CS 516 -1
Computational Geometry & Graph Drawing
(Spring 2013)

Today…

- go over admin details

  www.cs.ubc.ca/~kirk/cs516

- high-level discussion/illustration of topics
- start a (motivating) example: finding near neighbours (within fixed distance)

What *is* computational geometry?

- The study of geometric problems from a computational (algorithmic) perspective

What *is* computational geometry?

- The study of geometric problems from a computational (algorithmic) perspective
- The study of (familiar) computational problems under geometric constraints

What *is* computational geometry?

- The study of geometric problems from a computational (algorithmic) perspective
- The study of (familiar) computational problems under geometric constraints
- Note: it means other things to other people

What *is* computational geometry?

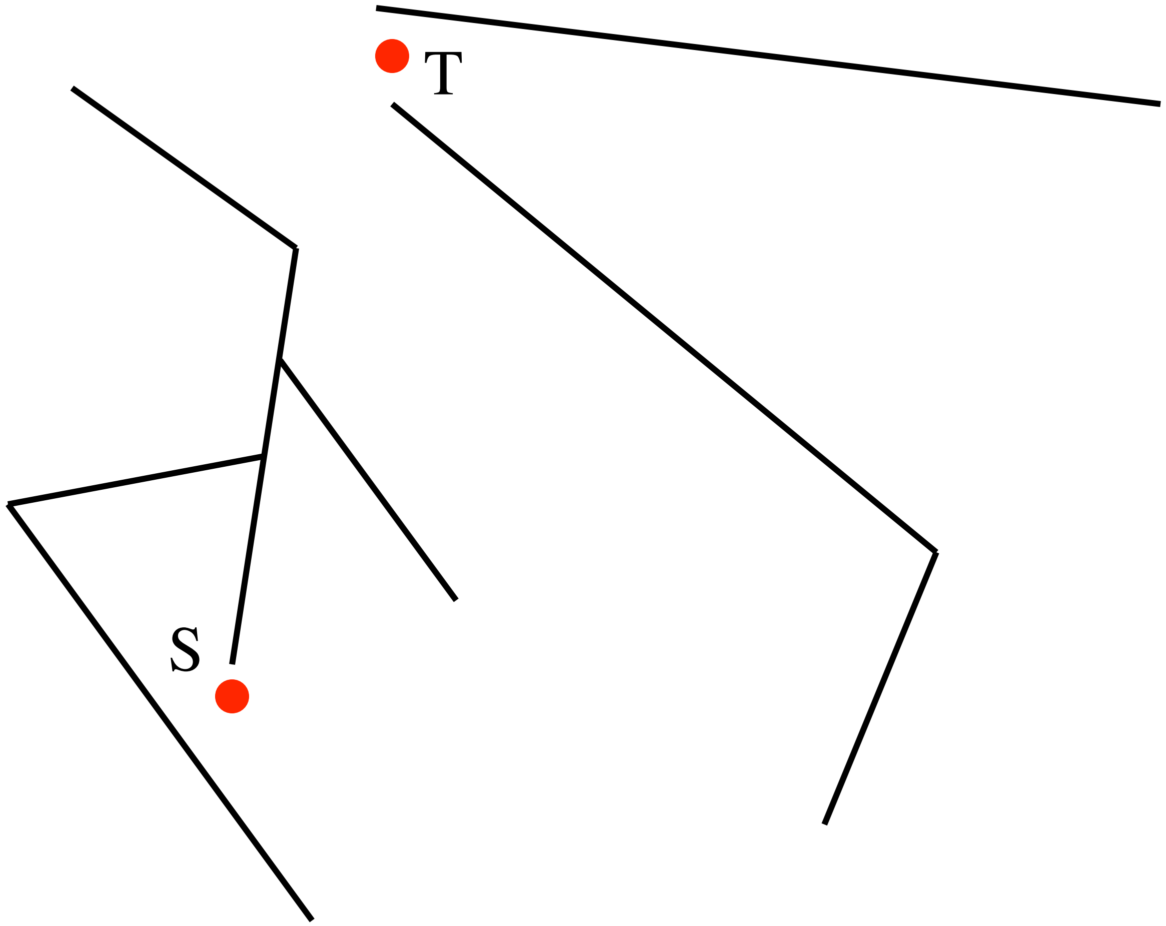
Topics in surface modeling: b-splines, non-uniform rational b-splines, physically based deformable surfaces, sweeps and generalized cylinders, offsets, blending and filleting surfaces. Non-linear solvers and intersection problems. Solid modeling: constructive solid geometry, boundary representation, non-manifold and mixed-dimension boundary representation models, octrees. Robustness of geometric computations. Interval methods. Finite and boundary element discretization methods for continuum mechanics problems. Scientific visualization. Variational geometry. Tolerances. Inspection methods. Feature representation and recognition. Shape interrogation for design, analysis, and manufacturing. Involves analytical and programming assignments.

How does it relate to graph drawing?

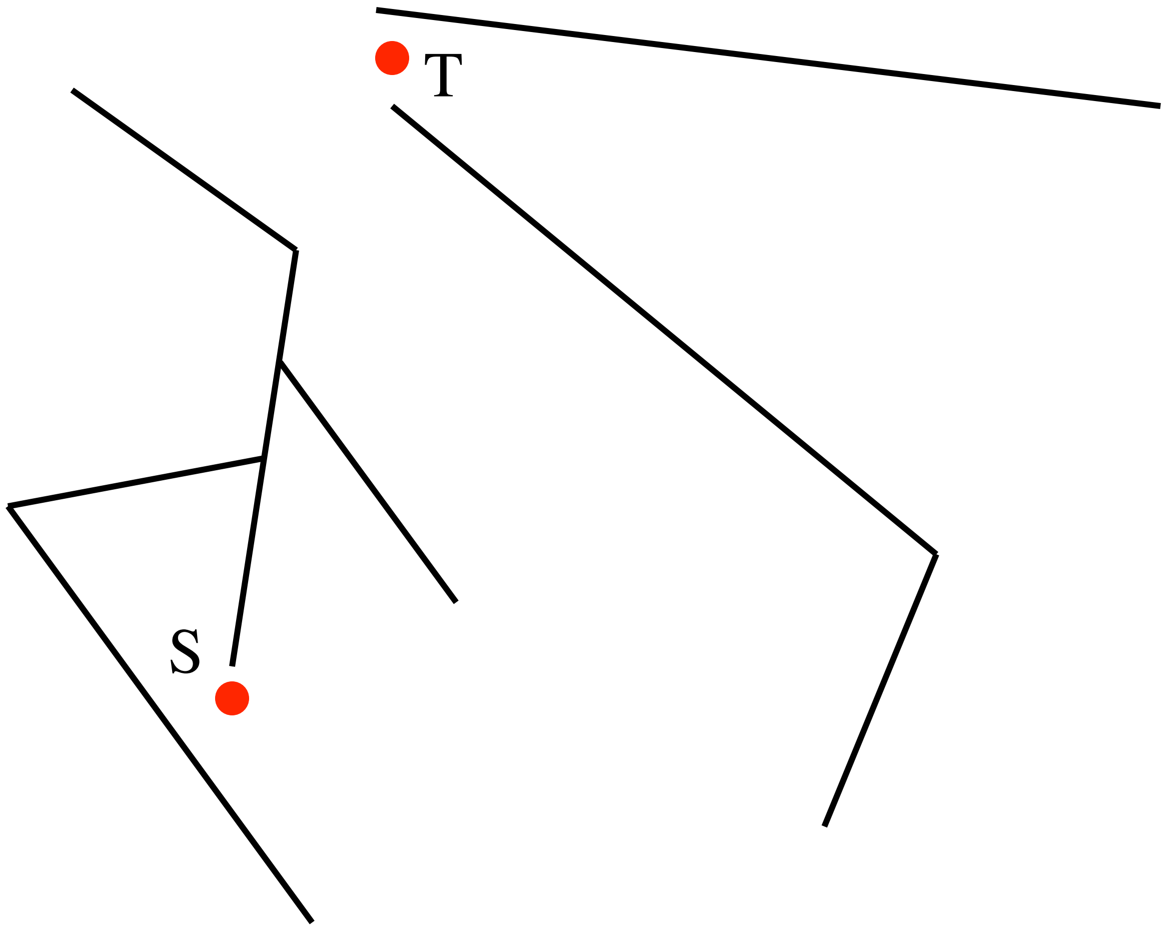
- Graph drawing uses methods from computational geometry, as well as geometric graph theory and information visualization, to construct pleasing and informative renderings of graphs arising from areas such as social networks, cartography, and bioinformatics.
- Computational geometry looks for the structure implicit in configurations of geometric objects. Graph drawing seeks geometric configurations that realize/embody given abstract structures.

An illustrative example

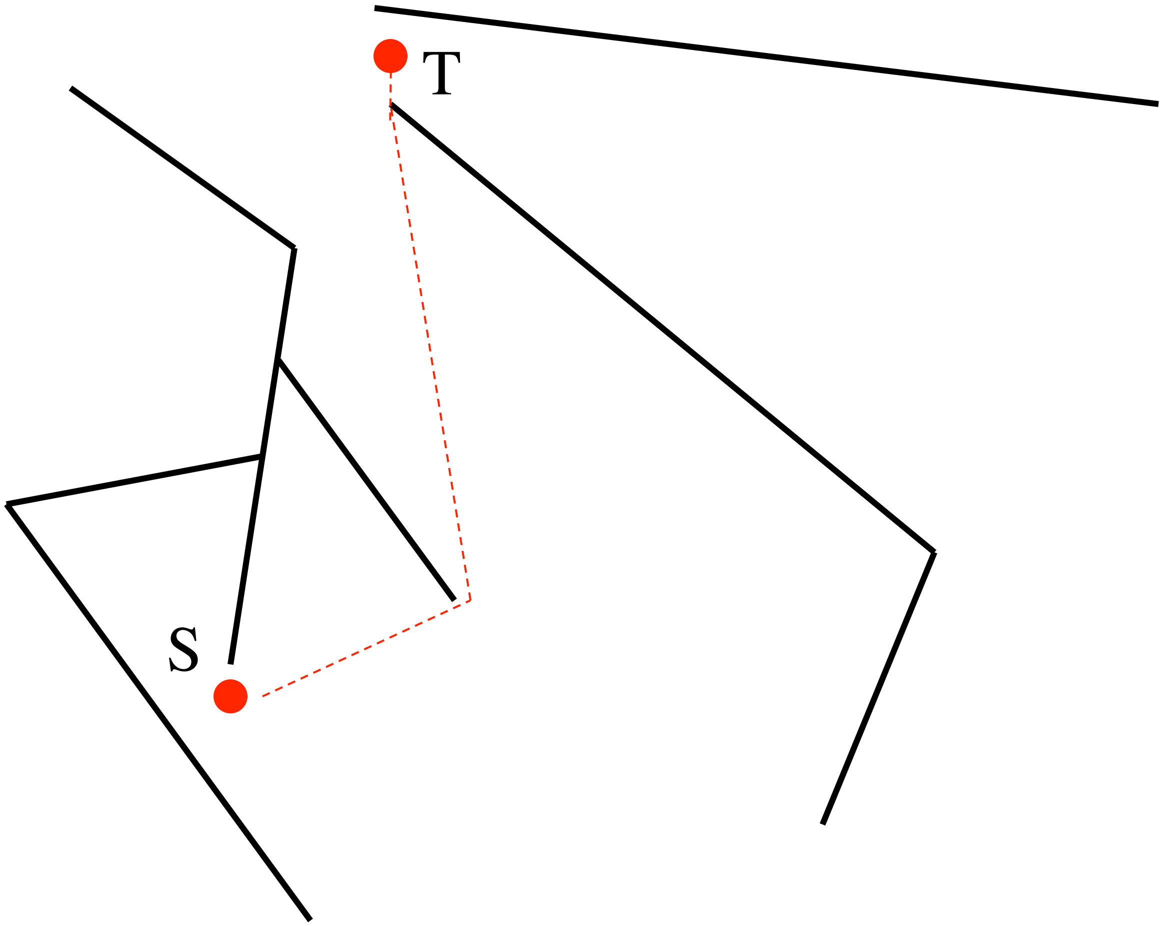
- Finding an *optimal* path from here to there…



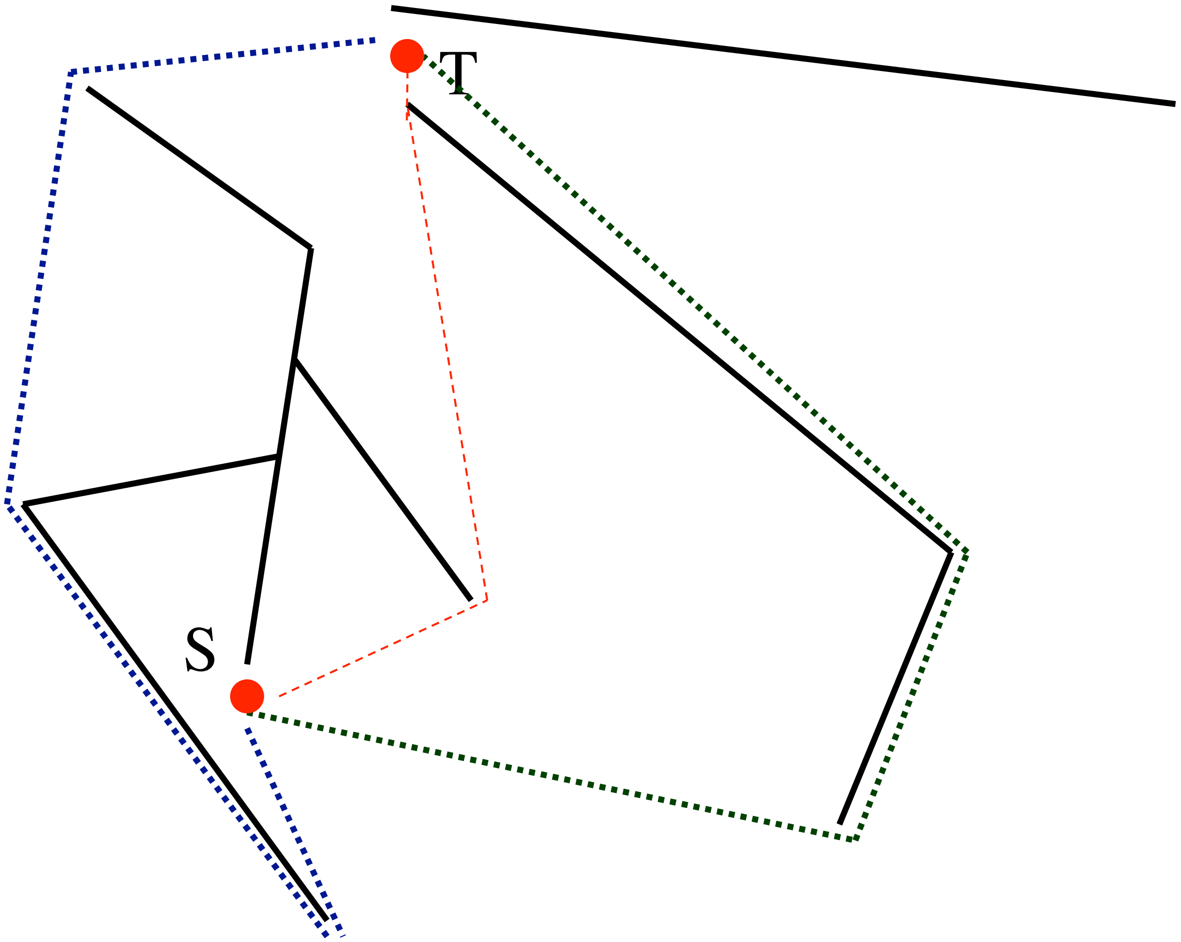
planning optimal motion in the plane
is “easy”



