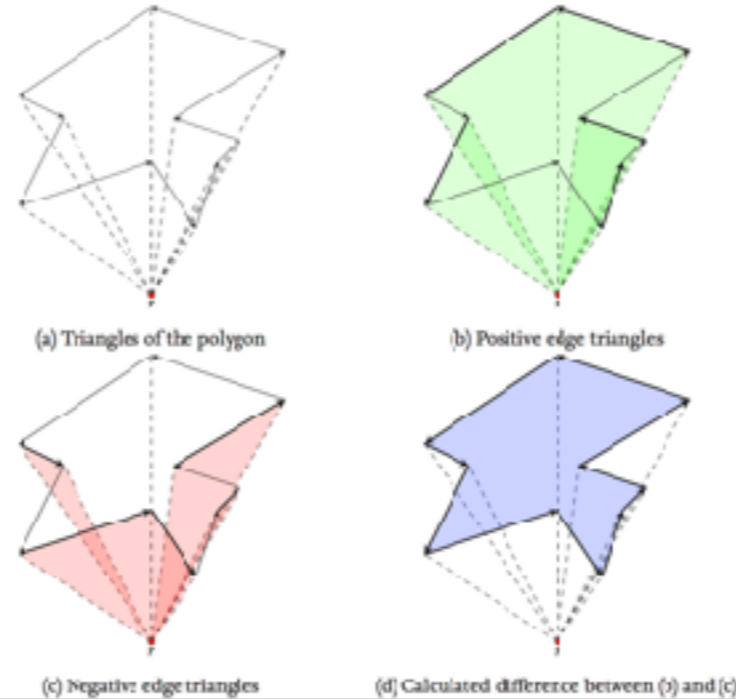
## New algorithmic approaches for area-optimal polygons

Michael Perk

June 29, 2018

Institute of Operating Systems and Computer Networks  
Prof. Dr. Sándor Fekete

Supervisors:  
Arne Schmidt, M. Sc.  
Andreas Haas, M. Sc.  
Phillip Keldenich, M. Sc.



$$\{\min, \max\} \sum_{e^+ \in E^+} z_{e^+} \cdot f_e - \sum_{e^- \in E^-} z_{e^-} \cdot f_e$$

$$\forall s_i \in S: \sum_{(j) \in S^+(s_i)} z_{ji} = 1$$

$$\forall s_i \in S: \sum_{(l_j) \in S^-(s_i)} z_{ij} = 1$$

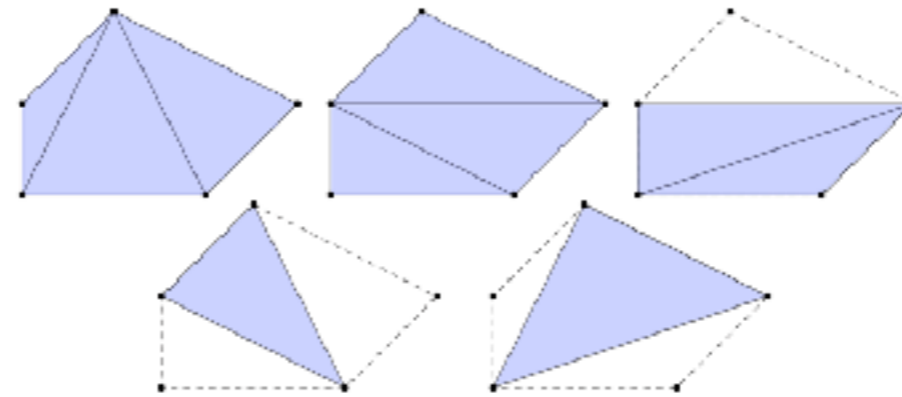
$$\forall e = \{i, j\} \in E: z_{ij} + z_{ji} \leq 1$$

$$\text{Vintersecting } \{i, j\}, \{k, l\} \in E: z_{ij} + z_{ji} + z_{kl} + z_{lk} \leq 1$$

$$(\forall \text{slabs } D)(\forall m = 1, \dots, |D|): \sum_{l=1}^m z_{l_0}^{e^+} - z_{l_0}^{e^-}$$

$$\forall \emptyset \neq D \subseteq S: \sum_{(k,l) \in D^-(D)} z_{kl} \geq 1$$

$$\forall e \in E^+ \cup E^-: e \in \{0, 1\}$$



$$\{\min, \max\} \sum_{\Delta \in T} f_{\Delta} \cdot x_{\Delta}$$

$$\sum_{\Delta \in T} x_{\Delta} = n - 2$$

$$\forall s_i \in S: \sum_{\Delta \in S(s_i)} x_{\Delta} \geq 1$$

$$\text{Vintersecting } \Delta_i, \Delta_j \in T: x_{\Delta_i} + x_{\Delta_j} \leq 1$$

$$\forall \emptyset \neq D \subseteq T, |D| \leq n - 3: \sum_{\Delta \in D} x_{\Delta} \leq \sum_{\Delta \in D} x_{\Delta} + |D| - 1$$

$$\forall \Delta \in T: x_{\Delta} \in \{0, 1\}$$



# Some Recent Progress



Technische  
Universität  
Braunschweig

Bachelor's Thesis

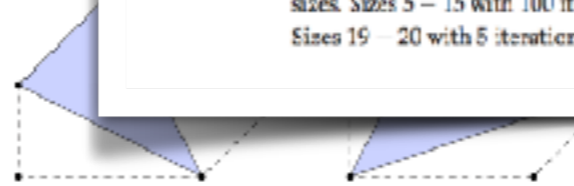
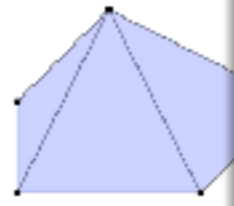
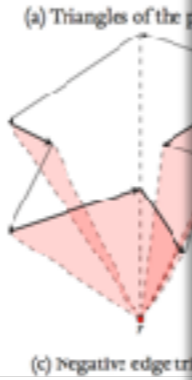
## New algorithmic approaches for area-optimal polygons

Michael Perk

June 29, 2018

Institute of Operating Systems and Computer Networks  
Prof. Dr. Sándor Fekete

Supervisors:  
Arne Schmidt, M. Sc.  
Andreas Haas, M. Sc.  
Phillip Keldenich, M. Sc.



$$\{\min, \max\} \sum z_{e^+} \cdot f_e - \sum z_{e^-} \cdot f_e$$

$$z_{ji} = 1$$

$$z_{ij} = 1$$

$$z_k + z_{lk} \leq 1$$

$$z_{ij} \in \{0, 1\}$$

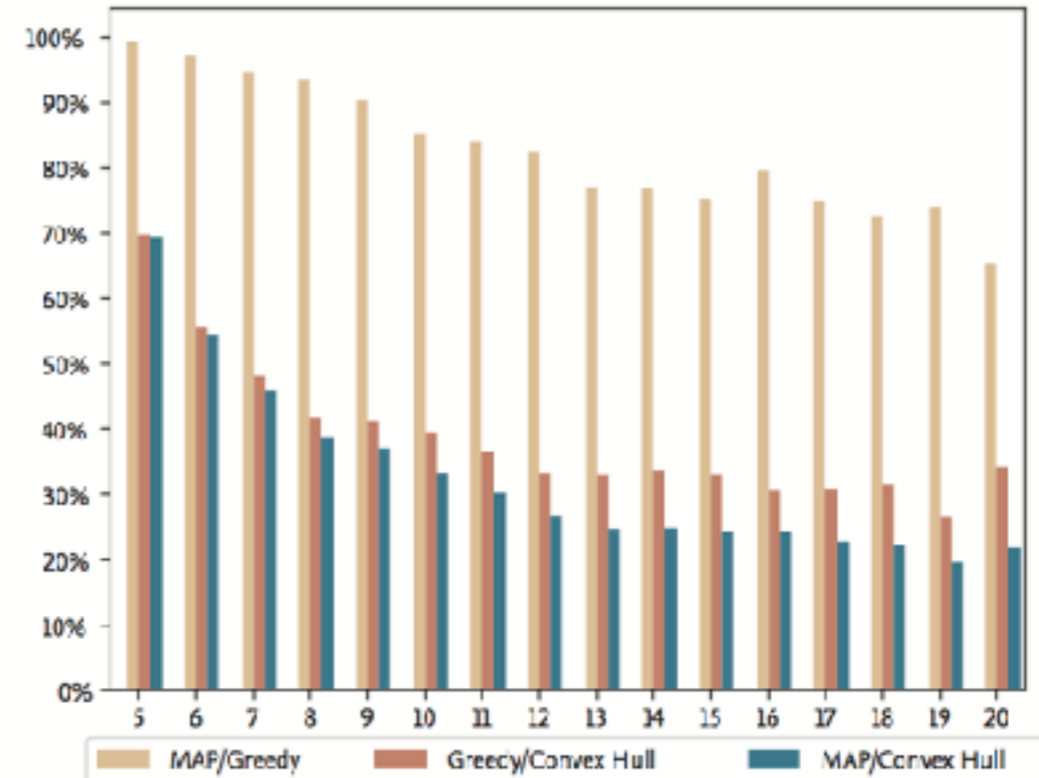


Figure 6.12: Bar chart which compares the calculated sizes of MAF, GREEDY MAP heuristic and the size of the convex hull. The values are the average obtained from instances of different sizes. Sizes 5 – 15 with 100 iterations per size. Sizes 16 – 18 with 20 iterations per size. Sizes 19 – 20 with 5 iterations per size.

$$\sum_{e \in E} x_e + |D| - 1$$

$$\forall \Delta \in T: x_\Delta \in \{0, 1\}$$



Technische  
Universität  
Braunschweig



Bachelor's Thesis

# New algorithmic approaches to area-optimal polygonization

Michael Perk

June 29, 2018

Institute of Operating Systems and  
Prof. Dr. Sándor Fekete

Supervisors:  
Arne Schmidt, M.  
Andreas Haas, M.  
Phillip Keldenich, M.