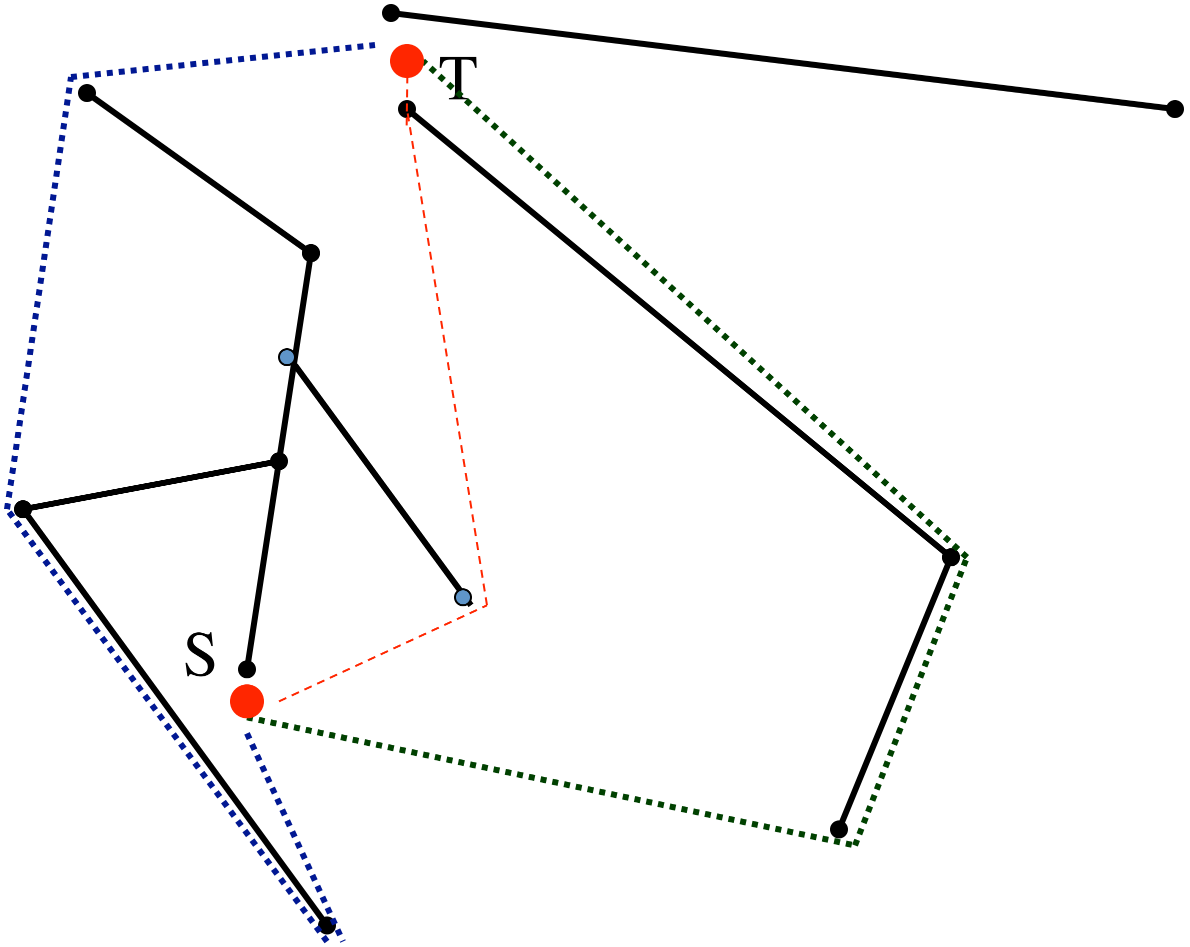
shortest motion



shortest motion

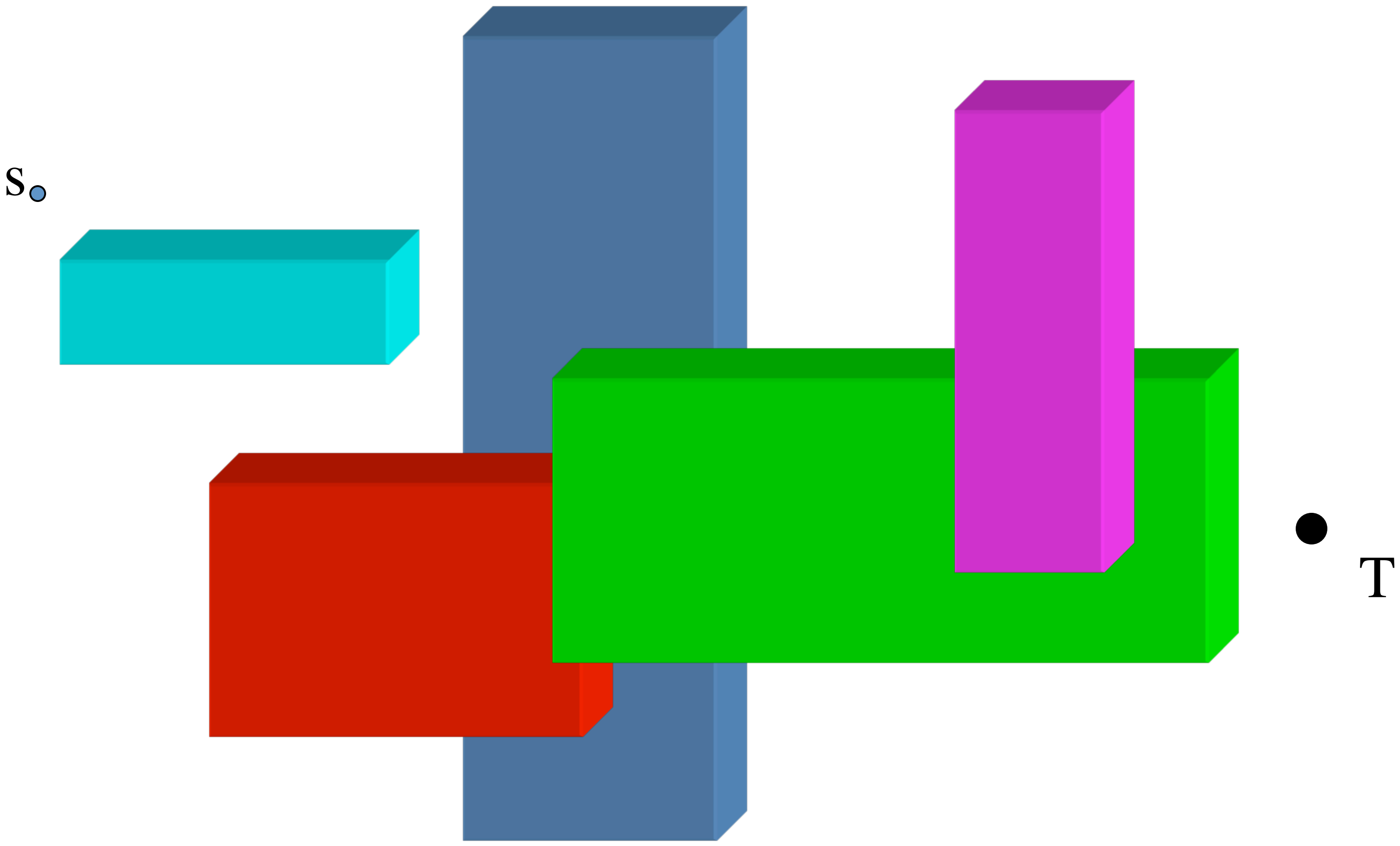


other locally shortest motions

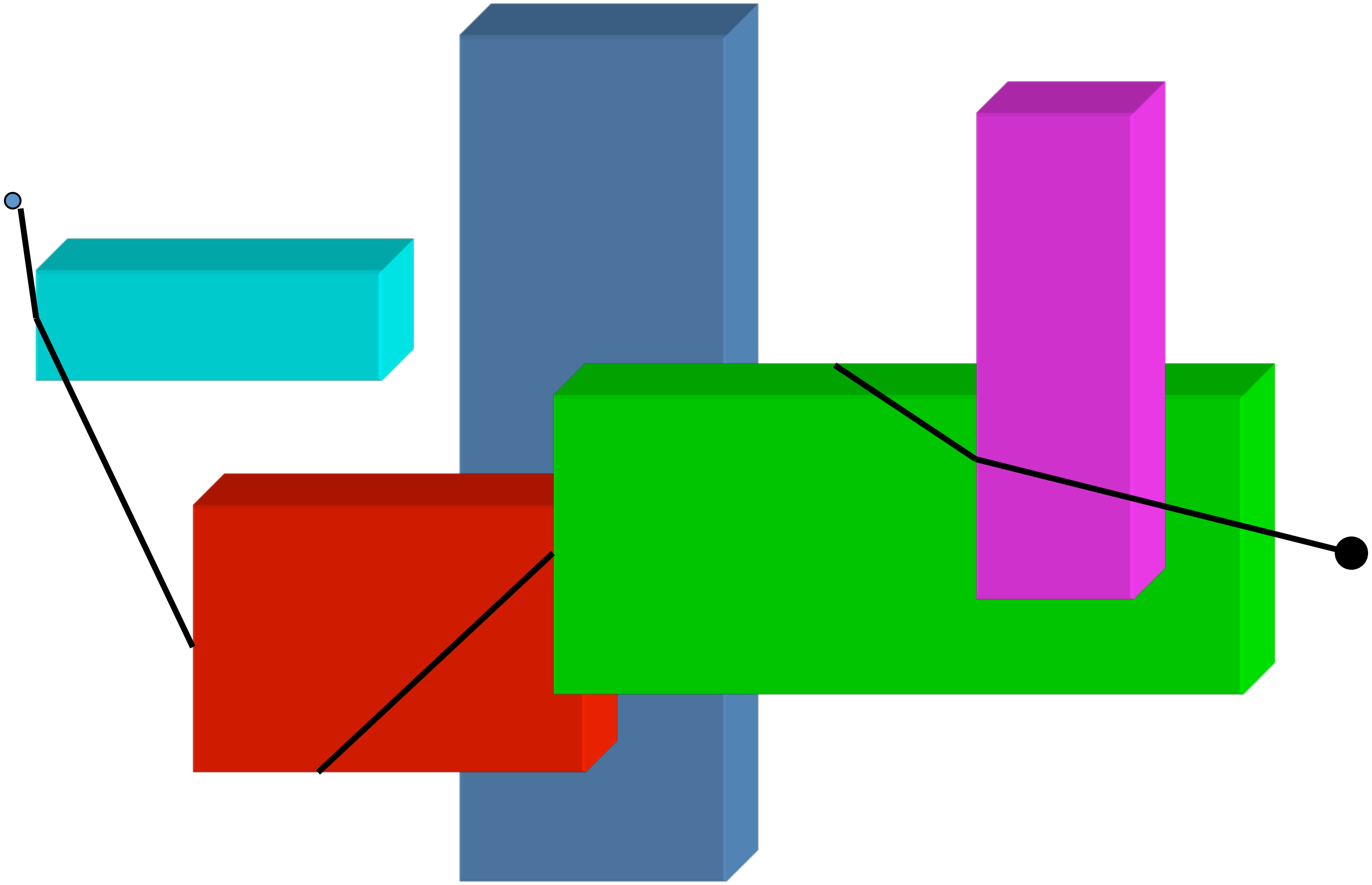


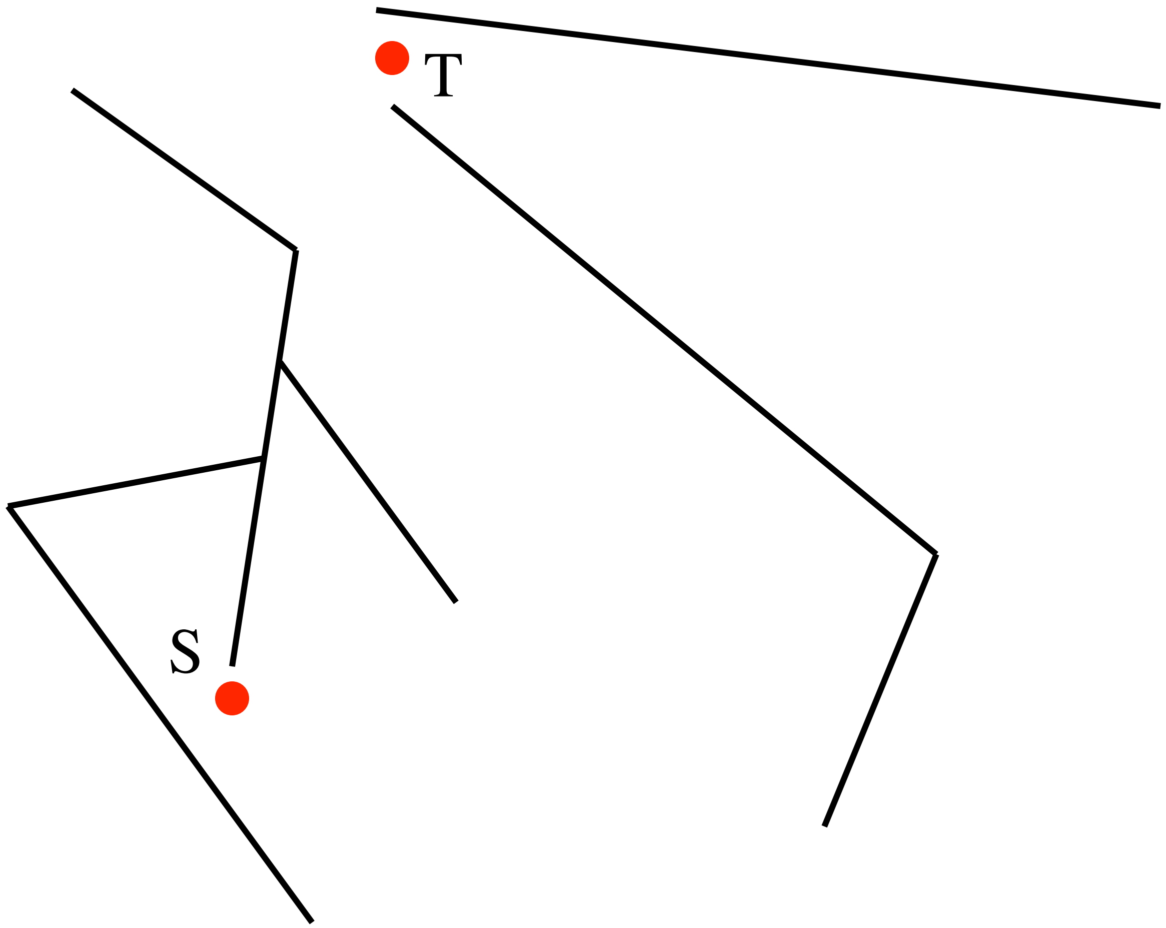
reduction to shortest path
in a (visibility) graph

three-dimensional obstacles

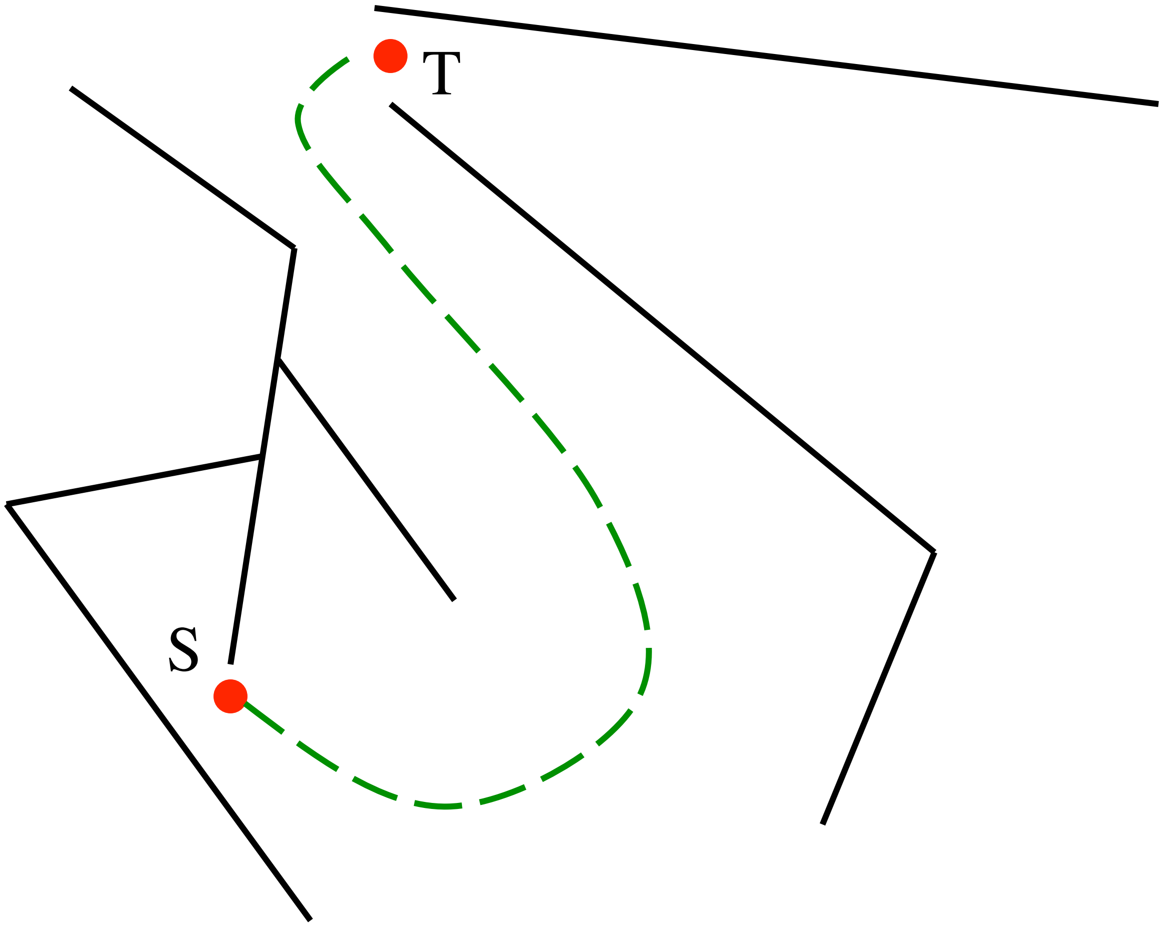


bends could occur anywhere along edges

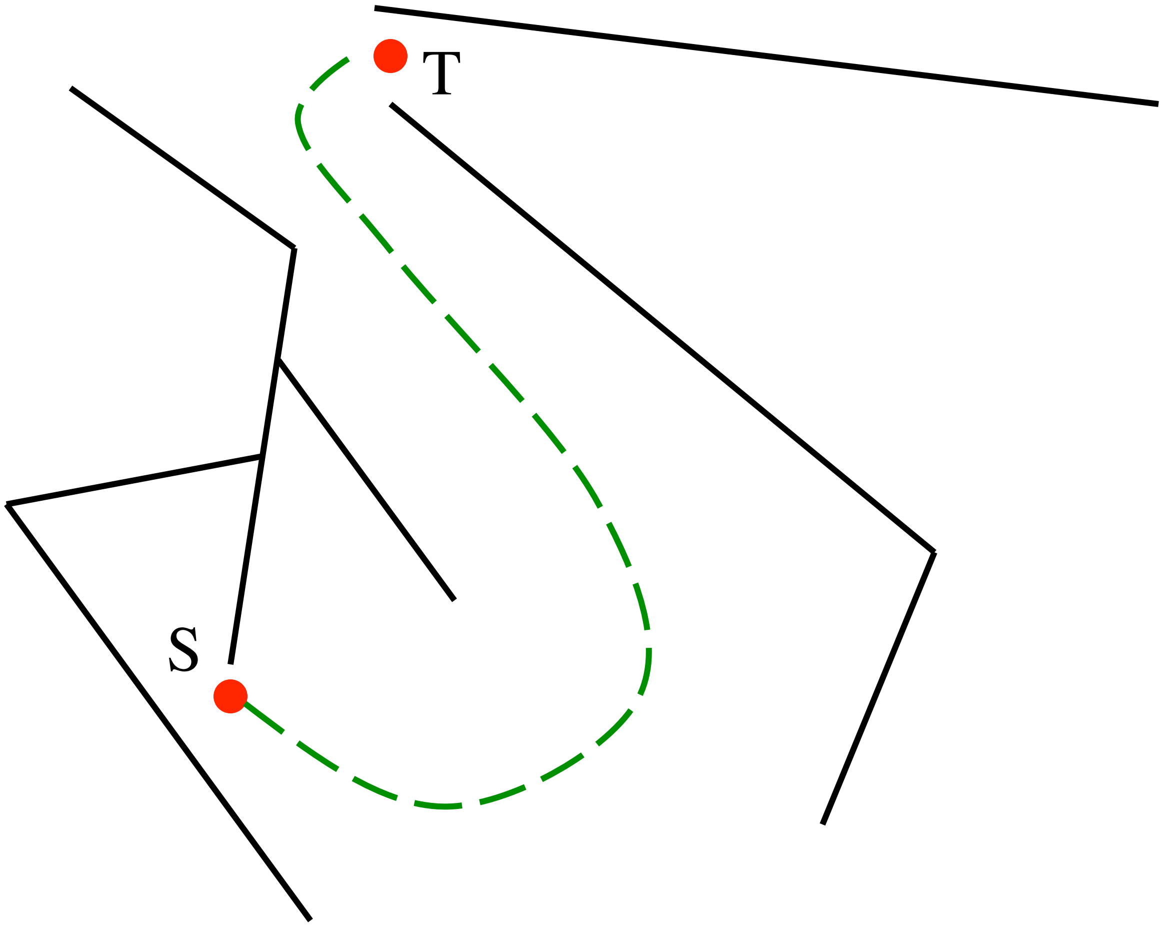




other notions of optimality:



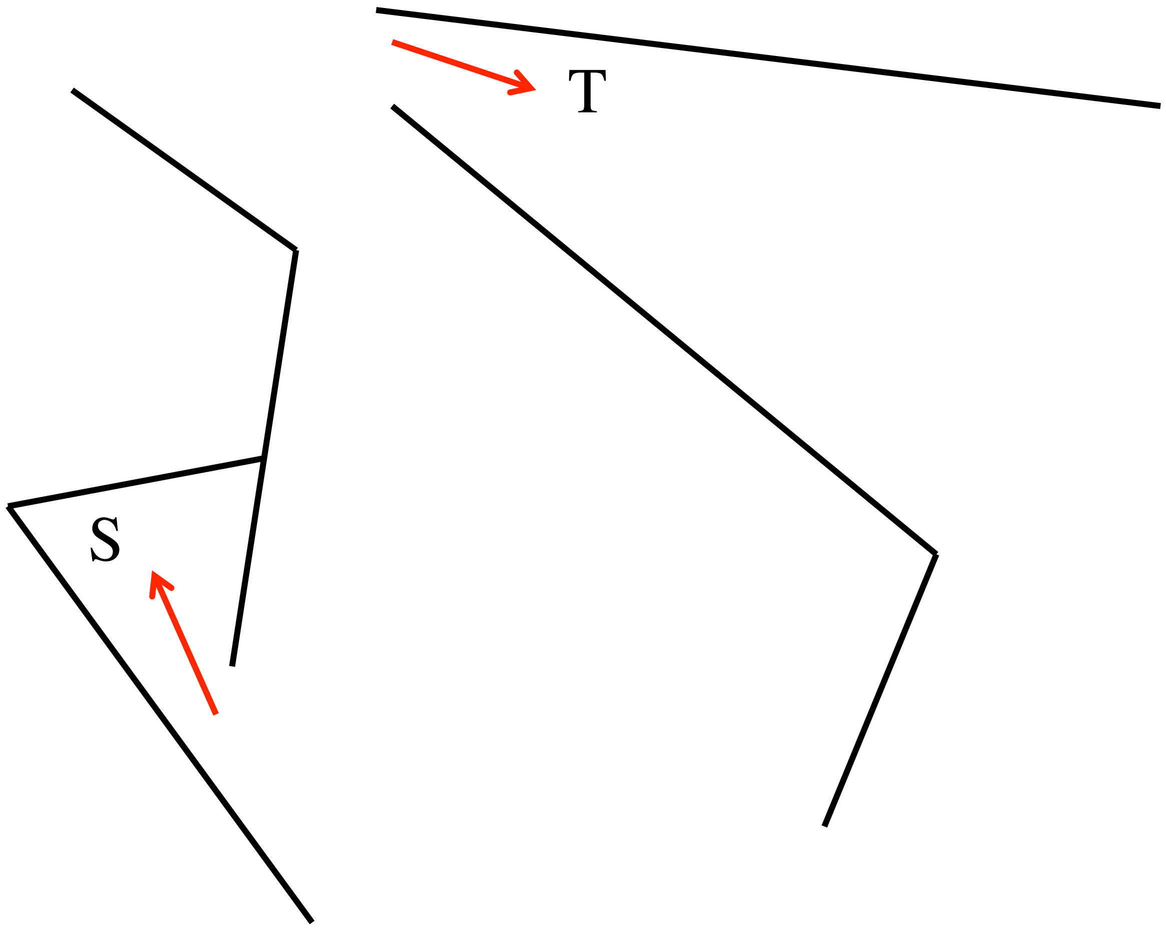
other notions of optimality:
maximum clearance



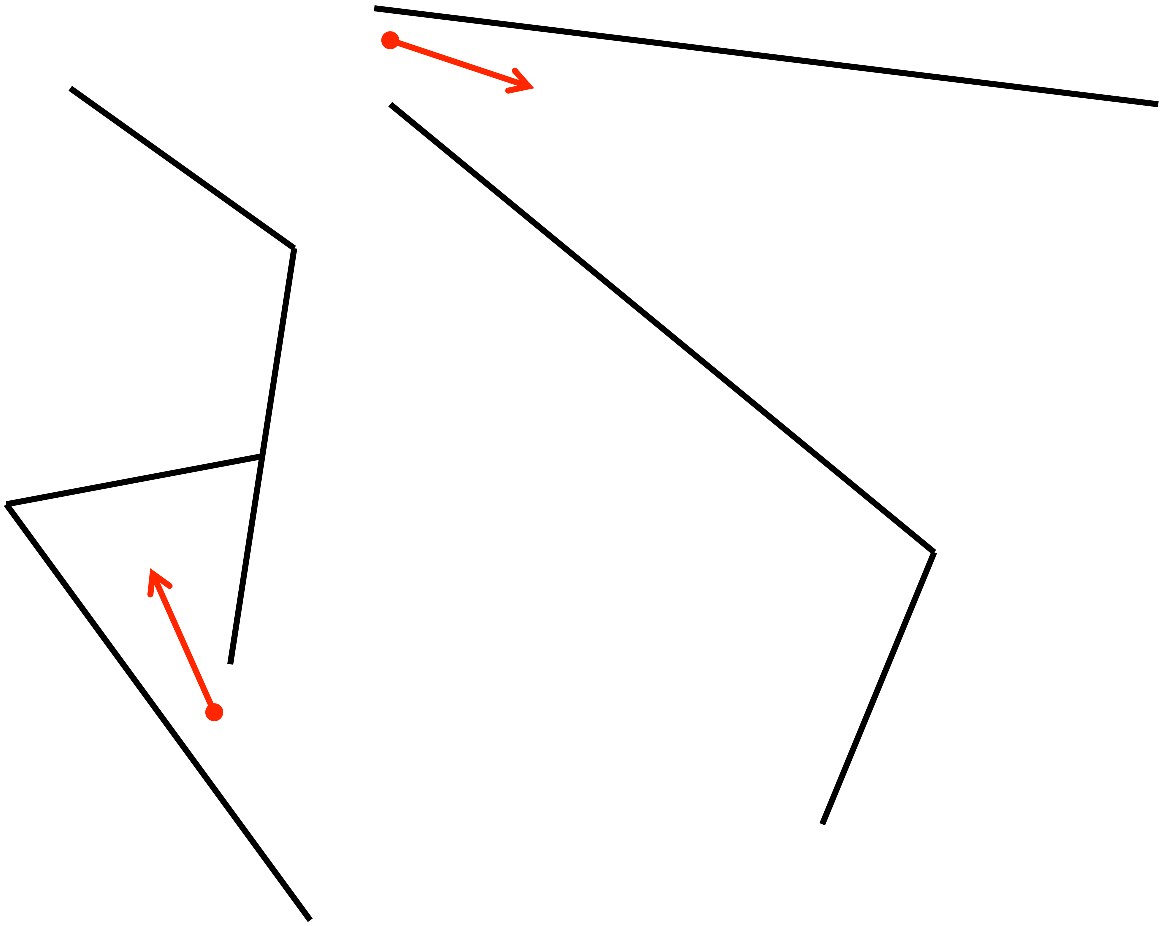
other notions of optimality:

maximum clearance

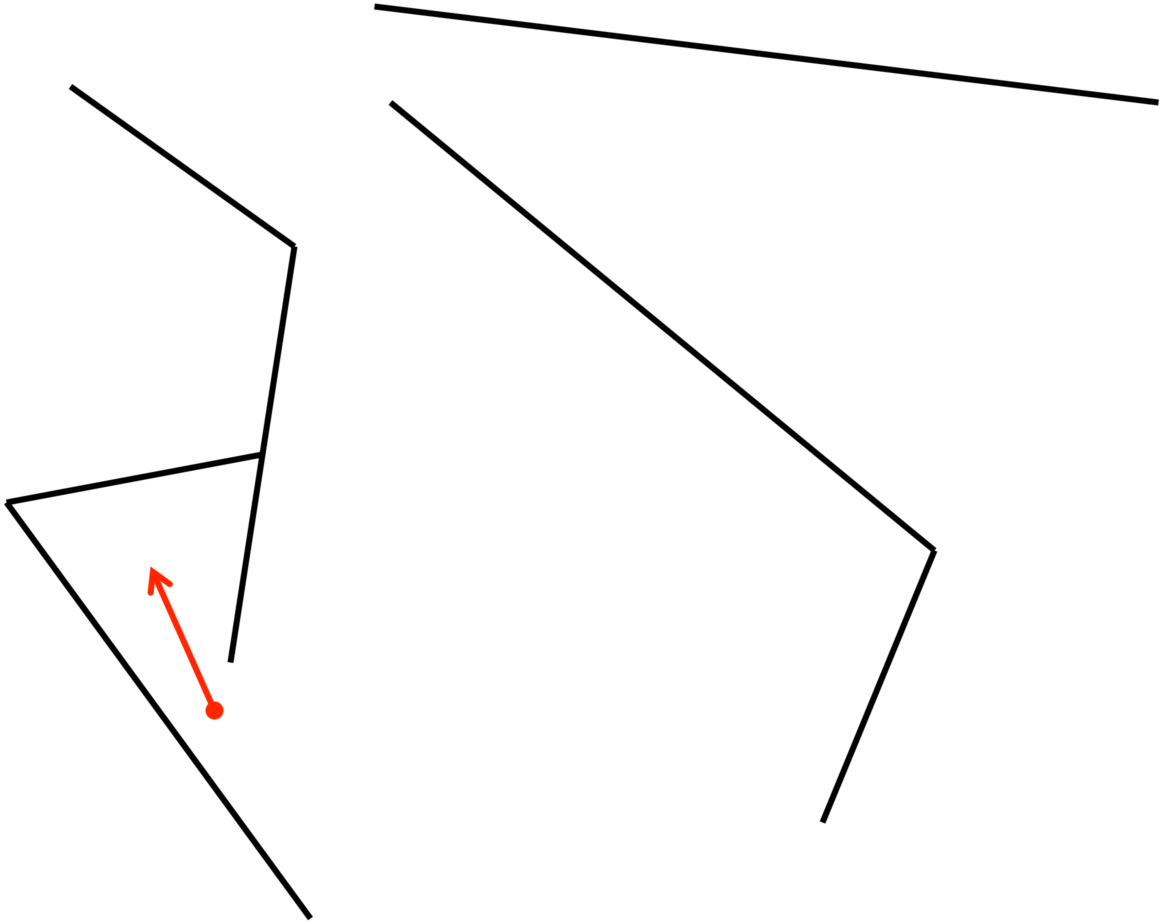
minimum curvature

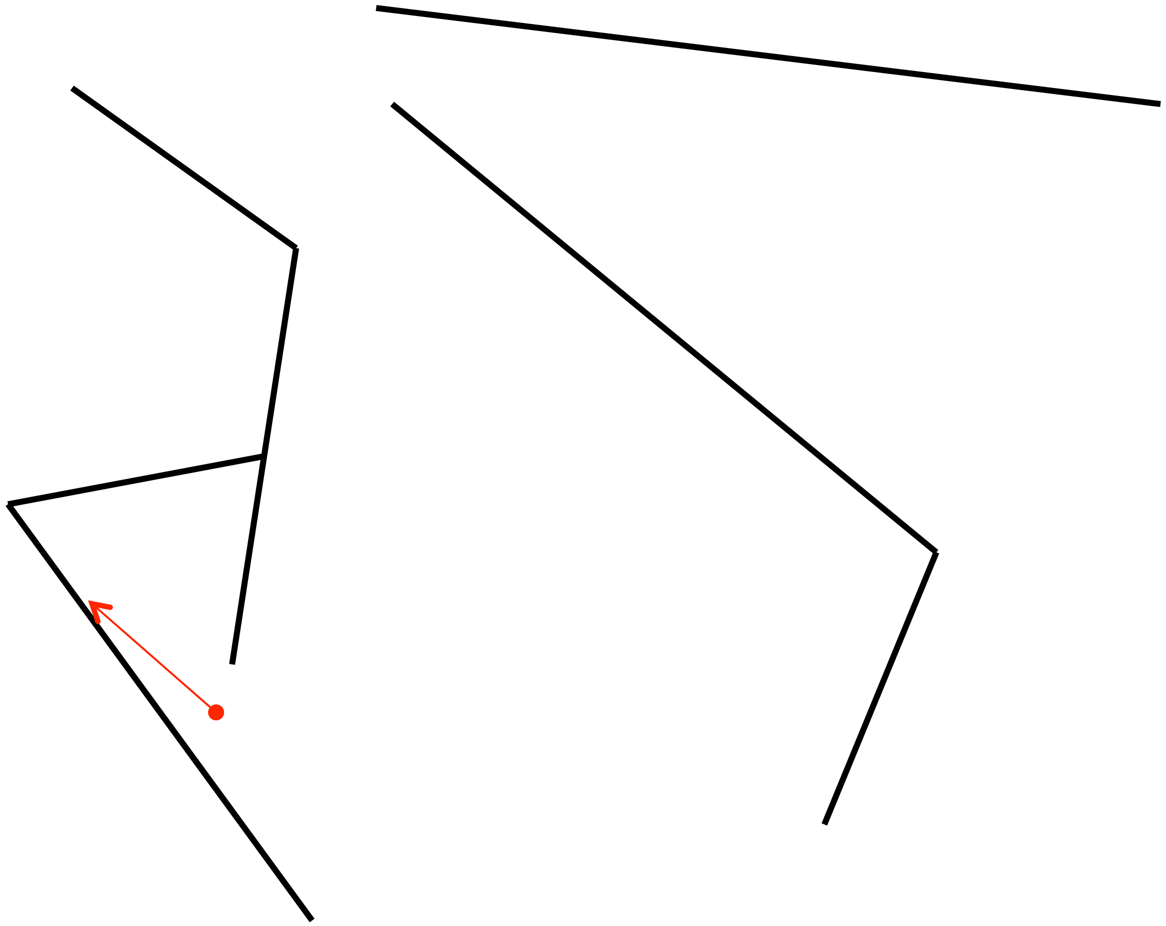


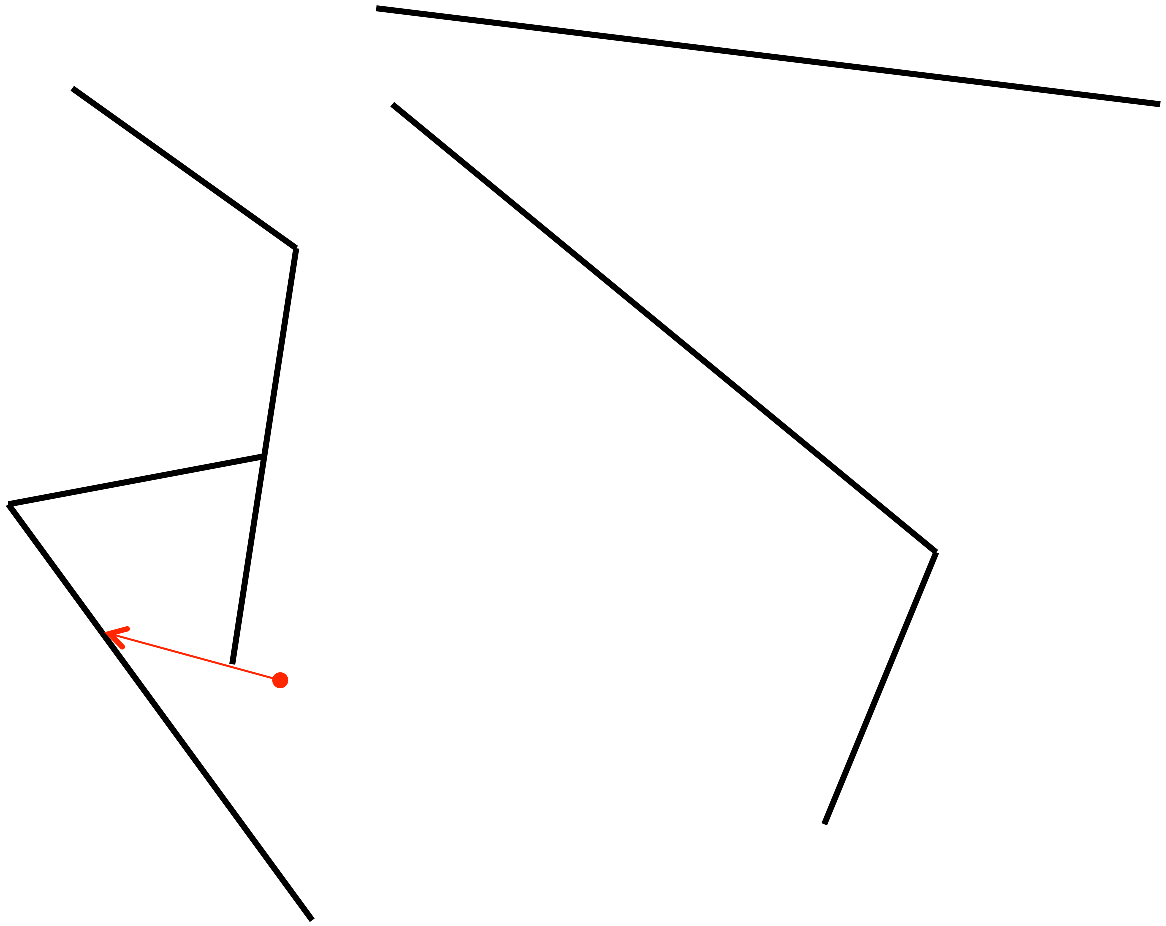
what is the shortest motion?