## Computing Area-Optimal Simple Polygonizations

SÁNDOR P. FEKETE, Department of Computer Science, TU Braunschweig, Germany

ANDREAS HAAS, Department of Computer Science, TU Braunschweig, Germany

PHILLIP KELDENICH, Department of Computer Science, TU Braunschweig, Germany

MICHAEL PERK, Department of Computer Science, TU Braunschweig, Germany

ARNE SCHMIDT, Department of Computer Science, TU Braunschweig, Germany

We consider methods for finding a simple polygon of minimum (MIN-AREA) or maximum (MAX-AREA) possible area for a given set of points in the plane. Both problems are known to be NP-hard; at the center of the recent CG Challenge, practical methods have received considerable attention. However, previous methods focused on heuristic methods, with no proof of optimality. We develop exact methods, based on a combination of geometry and integer programming. As a result, we are able to solve instances of up to  $n = 25$  points to provable optimality. While this extends the range of solvable instances by a considerable amount, it also illustrates the practical difficulty of both problem variants.

CCS Concepts: • Theory of computation → Design and analysis of algorithms; Computational Geometry.

Additional Key Words and Phrases: Computational Geometry, geometric optimization, algorithm engineering, exact algorithms, polygonalization, area optimization.

### ACM Reference Format:

Sándor P. Fekete, Andreas Haas, Phillip Keldenich, Michael Perk, and Arne Schmidt. 2020. Computing Area-Optimal Simple Polygonizations. *ACM Trans. Algor.* 99, 99, Article 99 (2020), 23 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

While the classic geometric Traveling Salesman Problem (TSP) is to find a (simple) polygon with a given set of vertices that has shortest perimeter, it is natural to look for a simple polygon with a given set of vertices that minimizes another basic geometric measure: the enclosed area. The problem MIN-AREA asks for a simple polygon with minimum enclosed area, while MAX-AREA demands one of maximum area; see Figure 1 for an illustration.

Both problem variants were shown to be NP-complete by Fekete [2, 3, 6], who also showed that no polynomial-time approximation scheme (PTAS) exists for MIN-AREA problem and gave a  $\frac{1}{2}$ -approximation algorithm for MAX-AREA.

Authors' addresses: Sándor P. Fekete, s.fekete@tu-bs.de, Department of Computer Science, TU Braunschweig, Braunschweig, Germany; Andreas Haas, a.haas@tu-bs.de, Department of Computer Science, TU Braunschweig, Braunschweig, Germany; Phillip Keldenich, p.keldenich@tu-bs.de, Department of Computer Science, TU Braunschweig, Braunschweig, Germany; Michael Perk, m.perk@tu-bs.de, Department of Computer Science, TU Braunschweig, Braunschweig, Germany; Arne Schmidt, arne.schmidt@tu-bs.de, Department of Computer Science, TU Braunschweig, Braunschweig, Germany.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

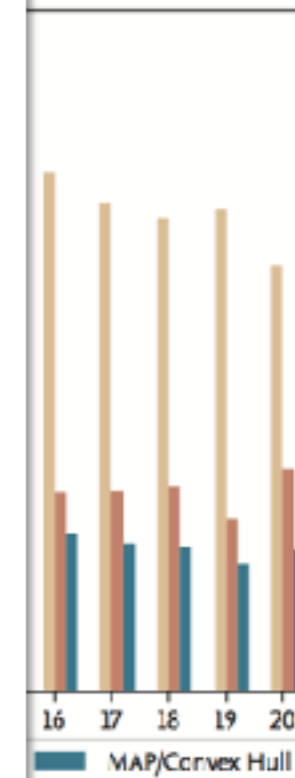
© 2020 Association for Computing Machinery.

Manuscript submitted to ACM

Manuscript submitted to ACM

1

$$\{y, \max\} \sum z_{e^+} \cdot f_e - \sum z_{e^-} \cdot f_e$$



MAP heuristic and the instances of different iterations per size.

$$z_{ji} = 1$$

$$z_{ij} = 1$$

$$\leq 1$$

$$z_{kl} + z_{lk} \leq 1$$

$$z_{e^+}$$

$$\{D\} z_{kl} \geq 1$$

$$\{D\} z_{kl} \geq 1$$

$$-2$$

$$1$$

$$1$$

$$\sum_{e \in E(D)} x_e + |D| - 1$$

$$\forall \Delta \in T: x_\Delta \in \{0, 1\}$$

# Some Recent Progress



Technische  
Universität  
Braunschweig

Bachelor's Thesis

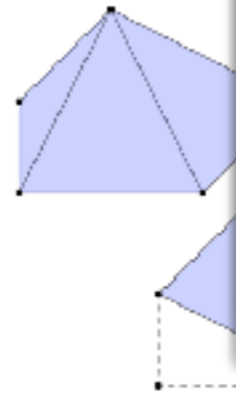
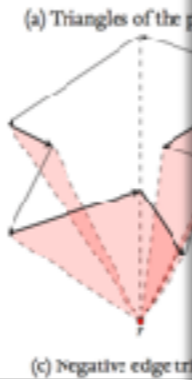
## New algorithmic approaches for area-optimal polygons

Michael Perk

June 29, 2018

Institute of Operating Systems and Computer Networks  
Prof. Dr. Sándor Fekete

Supervisors:  
Arne Schmidt, M. Sc.  
Andreas Haas, M. Sc.  
Phillip Keldenich, M. Sc.



$$\{\min, \max\} \sum z_{e^+} \cdot f_e - \sum z_{e^-} \cdot f_e$$

$$z_{ji} = 1$$

$$z_{ij} = 1$$

$$\leq 1$$

$$z_{kl} + z_{lk} \leq 1$$

$$z_{e_i}^{e_j}$$

$$\sum_{(D)} z_{kl} \geq 1$$

$$\sum_{(D)} z_{kl} \geq 1$$

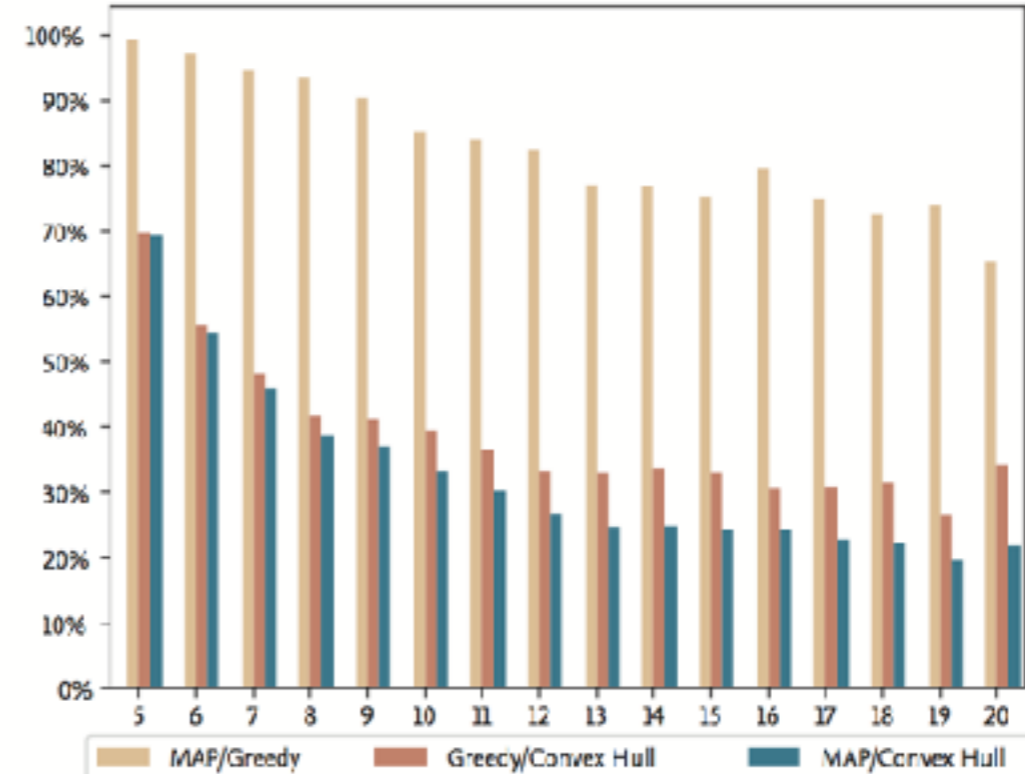


Figure 6.12.: Bar chart which compares the calculated sizes of MAF, GREEDY MAP heuristic and the size of the convex hull. The values are the average obtained from instances of different sizes. Sizes 5 – 15 with 100 iterations per size. Sizes 16 – 18 with 20 iterations per size. Sizes 19 – 20 with 5 iterations per size.

$$-2$$

$$1$$

$$1$$

$$\sum_{e \in D} x_e + |D| - 1$$

$$\forall \Delta \in T: x_\Delta \in \{0, 1\}$$

# The 2019 CG Challenge

# Area-Optimal Simple Polygonalizations: The CG Challenge 2019

ERIK D. DEMAINE, CSAIL, MIT, USA

SÁNDOR P. FEKETE, Department of Computer Science, TU Braunschweig, Germany

PHILLIP KELDENICH, Department of Computer Science, TU Braunschweig, Germany

DOMINIK KRUPKE, Department of Computer Science, TU Braunschweig, Germany

JOSEPH S. B. MITCHELL, Department of Applied Mathematics and Statistics, Stony Brook University, USA

We give an overview of theoretical and practical aspects of finding a simple polygon of minimum (MIN-AREA) or maximum (MAX-AREA) possible area for a given set of  $n$  points in the plane. Both problems are known to be  $\mathcal{NP}$ -hard and were the subject of the 2019 Computational Geometry Challenge, which presented the quest of finding good solutions to more than 200 instances, ranging from  $n = 10$  all the way to  $n = 1,000,000$ .