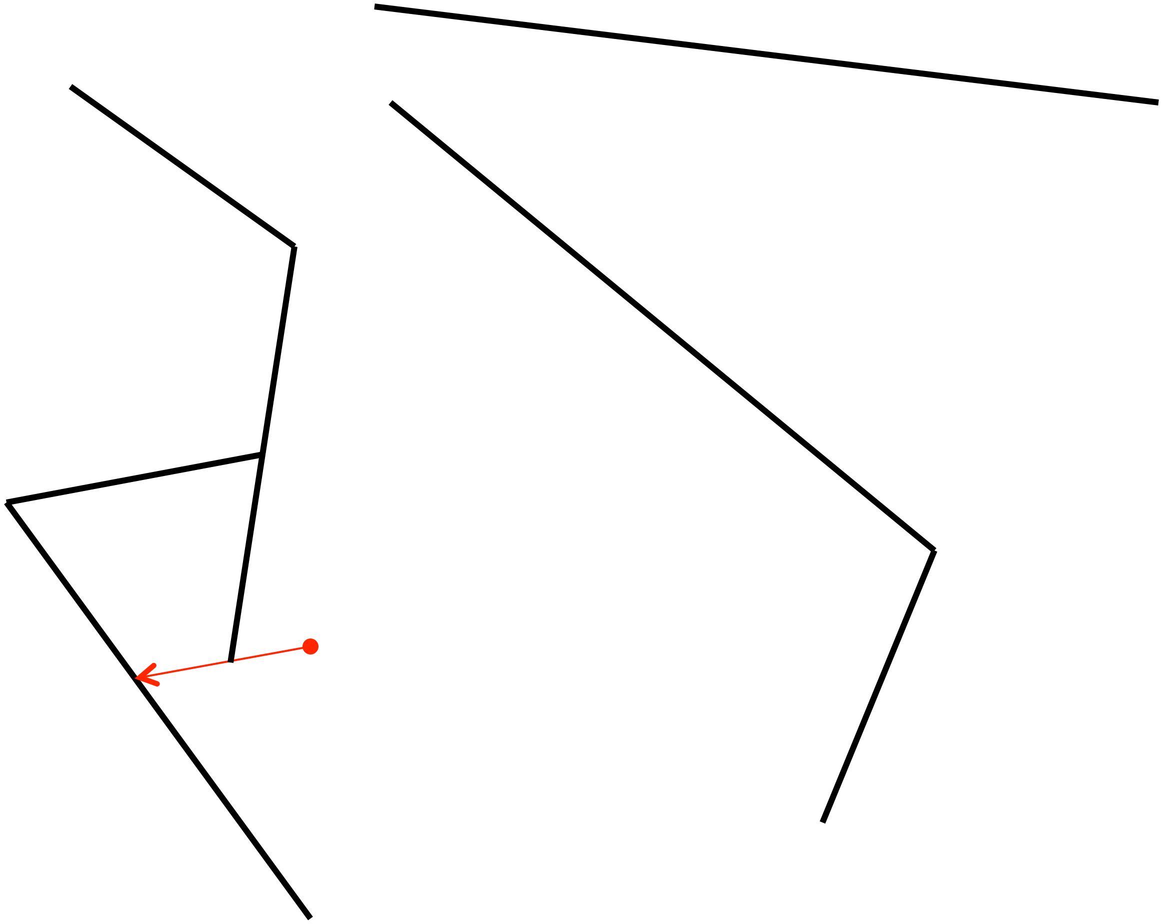


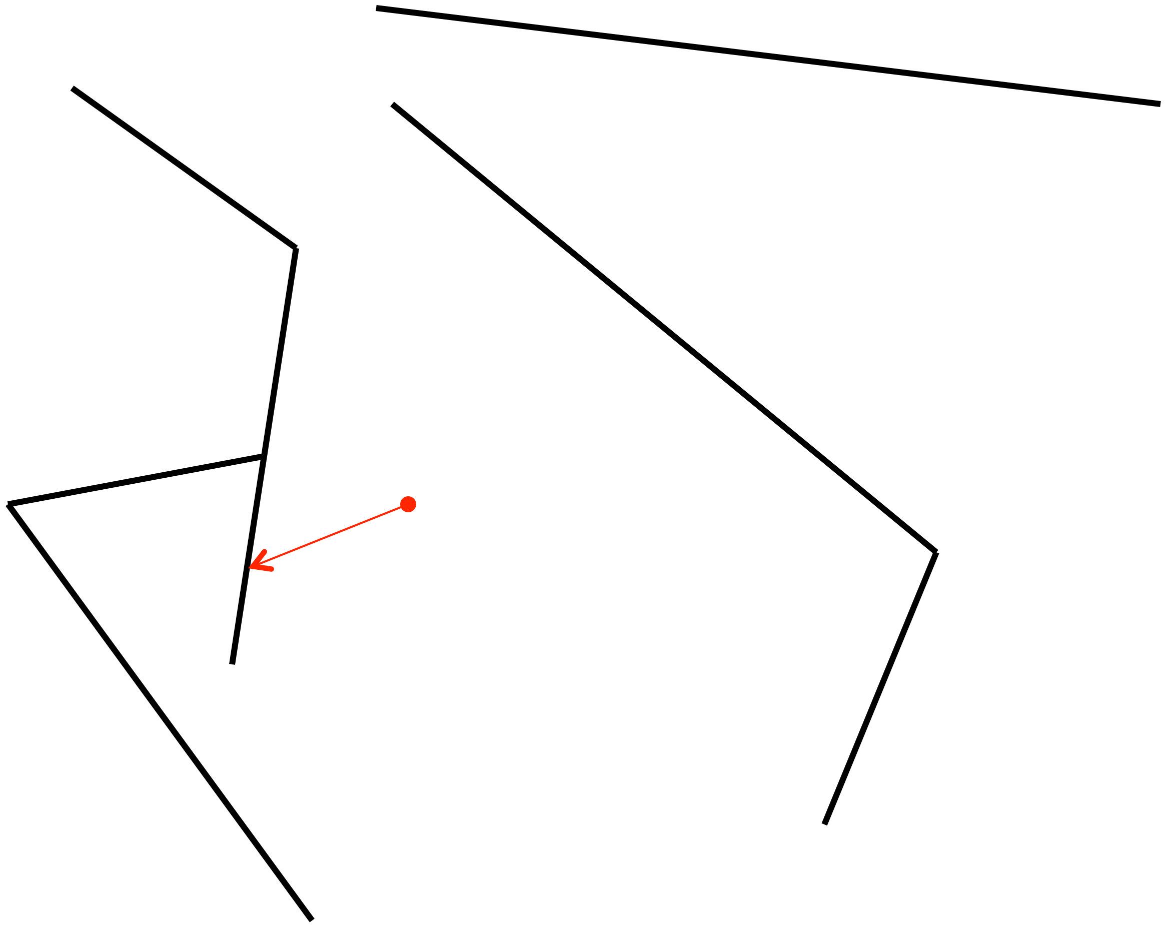
fix a point and minimize its trace

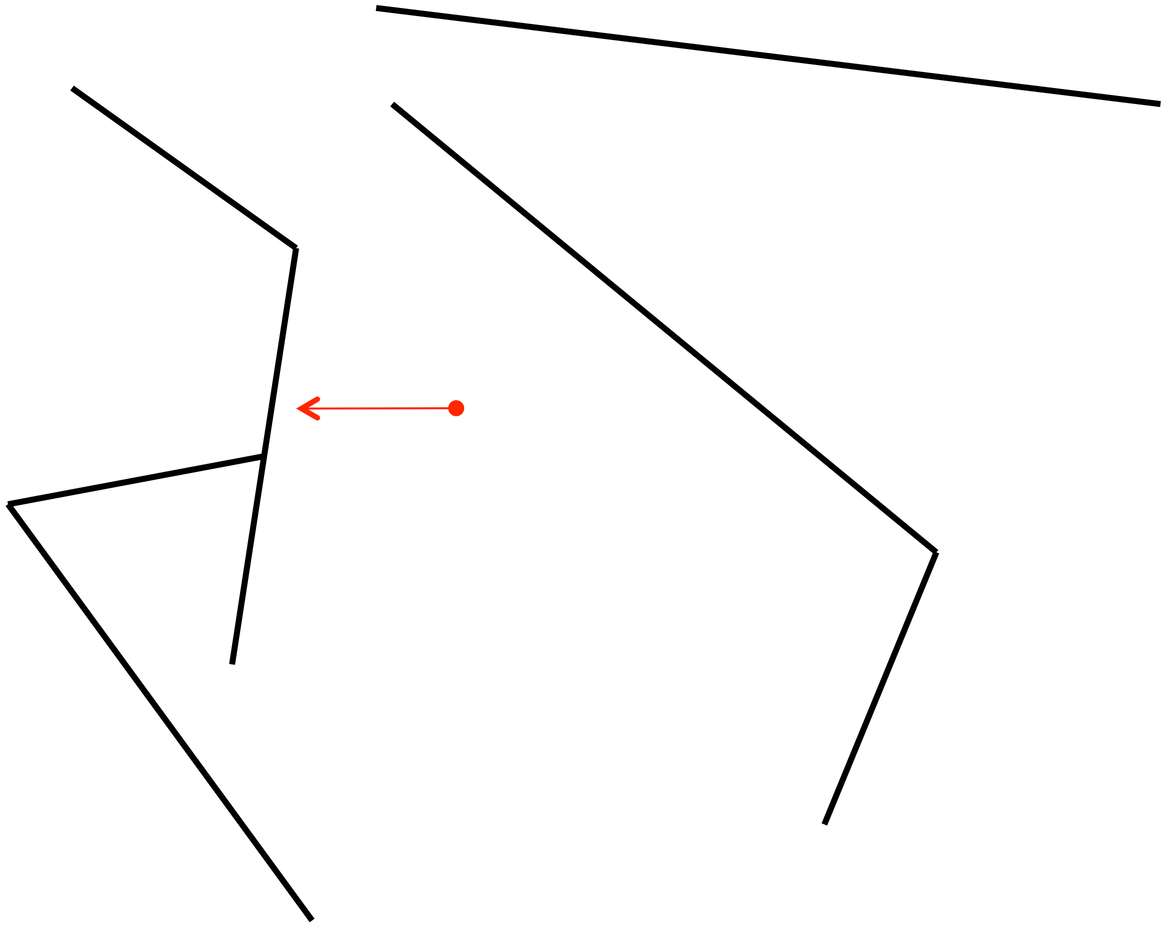


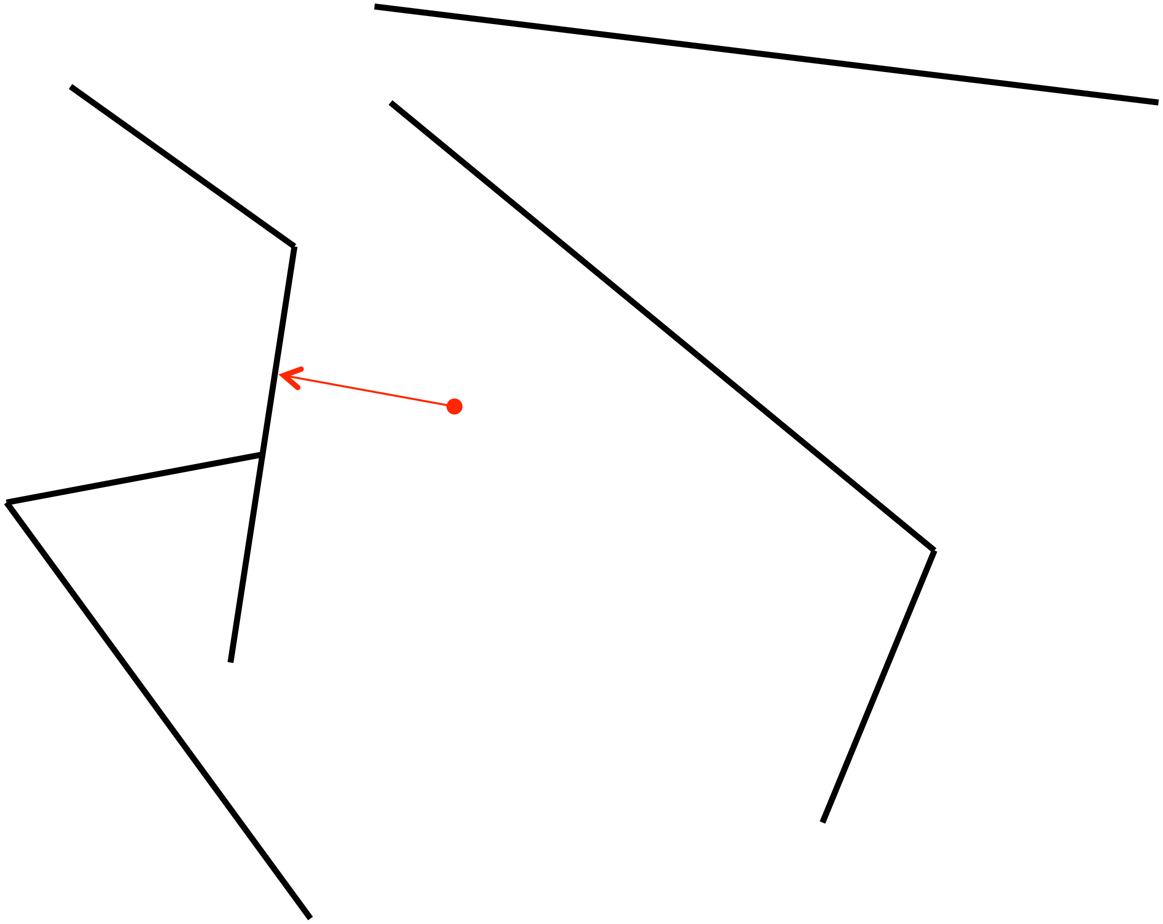


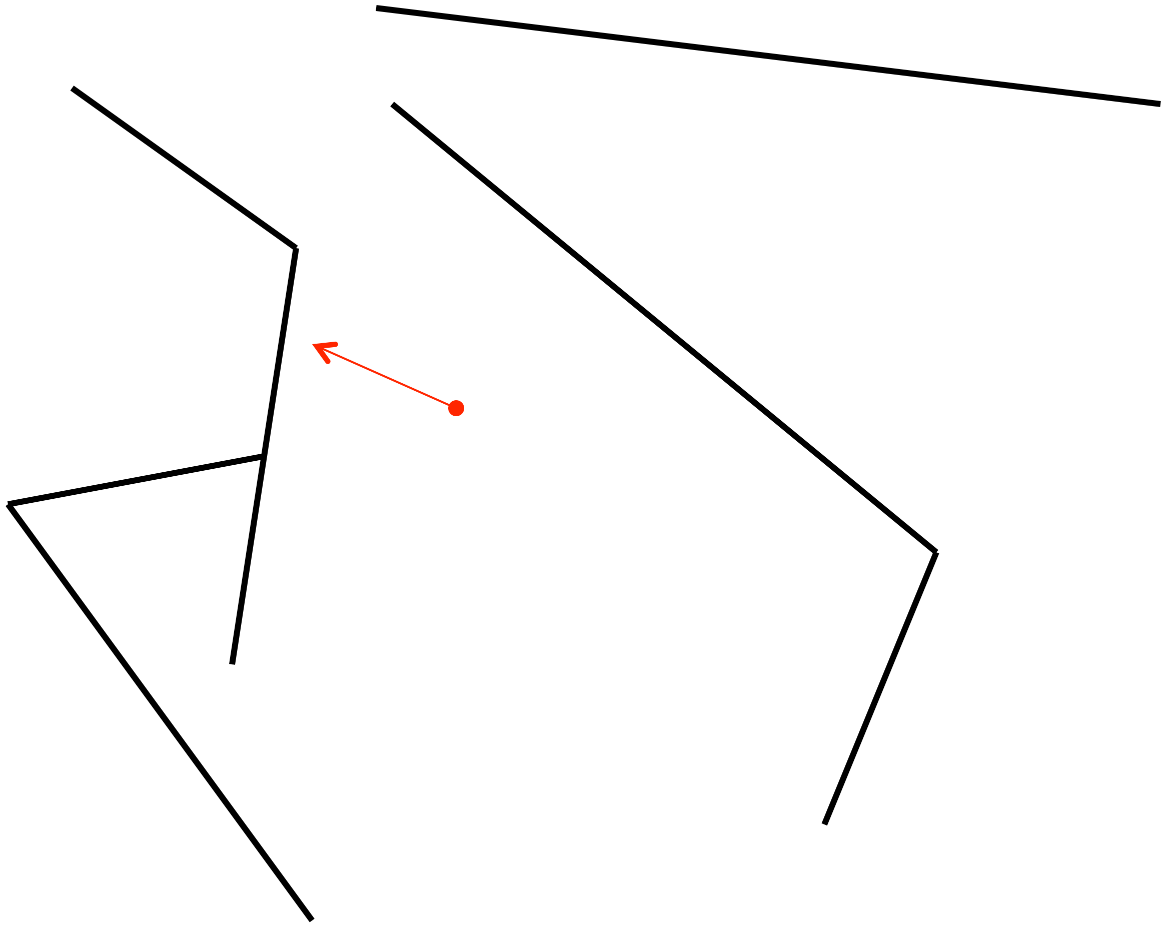


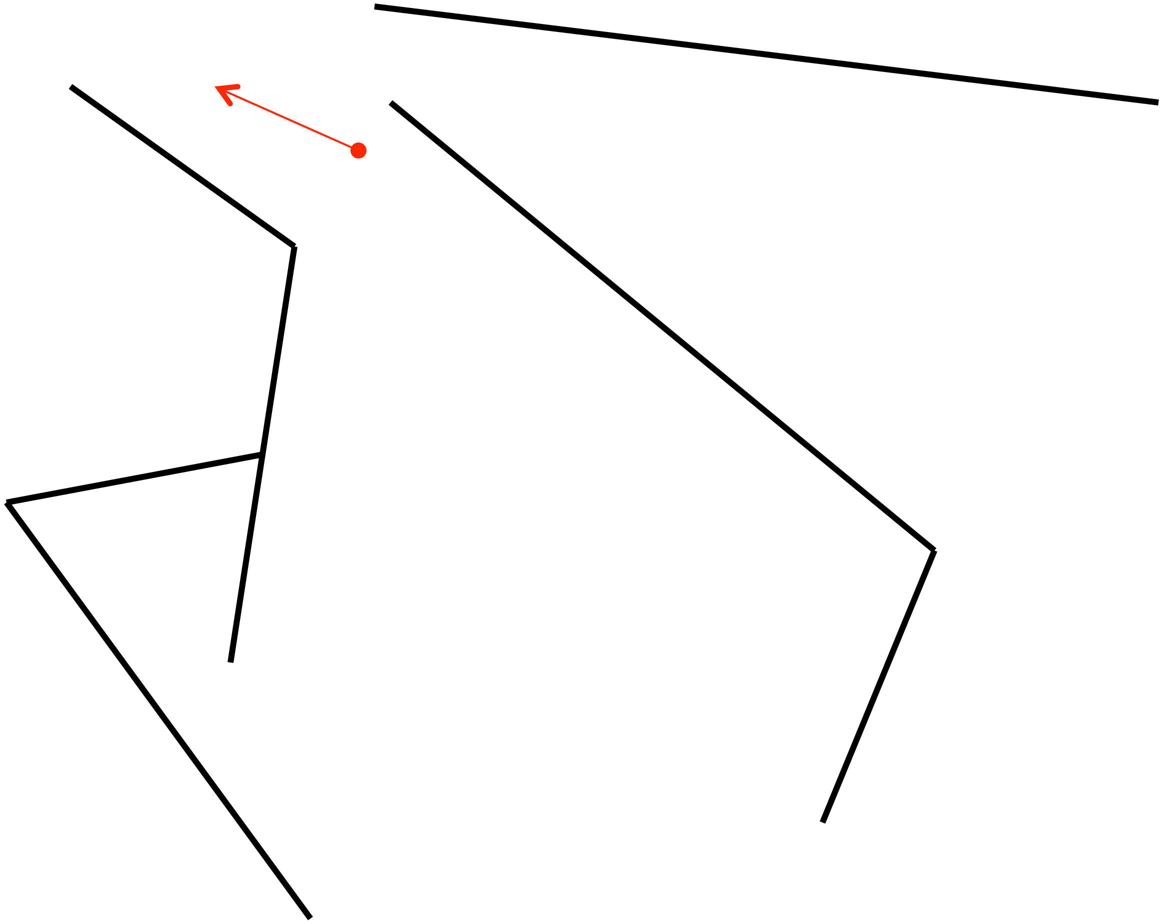


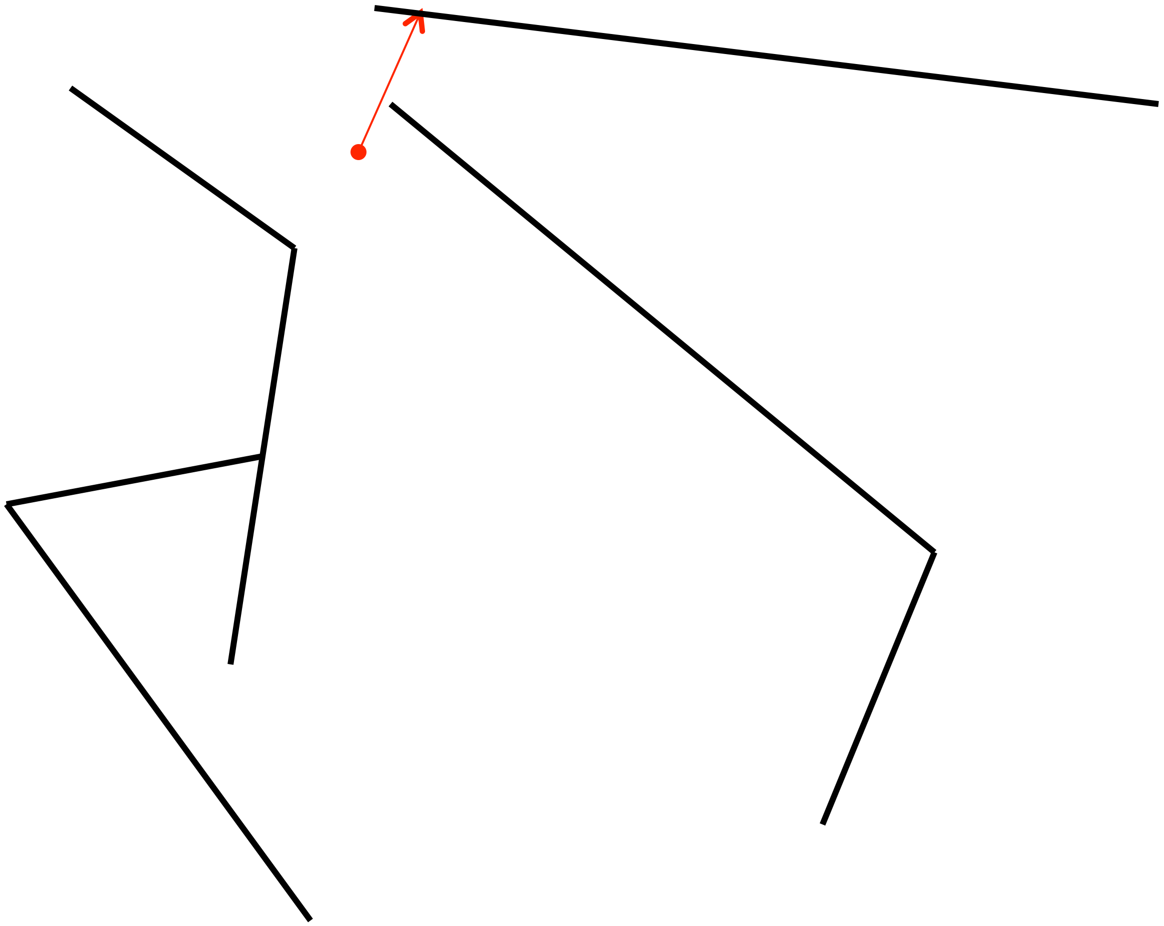


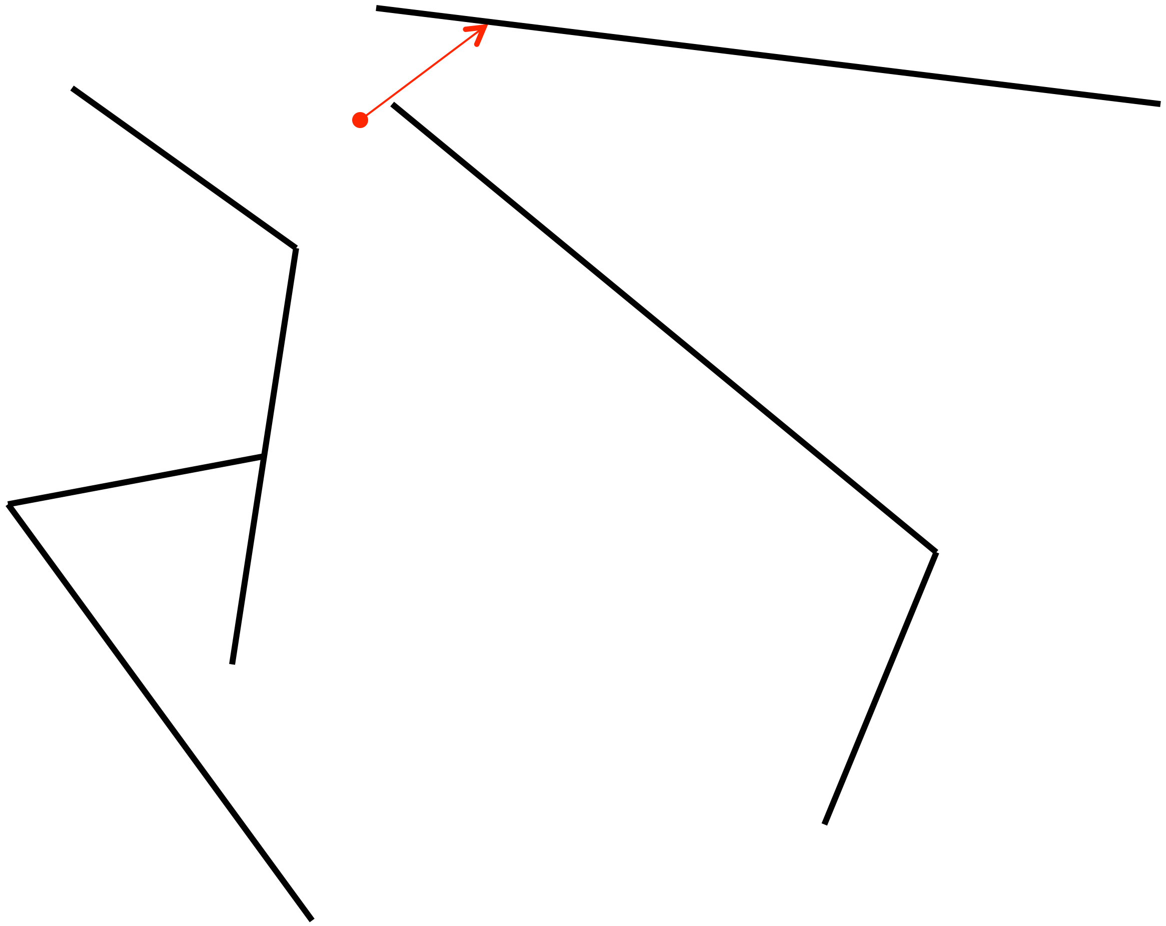


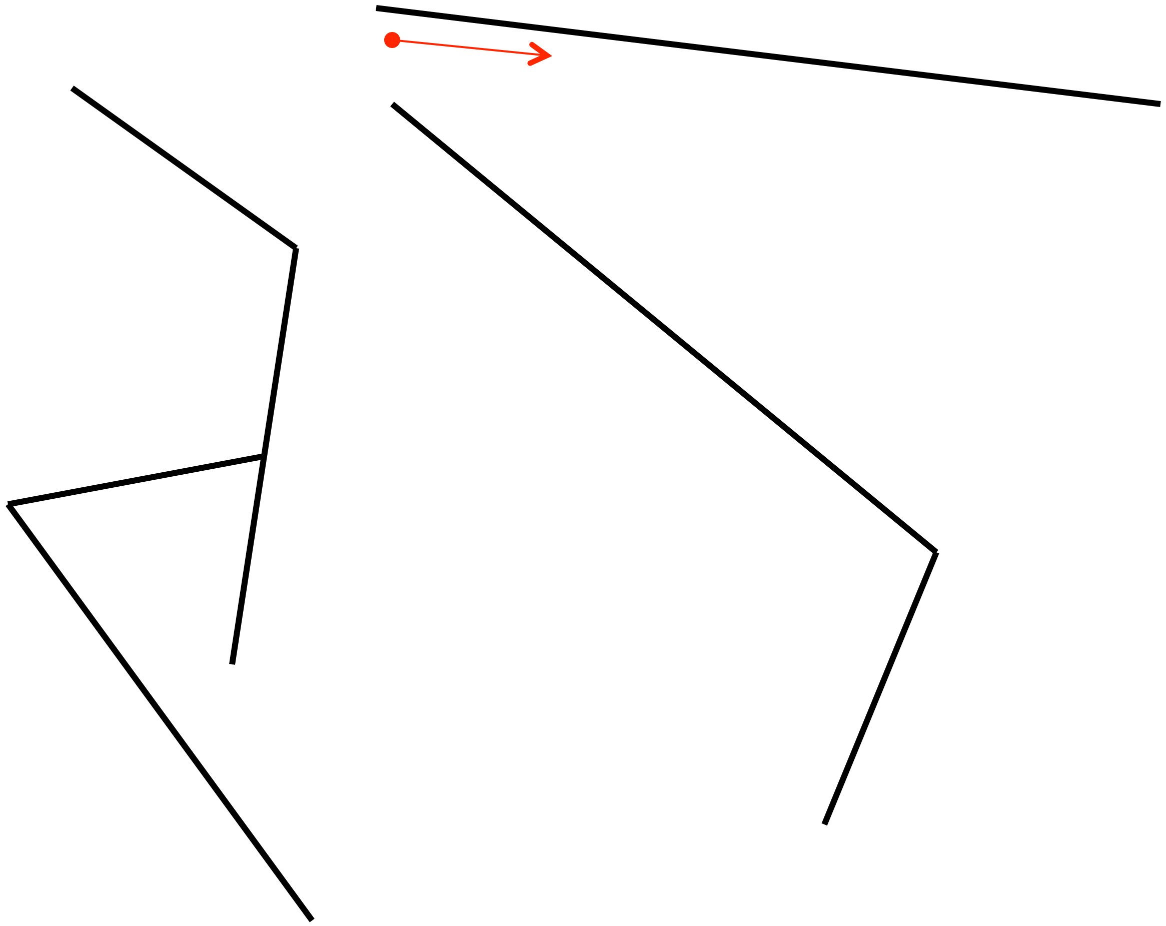


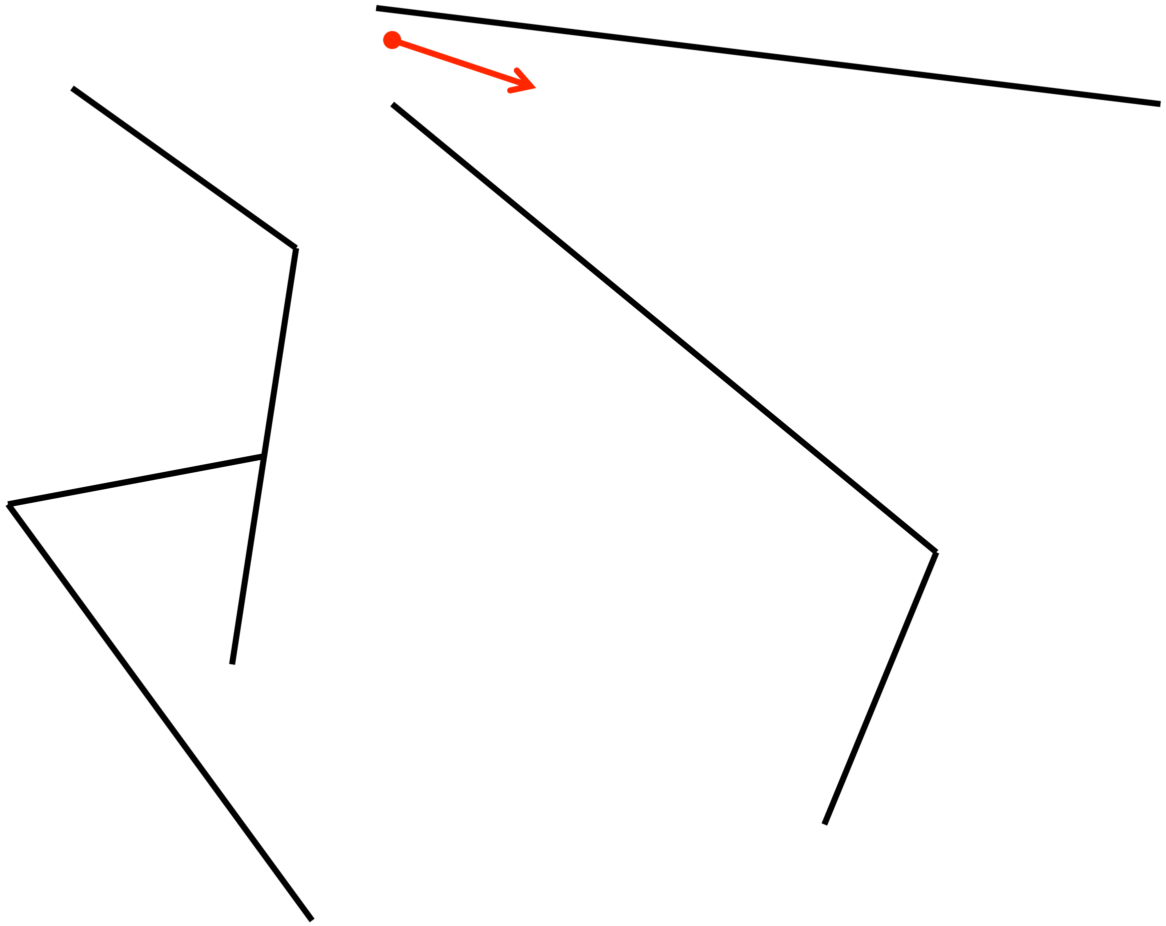


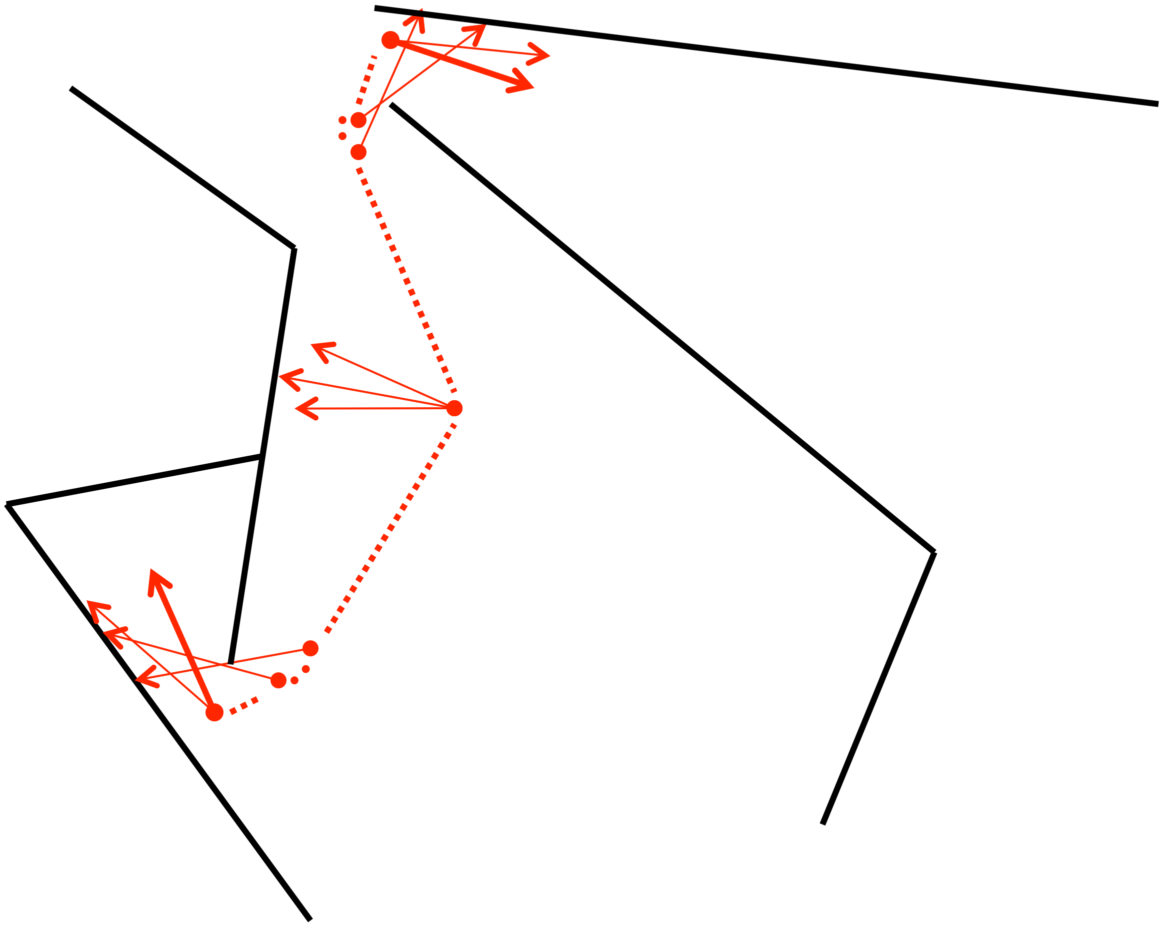




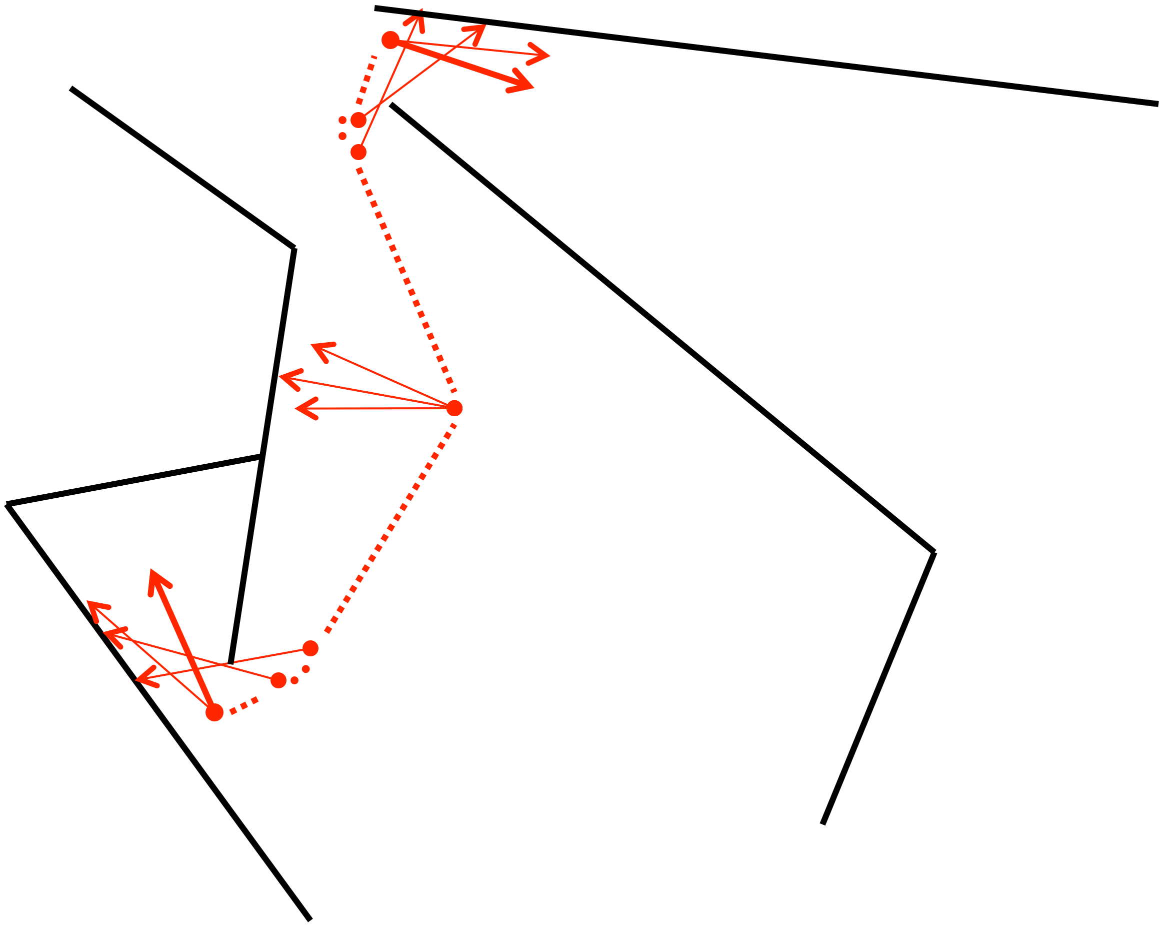