CCS Concepts: • Theory of computation → Design and analysis of algorithms; Computational Geometry.

## ACM Reference Format:

Erik D. Demaine, Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Joseph S. B. Mitchell. 2020. Area-Optimal Simple Polygonalizations: The CG Challenge 2019. *ACM J. Exp. Algor.* 99, 99, Article 99 (2020), 12 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

## 1 INTRODUCTION

### 1.1 The Computational Geometry Challenge

The "CG-SHOP Challenge" (Computational Geometry: Solving Hard Optimization Problems) originated as a workshop at the 2019 Computational Geometry Week (CG Week) in Portland, Oregon in June, 2019. The goal was to conduct a computational challenge competition that focused attention on a specific hard geometric optimization problem, encouraging researchers to devise and implement solution methods that could be compared scientifically based on how well they performed on a database of instances. While much of computational geometry research has targeted theoretical research, often seeking provable approximation algorithms for  $\mathcal{NP}$ -hard optimization problems, the goal of the CG Challenge was to set the metric of success based on computational results on a specific set of benchmark geometric instances. The 2019 CG Challenge focused on the problem of computing simple polygons whose vertices were a given set of points in the plane.

Authors' addresses: Erik D. Demaine, [edemaine@mit.edu](mailto:edemaine@mit.edu), CSAIL, MIT, Cambridge, MA, USA; Sándor P. Fekete, [s.fekete@tu-bs.de](mailto:s.fekete@tu-bs.de), Department of Computer Science, TU Braunschweig, Braunschweig, Germany; Phillip Keldenich, [p.keldenich@tu-bs.de](mailto:p.keldenich@tu-bs.de), Department of Computer Science, TU Braunschweig, Braunschweig, Germany; Dominik Krupke, [d.krupke@tu-bs.de](mailto:d.krupke@tu-bs.de), Department of Computer Science, TU Braunschweig, Braunschweig, Germany; Joseph S. B. Mitchell, [joseph.mitchell@stonybrook.edu](mailto:joseph.mitchell@stonybrook.edu), Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2020 Association for Computing Machinery.

Manuscript submitted to ACM

Manuscript submitted to ACM

# The 2019 CG Challenge

# The 2019 CG Challenge

The contest consists of a total of **247 instances**, as follows. For  $n =$   
{10,15,20,25,30,35,40,45,50,60,70,80,90,100,200,300,400,500,600,700,800,900,1000,  
2000,3000,4000,5000,6000,7000,8000,9000,10000,20000,30000,40000,50000,60000, 70000,80000,90000,100000}, there are six instances each. In  
addition, there will be one instance of size  $n=1000000$ .

These instances are of three different types:

- uniform: uniformly at random from a square
- edge: randomly generated according to the distribution of the rate of change (the "edges") of an image
- illumination: randomly generated according to the distribution of brightness of an image (such as an illumination map)

Each instance consists of  $n$  points in the plane with even integer coordinates. (This ensures that the area of any simple polygon will be an integer.)

# The 2019 CG Challenge

The contest consists of a total of **247 instances**, as follows. For  $n = \{10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, 90000, 100000\}$ , there are six instances each. In addition, there will be one instance of size  $n=1000000$ .

These instances are of three different types:

- uniform: uniformly at random from a square
- edge: randomly generated according to the distribution of the rate of change (the "edges") of an image
- illumination: randomly generated according to the distribution of brightness of an image (such as an illumination map)

Each instance consists of  $n$  points in the plane with even integer coordinates. (This ensures that the area of any simple polygon will be an integer.)

For each instance, the **score** is the ratio between the achieved area, divided by the area of the convex hull, i.e., a number between 0 and 1. For instances without a feasible solution, the default score is 1 (for minimization) or 0 (for maximization). The total score is the sum of all 247 individual scores.



# The 2019 CG Challenge

The contest consists of a total of **247 instances**, as follows. For  $n = \{10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, 90000, 100000\}$ , there are six instances each. In addition, there will be one instance of size  $n=1000000$ .

These instances are of three different types:

- **uniform**: uniformly at random from a square
- **edge**: randomly generated according to the distribution of the rate of change (the "edges") of an image
- **illumination**: randomly generated according to the distribution of brightness of an image (such as an illumination map)

Each instance consists of  $n$  points in the plane with even integer coordinates. (This ensures that the area of any simple polygon will be an integer.)

For each instance, the **score** is the ratio between the achieved area, divided by the area of the convex hull, i.e., a number between 0 and 1. For instances without a feasible solution, the default score is 1 (for minimization) or 0 (for maximization). The total score is the sum of all 247 individual scores.

The contest will be run in several different categories. These categories include:

- (Score\_min) The best total score for minimum area polygons
- (Score\_max) The best total score for maximum area polygons
- (Opt\_min) The largest number of instances solved to optimality for a minimum area polygon
- (Opt\_max) The largest number of instances solved to optimality for a maximum area polygon
- (Bound\_min) The best bounds for minimization
- (Bound\_max) The best bounds for maximization

# The 2019 CG Challenge

The contest consists of a total of **247 instances**, as follows. For  $n = \{10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, 90000, 100000\}$ , there are six instances each. In addition, there will be one instance of size  $n=1000000$ .

These instances are of three different types:

- **uniform**: uniformly at random from a square
- **edge**: randomly generated according to the distribution of the rate of change (the "edges") of an image
- **illumination**: randomly generated according to the distribution of brightness of an image (such as an illumination map)

Each instance consists of  $n$  points in the plane with even integer coordinates. (This ensures that the area of any simple polygon will be an integer.)

For each instance, the **score** is the ratio between the achieved area, divided by the area of the convex hull, i.e., a number between 0 and 1. For instances without a feasible solution, the default score is 1 (for minimization) or 0 (for maximization). The total score is the sum of all 247 individual scores.

The contest will be run in several different categories. These categories include:

- (Score\_min) The best total score for minimum area polygons
- (Score\_max) The best total score for maximum area polygons
- (Opt\_min) The largest number of instances solved to optimality for a minimum area polygon
- (Opt\_max) The largest number of instances solved to optimality for a maximum area polygon
- (Bound\_min) The best bounds for minimization
- (Bound\_max) The best bounds for maximization

**Contest opens 24:00 (midnight, CET), February 28, 2019.**

**Contest closes 24:00 (midnight, CET), May 31, 2019.**

# Instances

# Instances



Hard instances project

# Instances











## Hard instances project

Hard instances project [Home](#) [Search](#)

Instance:

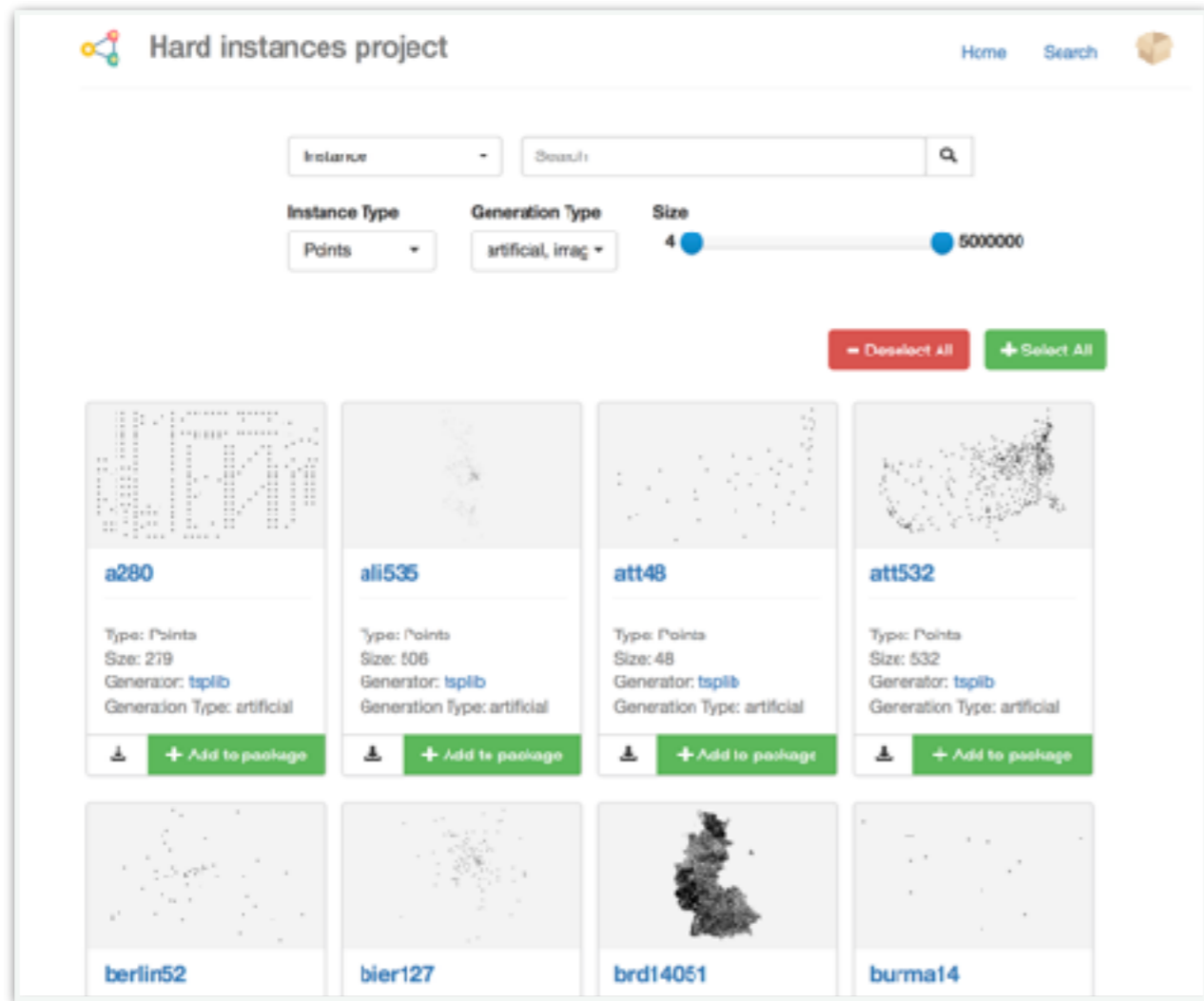
Instance Type:  Generation Type:  Size:

 <b>a280</b> Type: Points Size: 279 Generator: <a href="#">tsplib</a> Generation Type: artificial <input type="button" value="+ Add to package"/>	 <b>ali535</b> Type: Points Size: 506 Generator: <a href="#">tsplib</a> Generation Type: artificial <input type="button" value="+ Add to package"/>	 <b>att48</b> Type: Points Size: 48 Generator: <a href="#">tsplib</a> Generation Type: artificial <input type="button" value="+ Add to package"/>	 <b>att532</b> Type: Points Size: 532 Generator: <a href="#">tsplib</a> Generation Type: artificial <input type="button" value="+ Add to package"/>
 <b>berlin52</b>	 <b>bier127</b>	 <b>brd14051</b>	 <b>burma14</b>

# Instances

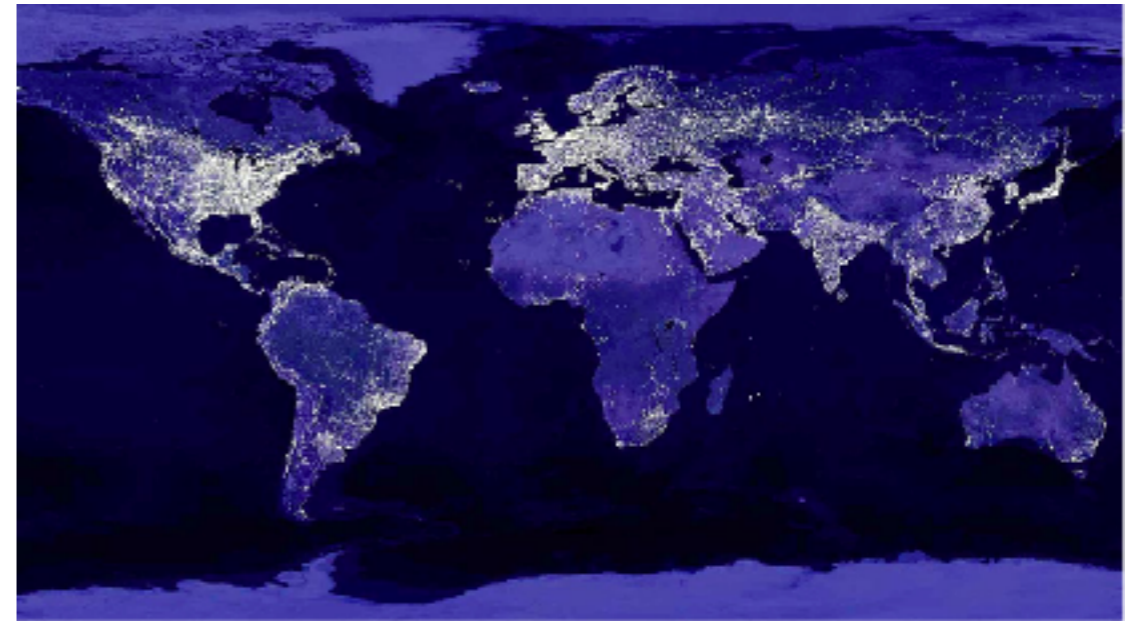


## Hard instances project



The screenshot shows the web interface for the 'Hard instances project'. At the top, there is a navigation bar with 'Home' and 'Search' links. Below this is a search bar with a dropdown menu for 'Instance' and a search input field. There are also filters for 'Instance Type' (set to 'Points'), 'Generation Type' (set to 'artificial, iraq'), and 'Size' (set to 4 on a scale up to 5,000,000). Below the filters are two buttons: 'Deselect All' and 'Select All'. The main content area displays a grid of instance cards. Each card shows a small map of the instance, its name, type, size, generator, and generation type, along with an 'Add to package' button.

Instance Name	Type	Size	Generator	Generation Type
a280	Points	279	tsplib	artificial
ali535	Points	506	tsplib	artificial
att48	Points	48	tsplib	artificial
att532	Points	532	tsplib	artificial
berlin52	Points			
bier127	Points			
brd14051	Points			
burma14	Points			



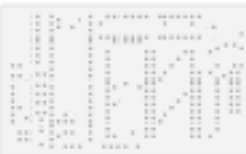
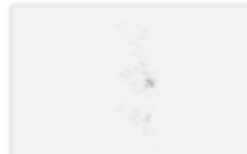






# Instances

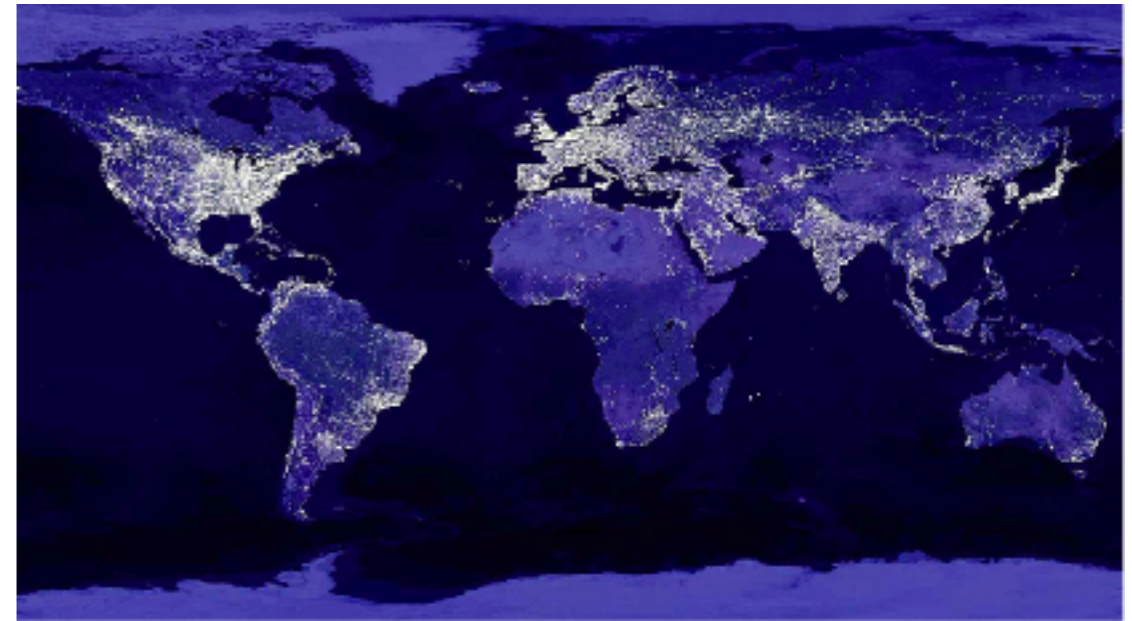
## Hard instances project

Hard instances project Home Search

Instance:

Instance Type:  Generation Type:  Size:

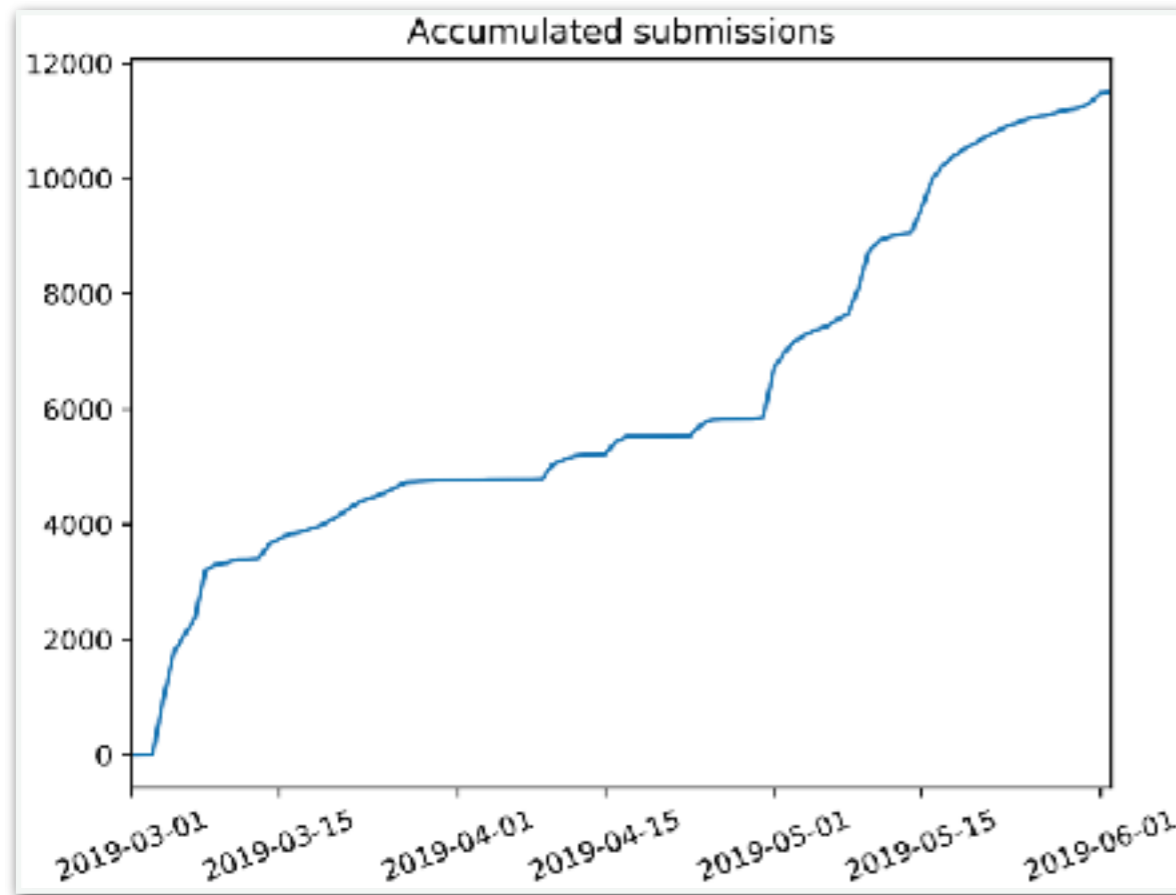
 <b>a280</b> Type: Points Size: 279 Generator: <a href="#">tsplib</a> Generation Type: artificial <input type="button" value="+ Add to package"/>	 <b>ali535</b> Type: Points Size: 106 Generator: <a href="#">tsplib</a> Generation Type: artificial <input type="button" value="+ Add to package"/>	 <b>att48</b> Type: Points Size: 48 Generator: <a href="#">tsplib</a> Generation Type: artificial <input type="button" value="+ Add to package"/>	 <b>att532</b> Type: Points Size: 532 Generator: <a href="#">tsplib</a> Generation Type: artificial <input type="button" value="+ Add to package"/>
 <b>berlin52</b>	 <b>bier127</b>	 <b>brd14051</b>	 <b>burma14</b>



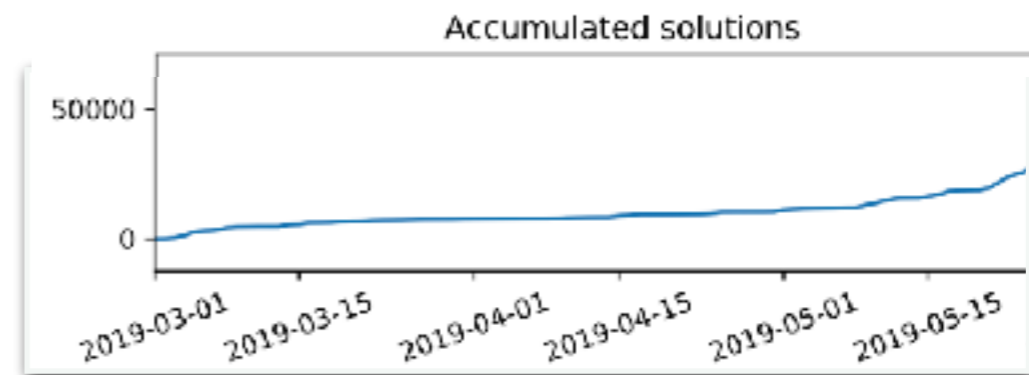
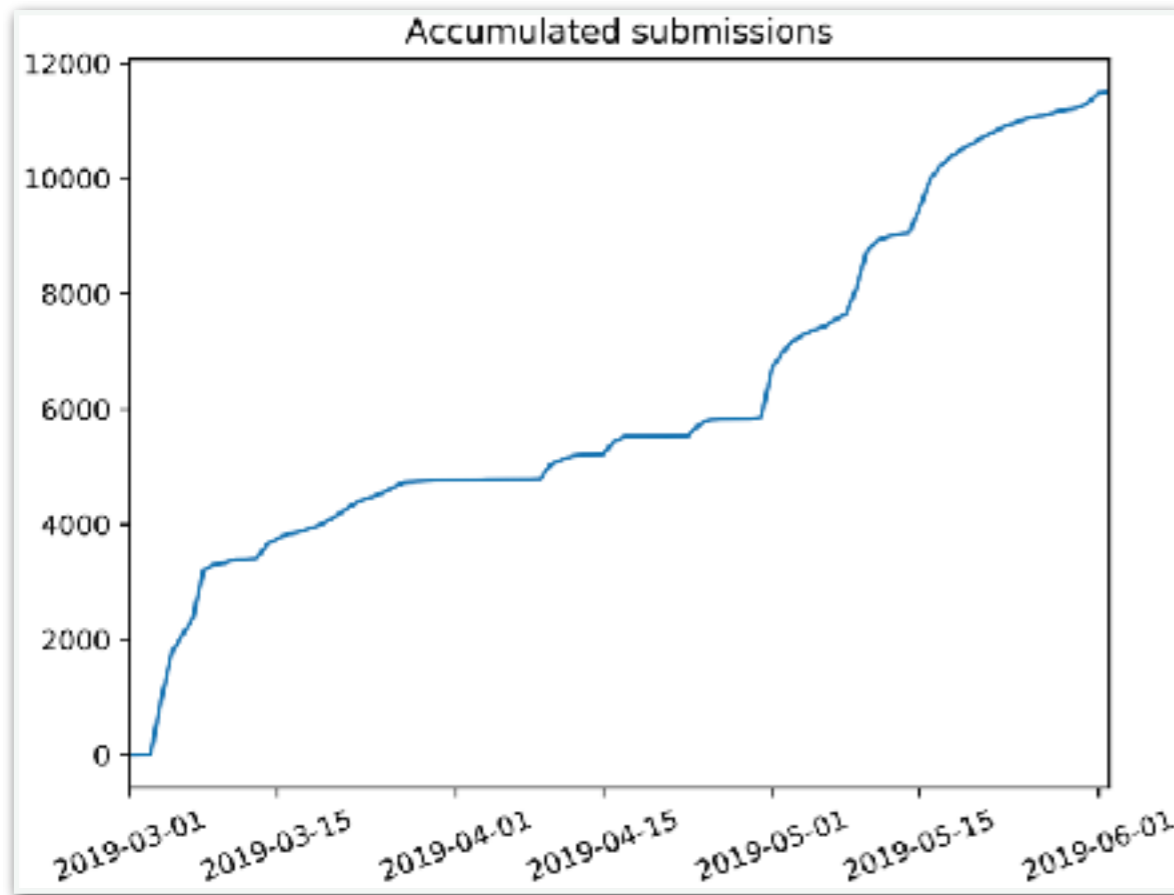
# Progress