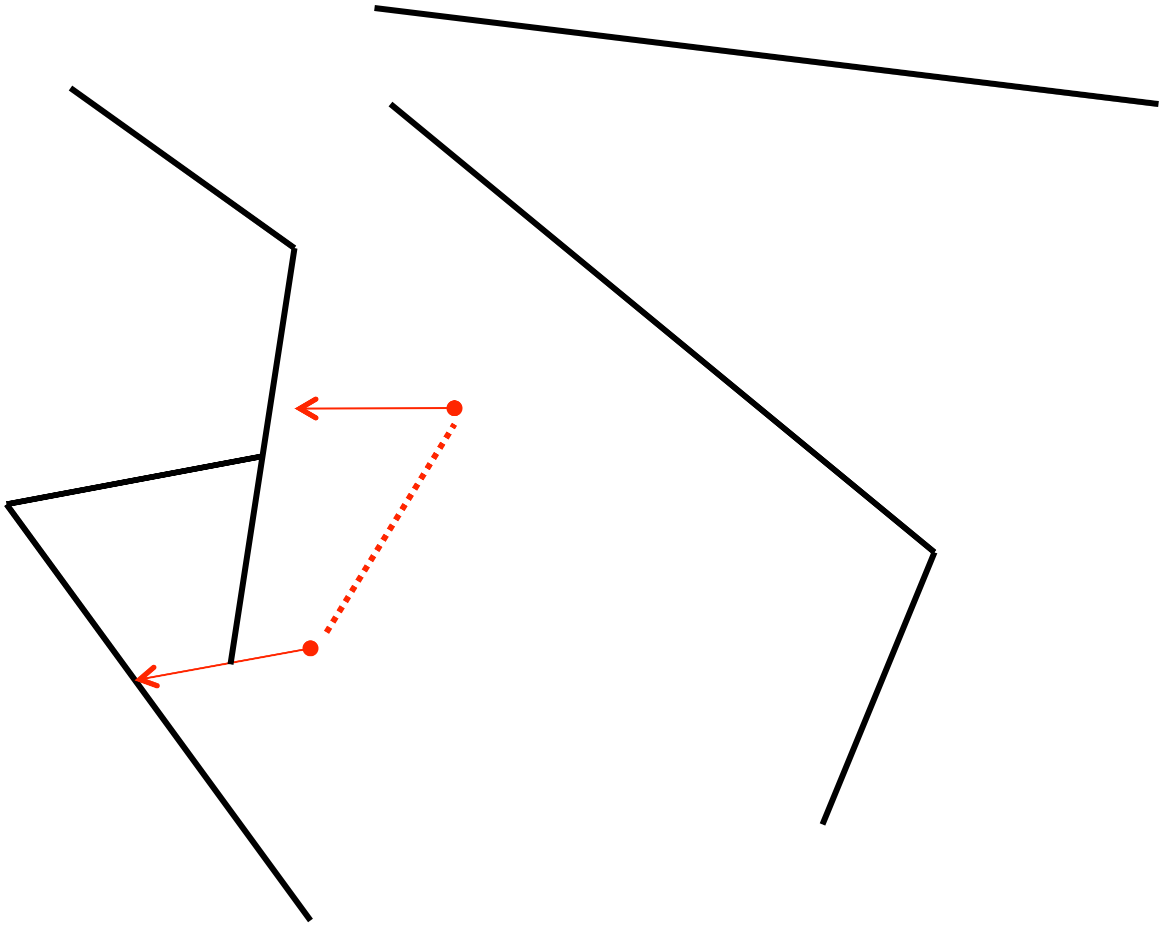




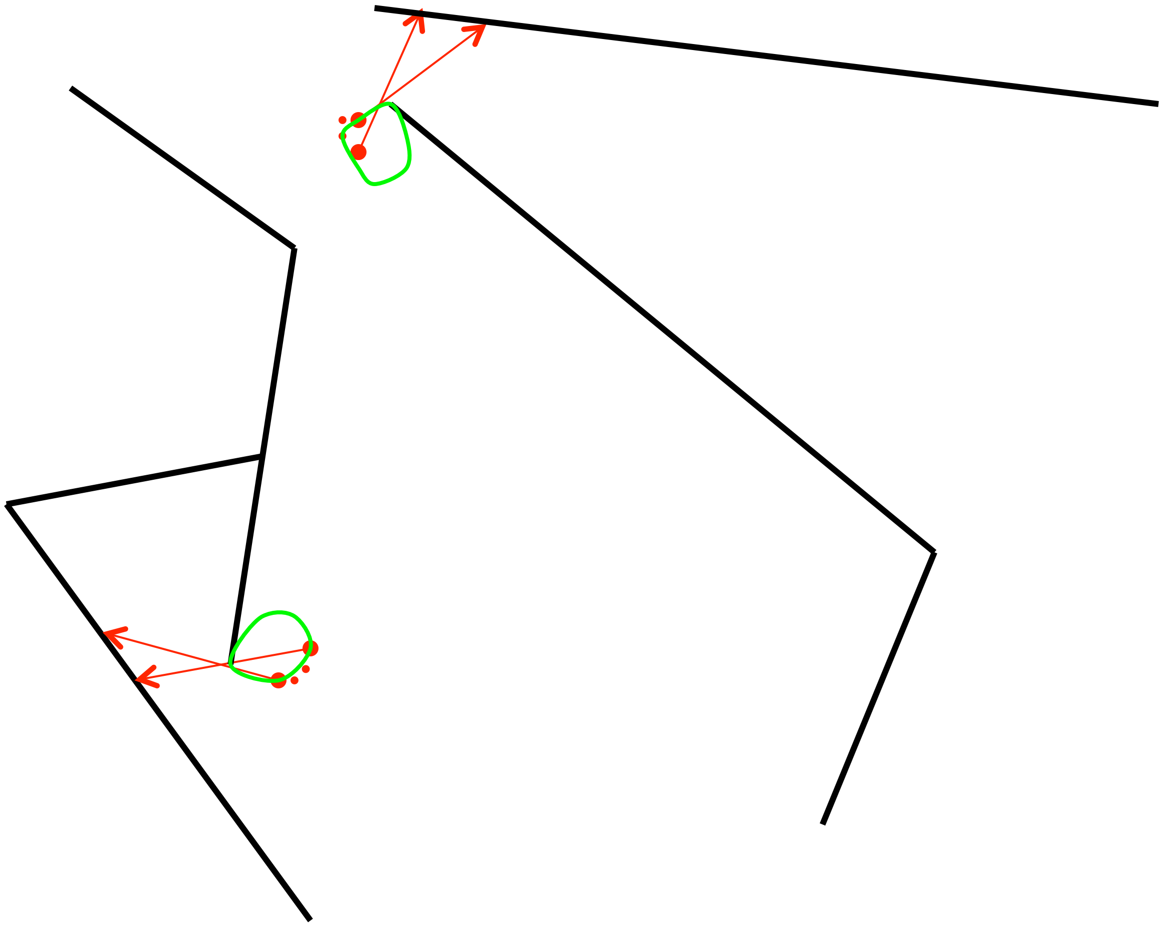
shortest trace of endpoint



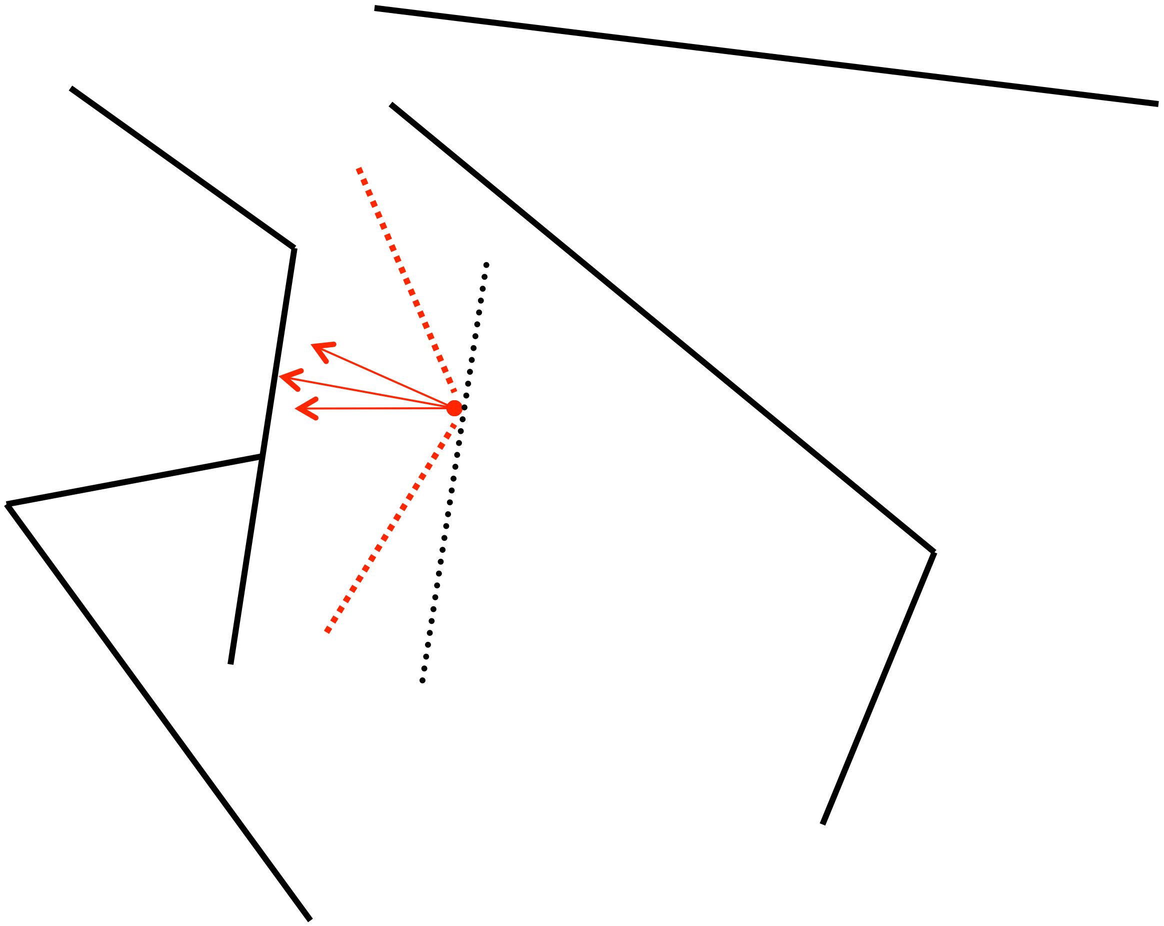
shortest trace of endpoint



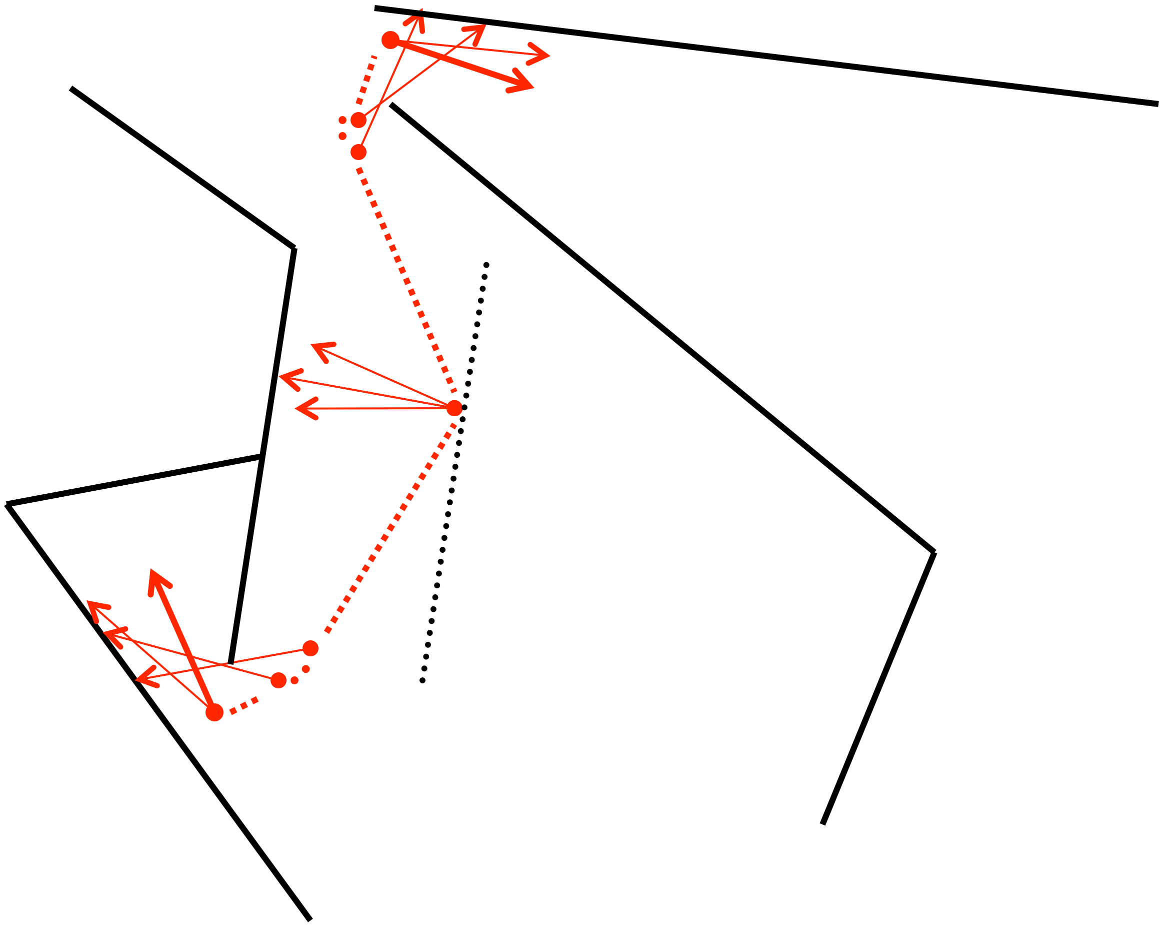
translation



sliding (conchoid curve)



reflection (rotation)



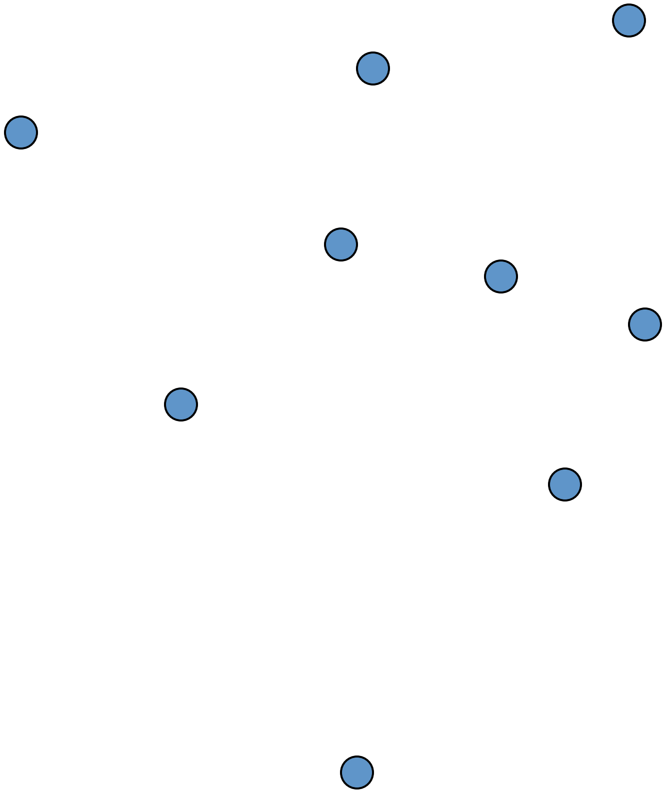
can’t predetermine all
“stopover” points

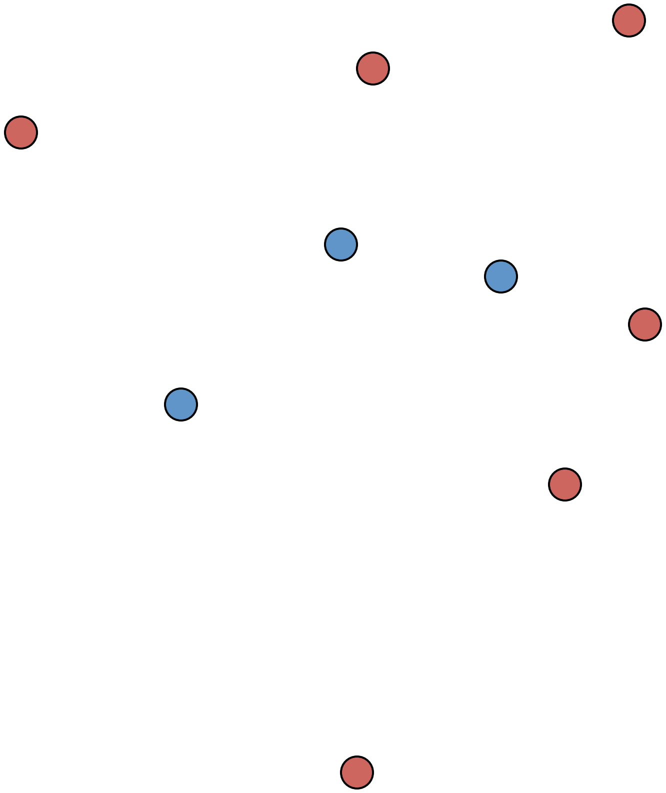
Course Topics

- Geometric optimization
  - finding extreme points
  - low dimensional linear programming