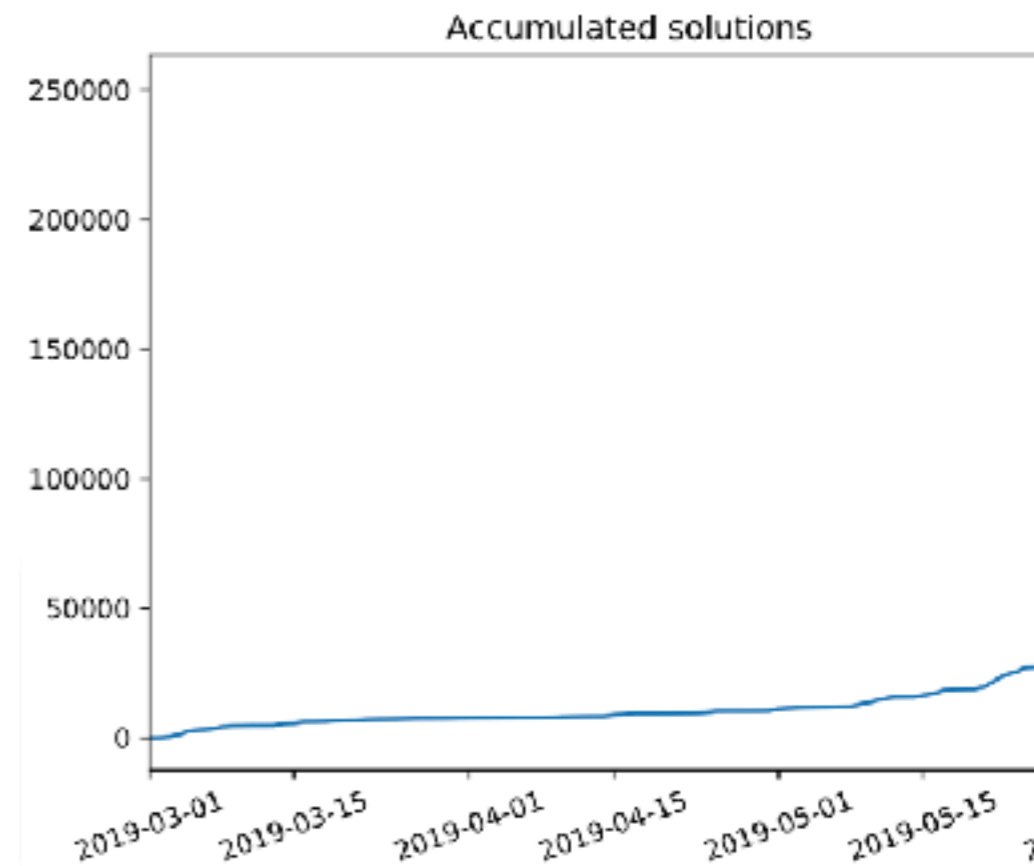
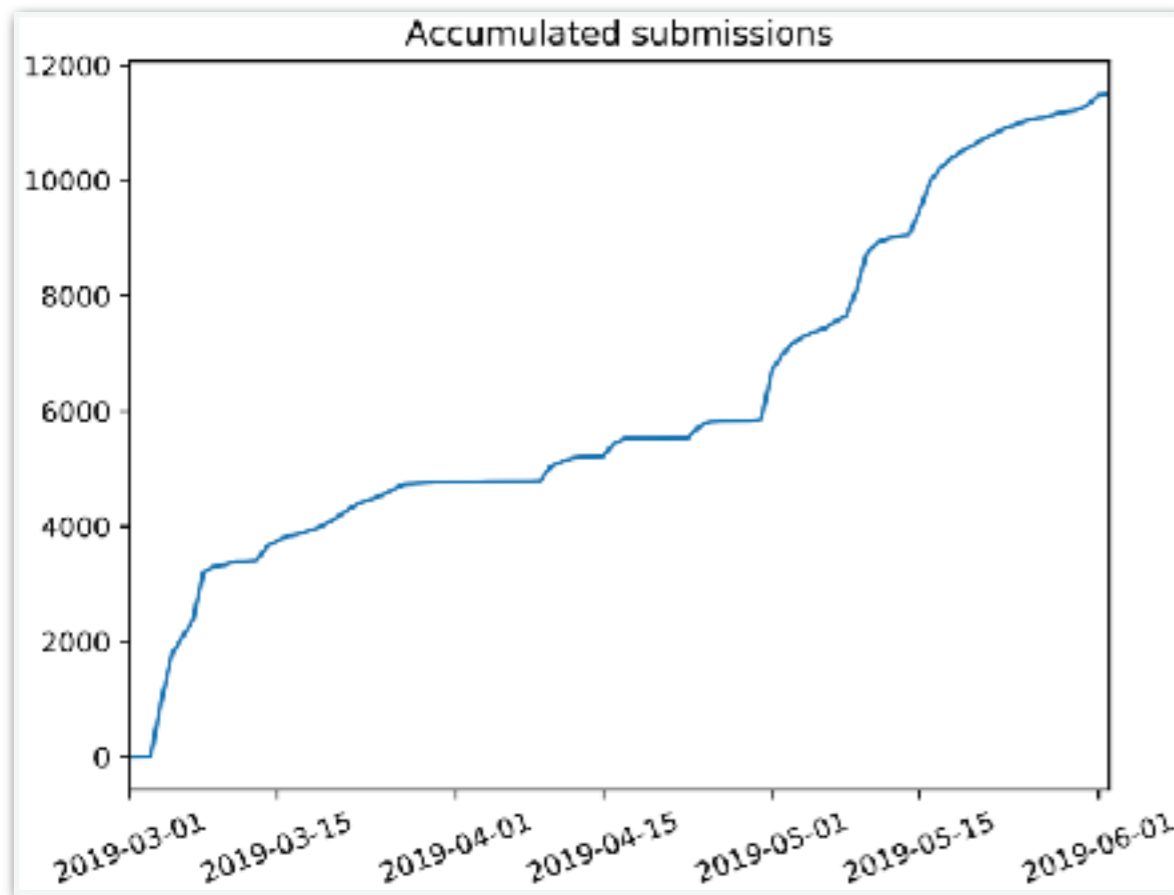
# Progress



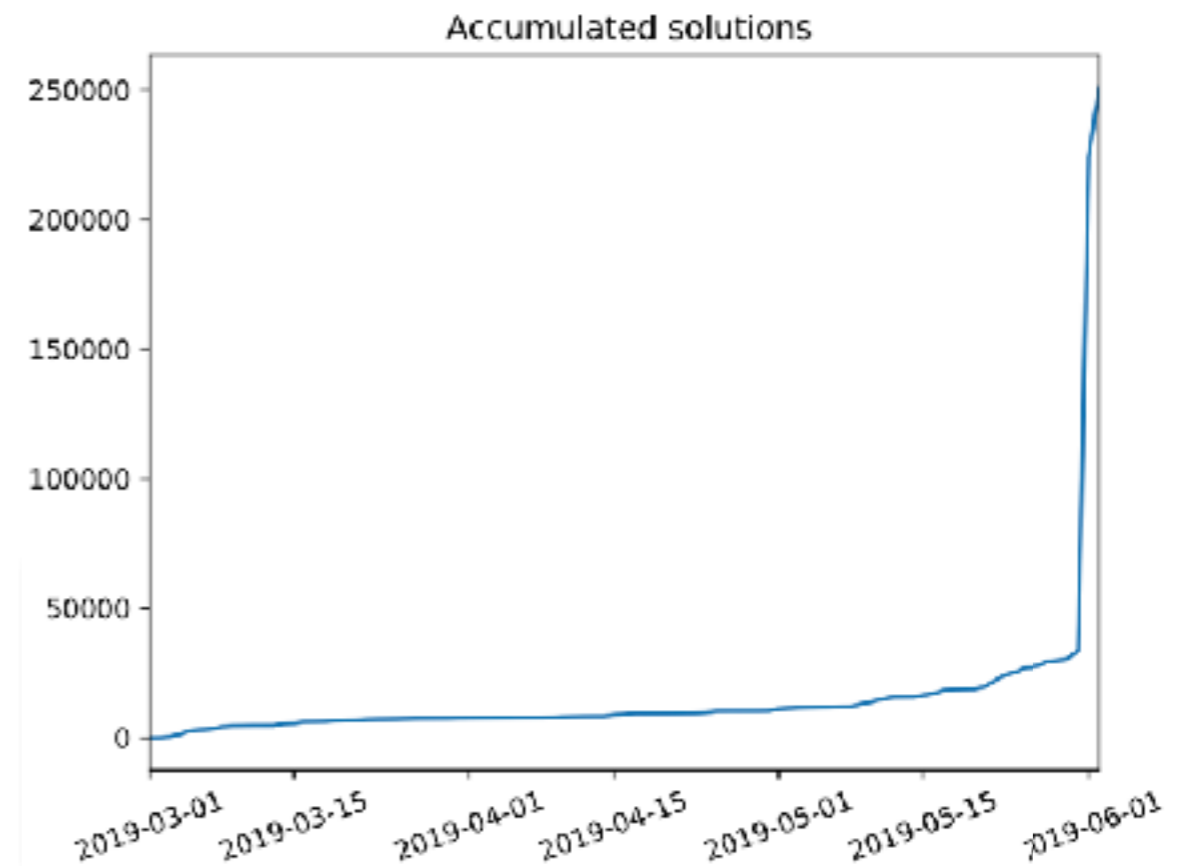
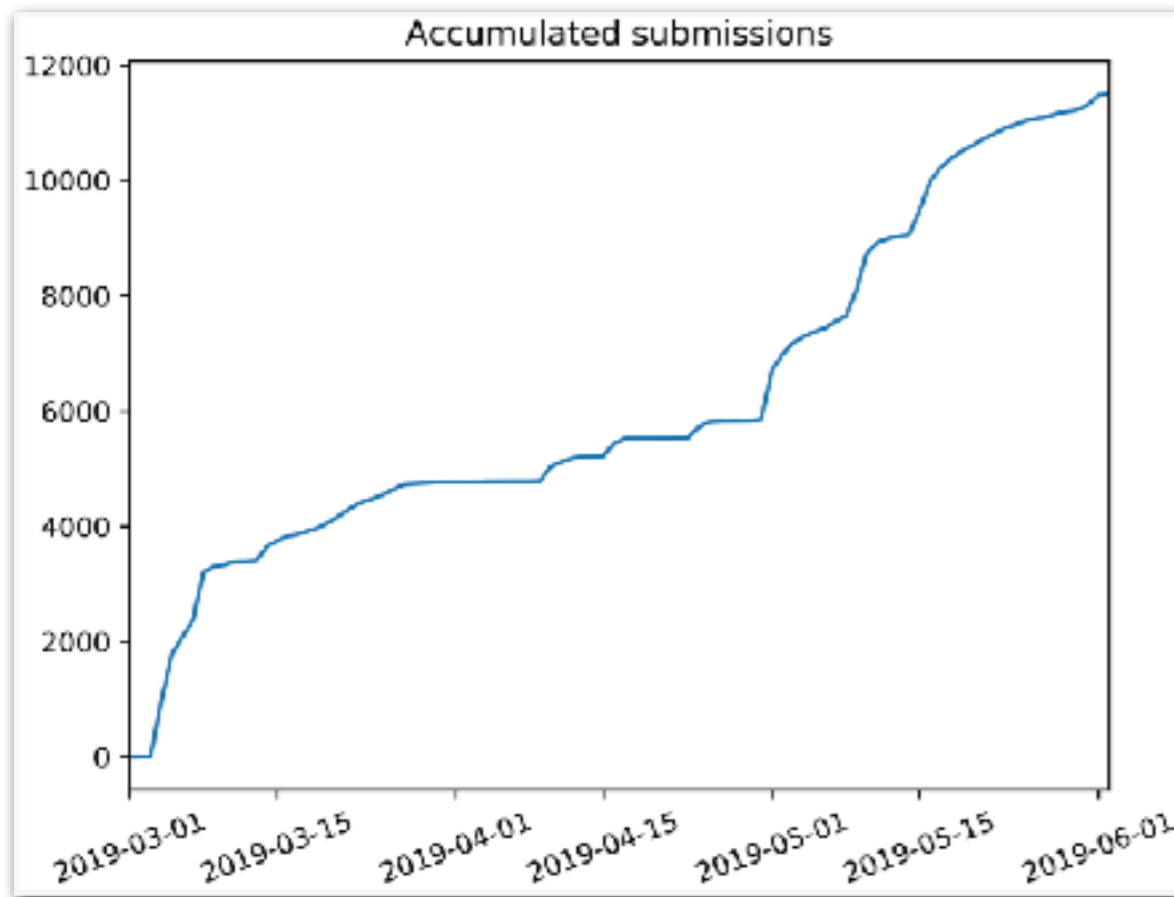
# Progress

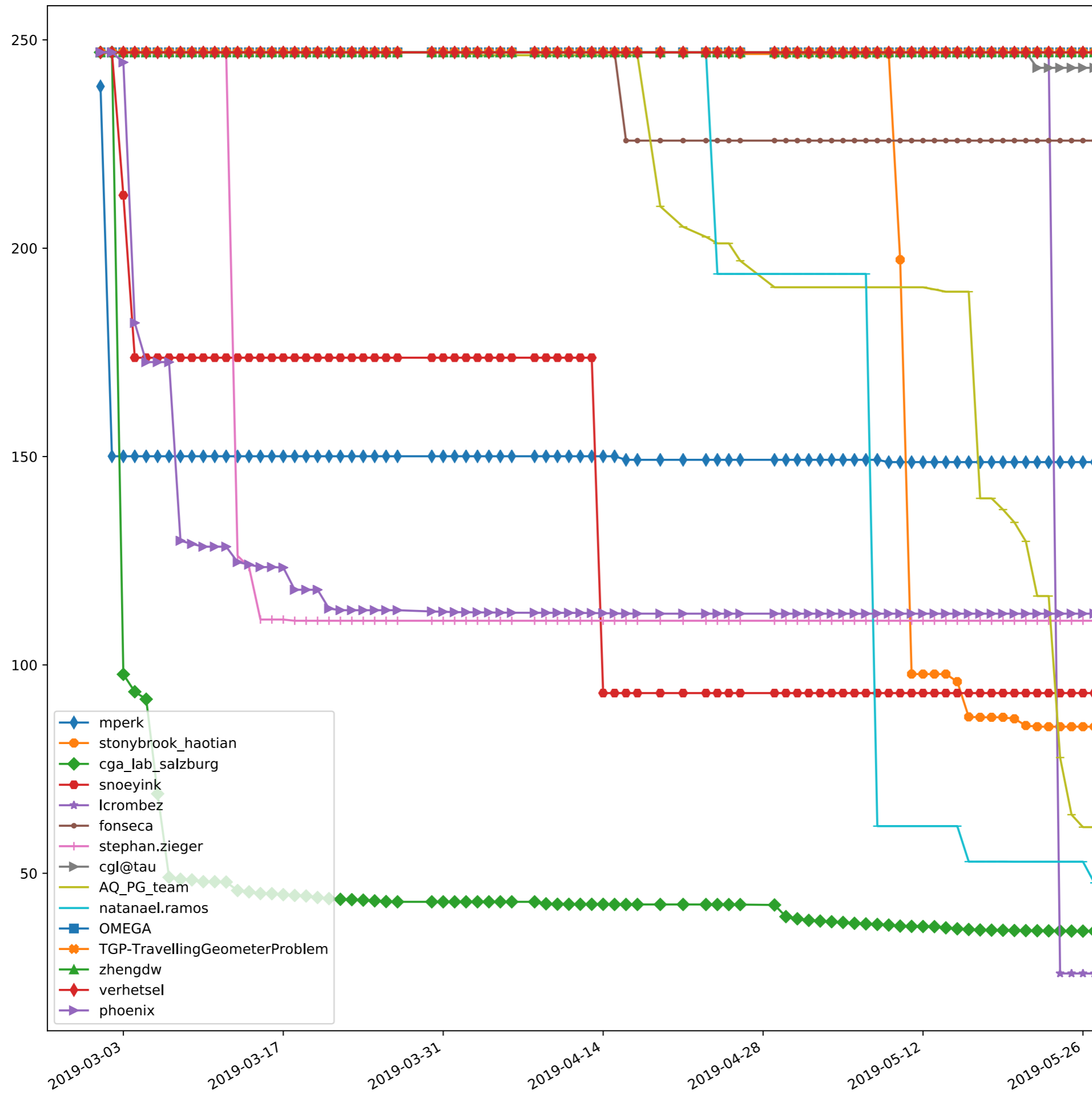


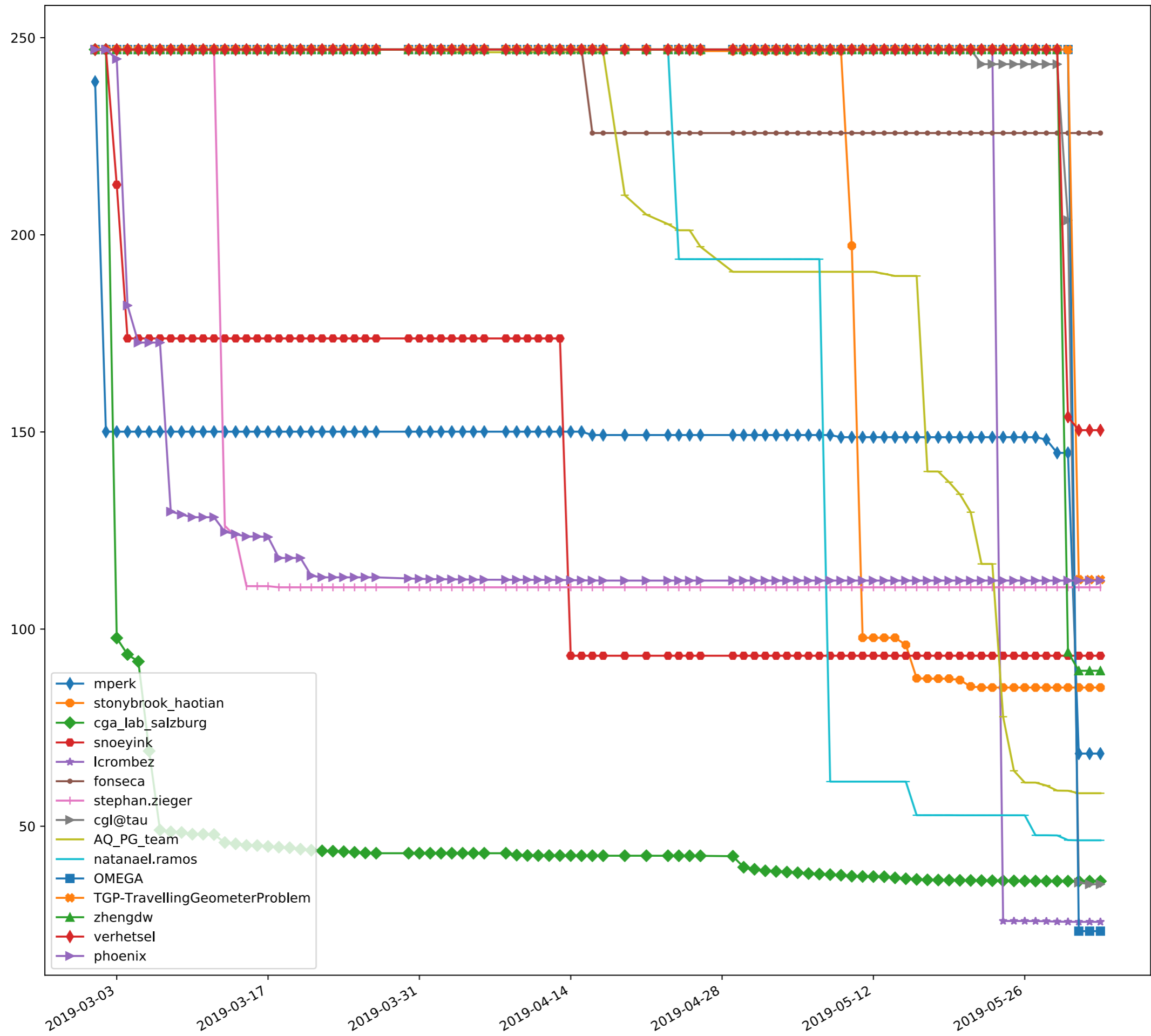
# Progress



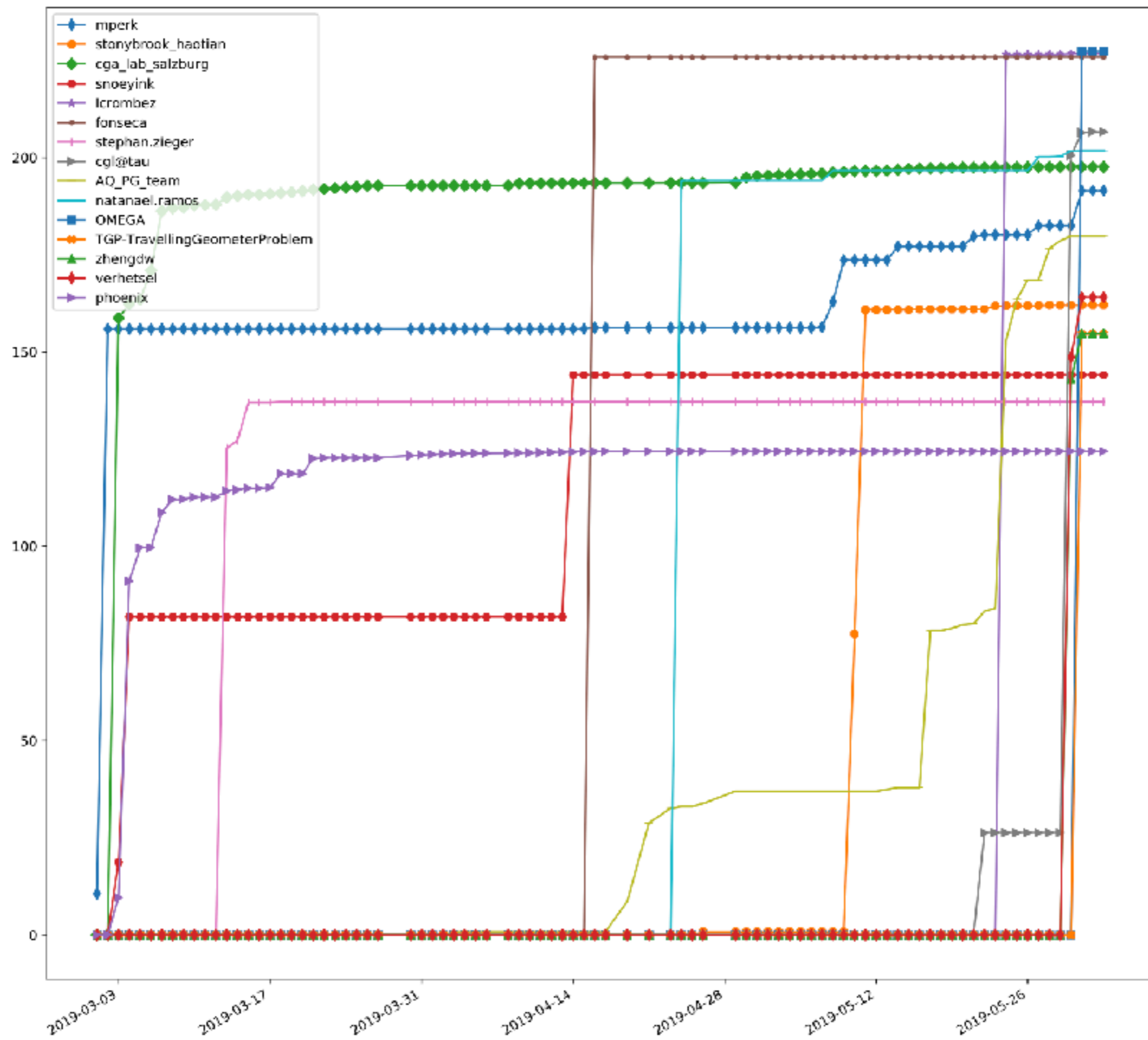
# Progress



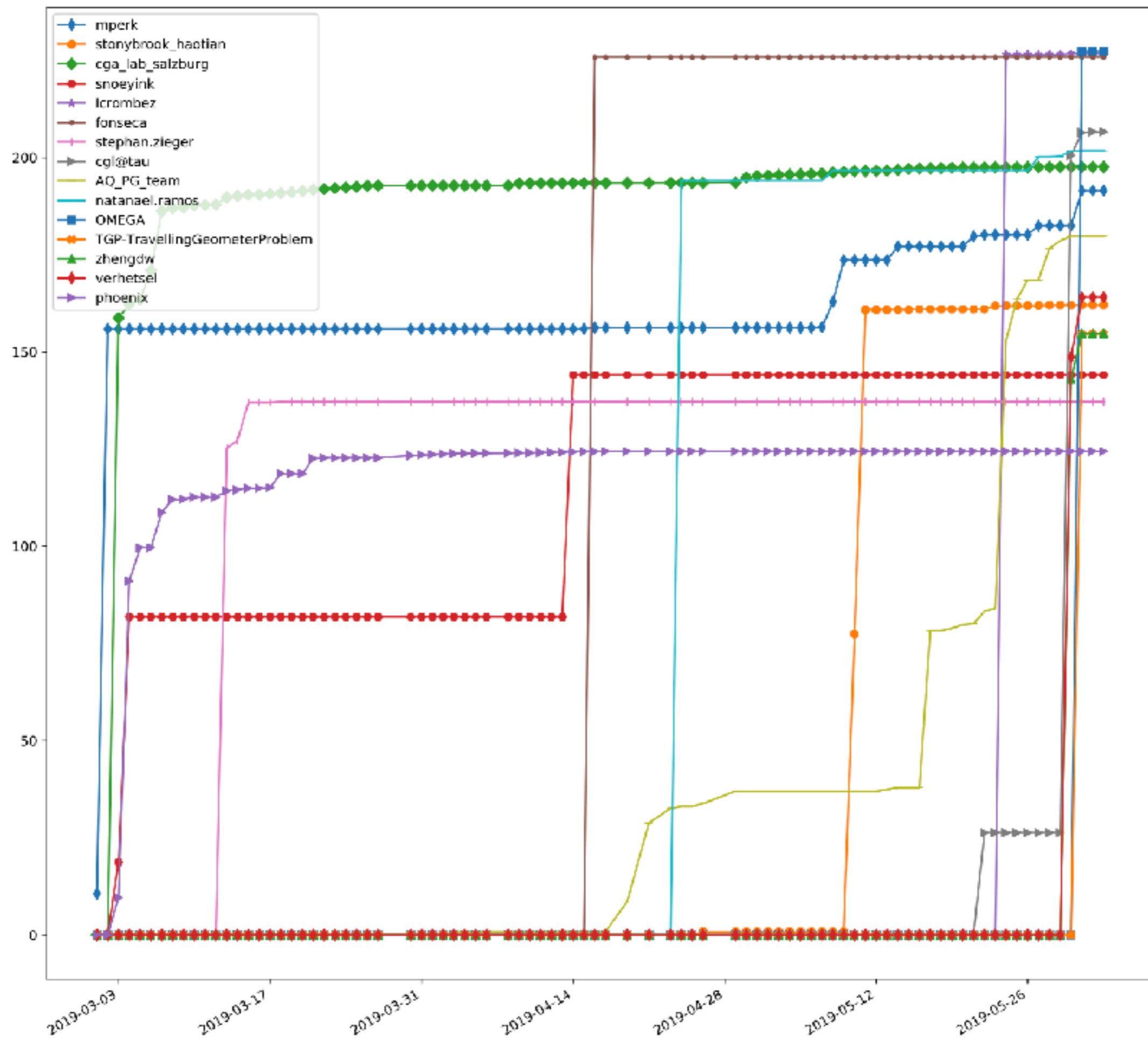






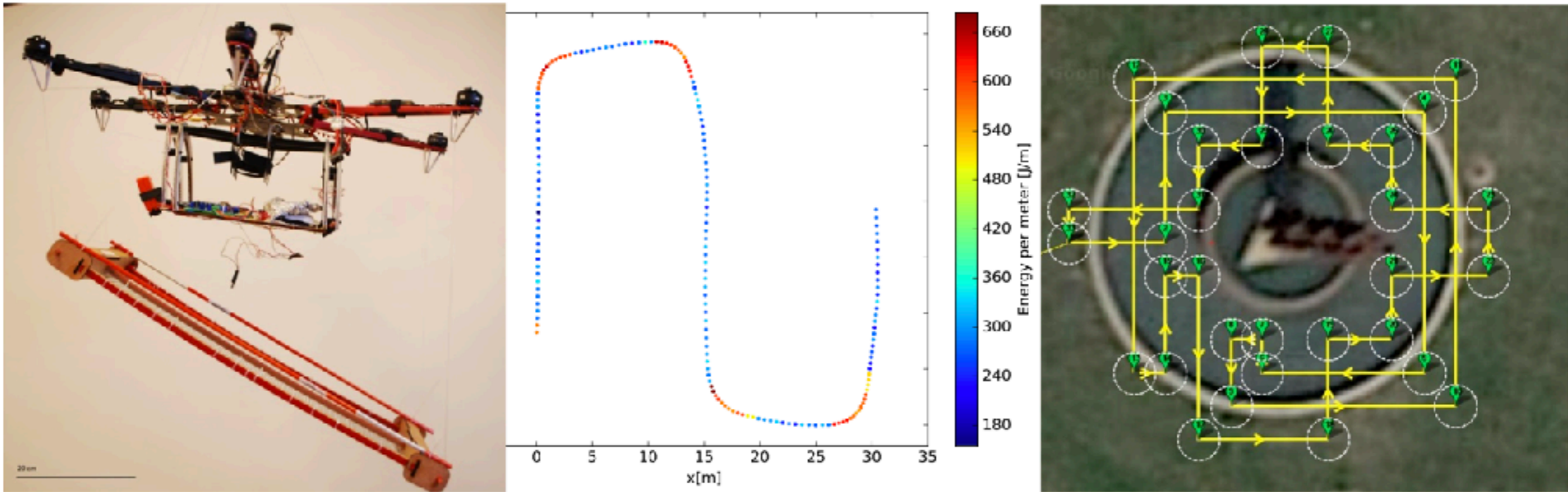






- 1. Introduction**
- 2. Longest Tours**
- 3. Stars and Matchings**
- 4. Nonsimple Polygons**
- 5. Optimal Area**
- 6. Turn Cost**

# Video!



## **Zapping Zika with a Mosquito-Managing Drone: Computing Optimal Flight Patterns with Minimum Turn Cost**

**Aaron Becker, Mustapha Debboun, Sándor P. Fekete  
Dominik Krupke, An Nguyen**



# Thank you!

# Thank you!



Mathematicians are weird.