  - facility location
- Geometric search
  - Planar point location
  - Data structures that facilitate geometric search
- Intersection detection
  - Segments, polygons, polyhedra…

Course Topics

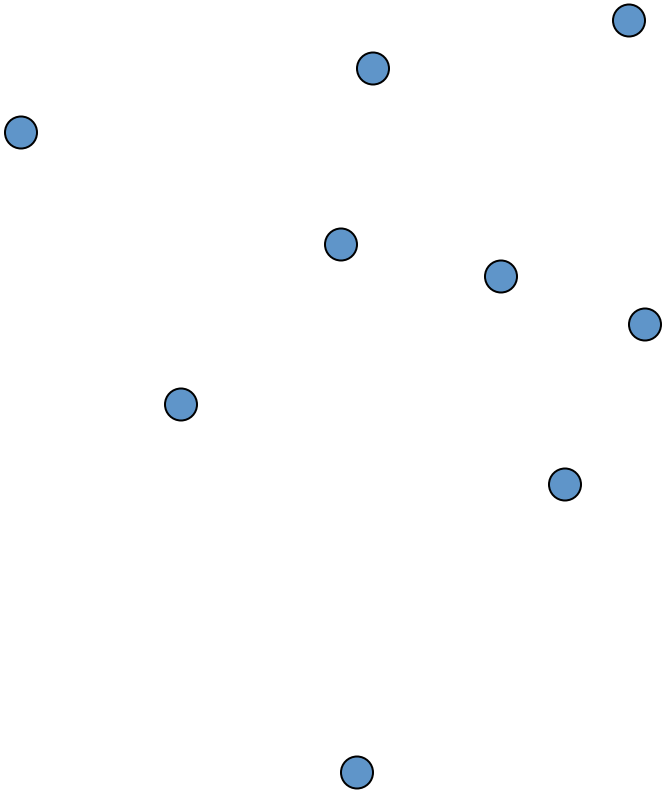
- Geometric optimization
    - finding extreme points
    - low dimensional linear programming
    - facility location
- Geometric search
    - Planar point location
    - Data structures that facilitate geometric search
- Intersection detection
    - Segments, polygons, polyhedra…

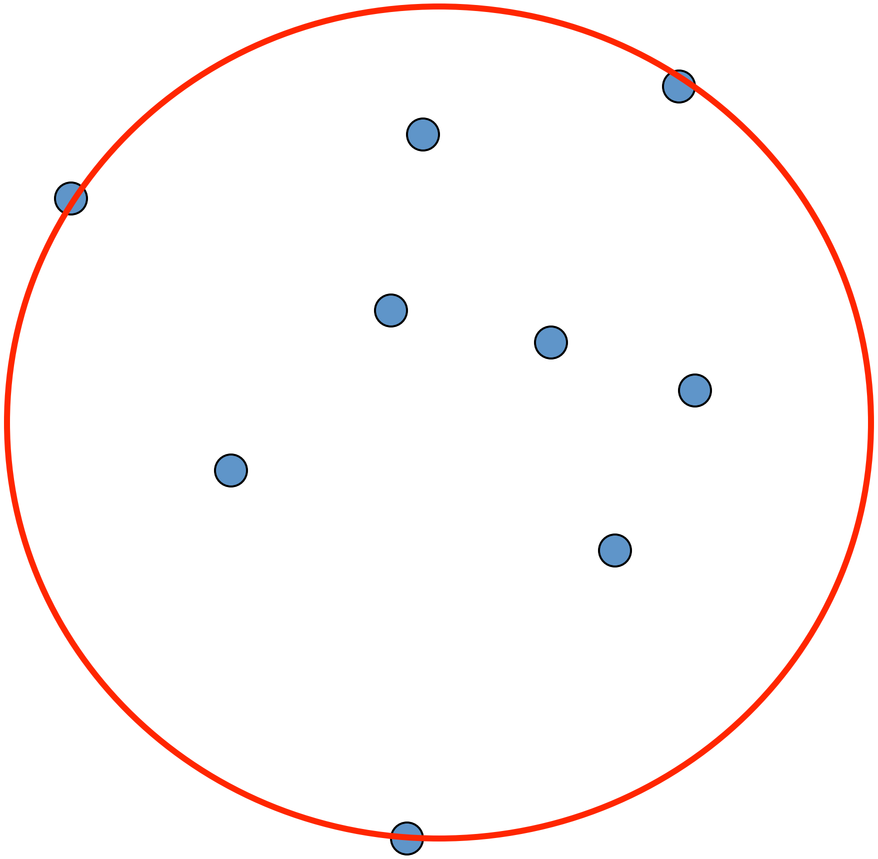


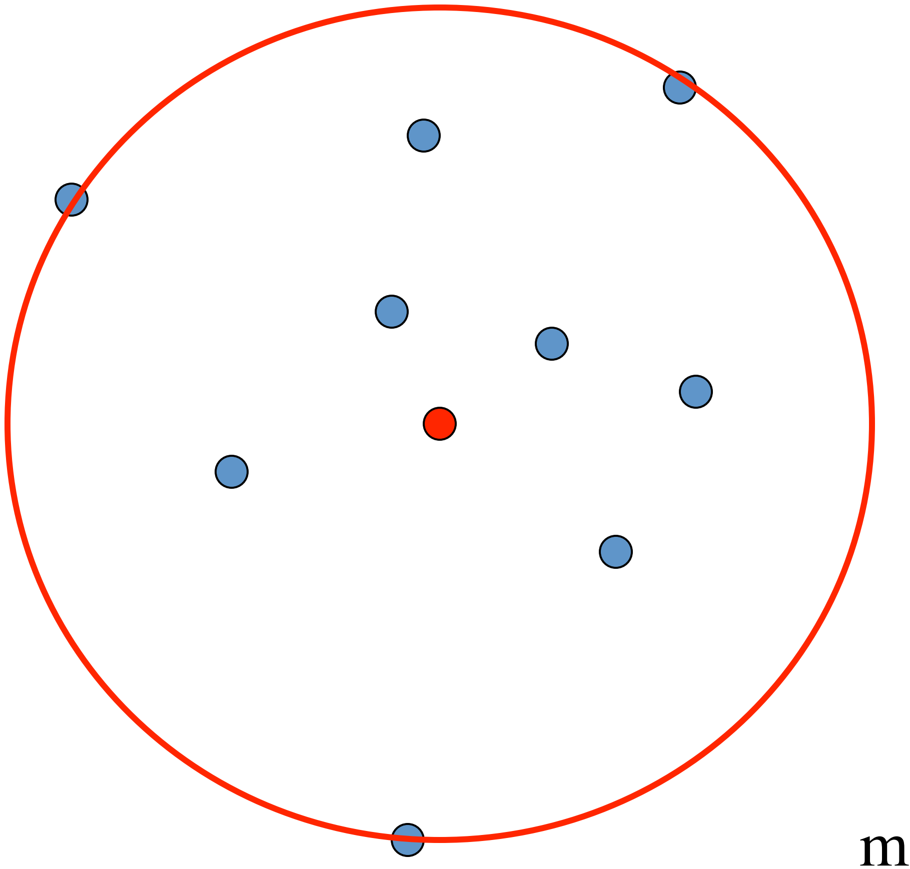


Course Topics

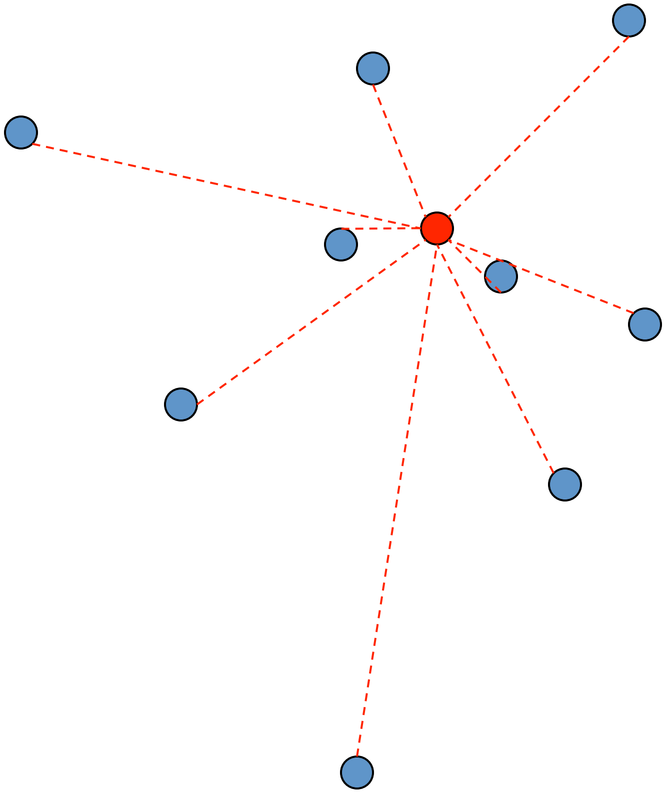
- Geometric optimization
    - finding extreme points
    - low dimensional linear programming
    - facility location
- Geometric search
    - Planar point location
    - Data structures that facilitate geometric search
- Intersection detection
    - Segments, polygons, polyhedra…







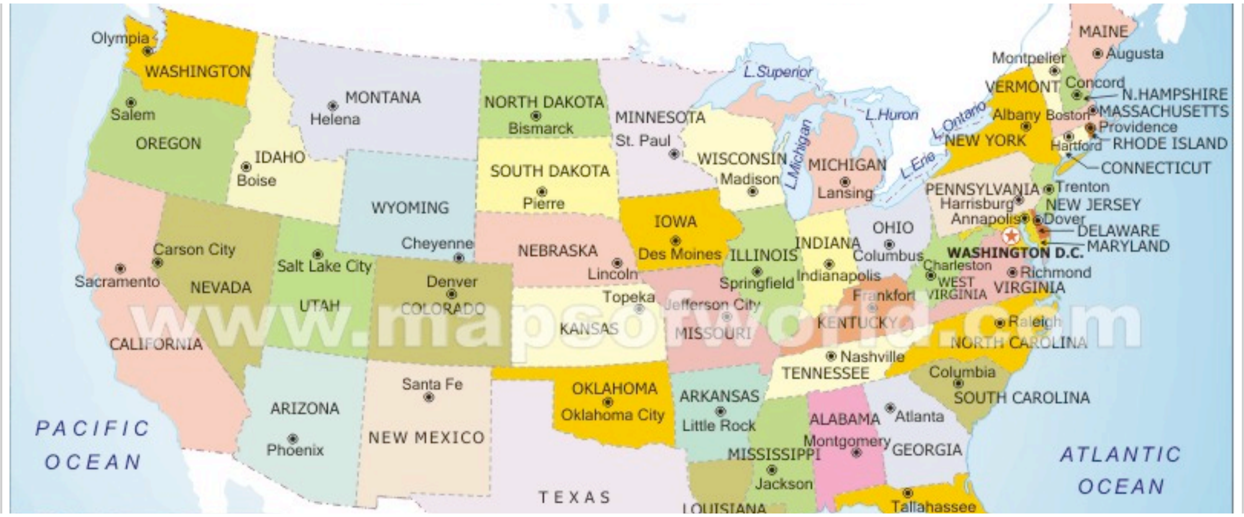
minimize max
distance

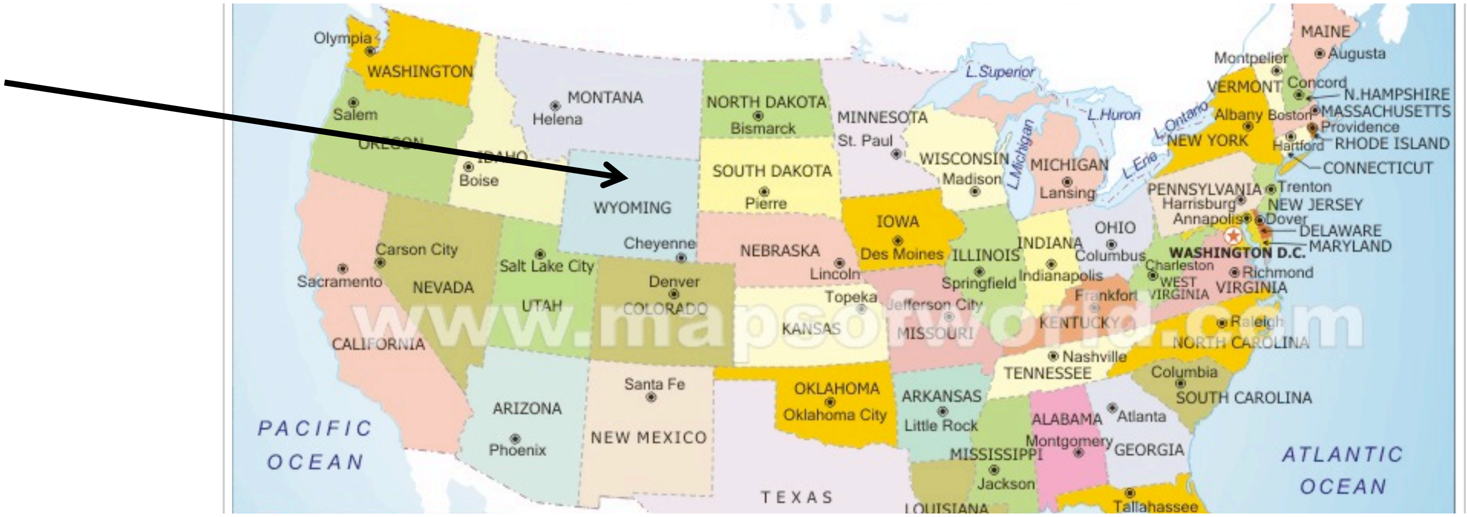


minimize sum of distances

Course Topics

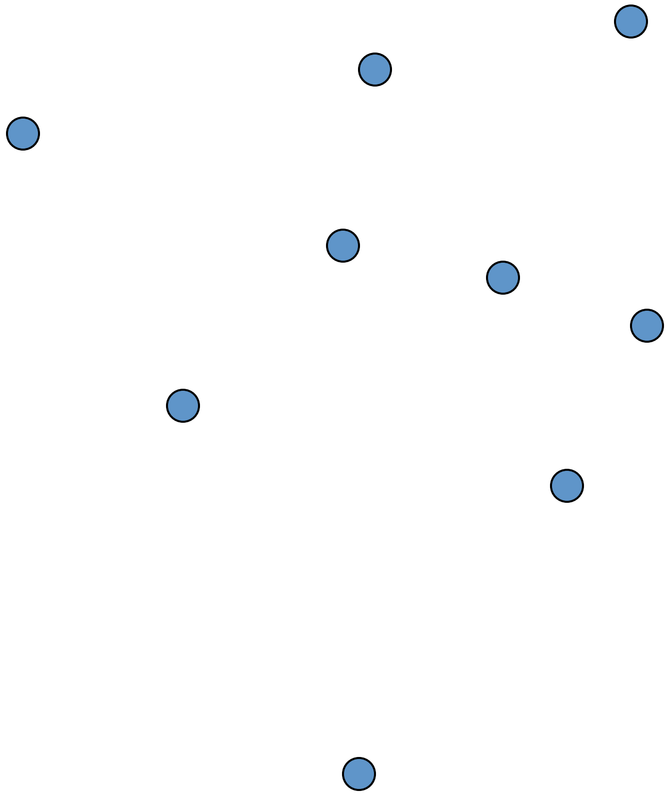
- Geometric optimization
    - finding extreme points
    - low dimensional linear programming
    - facility location
- Geometric search
    - Planar point location
    - Data structures that facilitate geometric search
- Intersection detection
    - Segments, polygons, polyhedra…

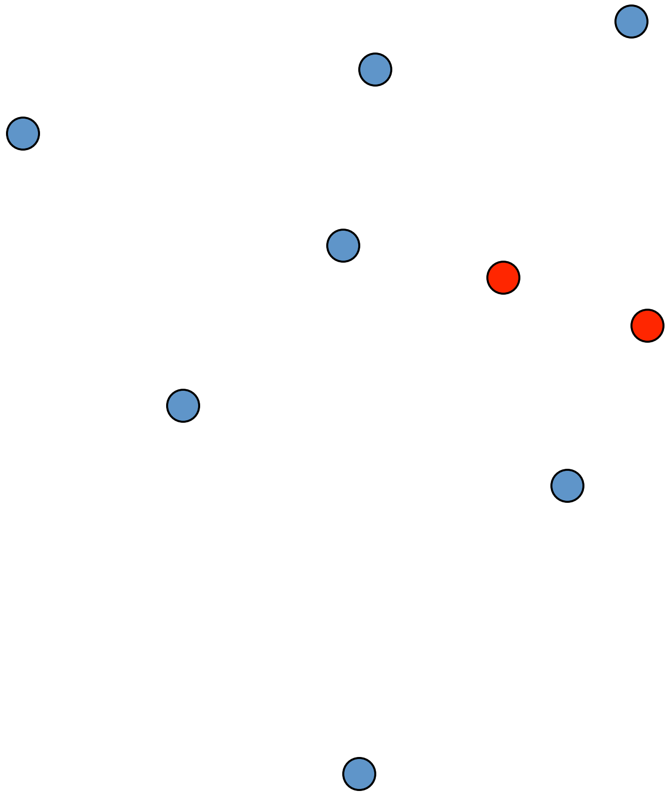

Olympia
WASHINGTON
Salem
OREGON
MONTANA
Helena
IDAHO
Boise
NORTH DAKOTA
Bismarck
MINNESOTA
St. Paul
SOUTH DAKOTA
Pierre
WYOMING
Cheyenne
L.Superior
L.Michigan
L.Huron
L.Ontario
L.Erie
WISCONSIN
Madison
MICHIGAN
Lansing
MAINE
Augusta
Montpelier
VERMONT
Concord
N.HAMPSHIRE
Albany
Boston
MASSACHUSETTS
Providence
NEW YORK
Hartford
RHODE ISLAND
CONNECTICUT
PENNSYLVANIA
Trenton
Harrisburg
NEW JERSEY
Annapolis
Dover
DELAWARE
MARYLAND
WASHINGTON D.C.
IOWA
Des Moines
NEBRASKA
Lincoln
ILLINOIS
Springfield
INDIANA
Indianapolis
OHIO
Columbus
Charleston
WEST VIRGINIA
Richmond
VIRGINIA
Carson City
Sacramento
NEVADA
Salt Lake City
UTAH
Denver
COLORADO
Topeka
KANSAS
Jefferson City
MISSOURI
Frankfort
KENTUCKY
Raleigh
NORTH CAROLINA
CALIFORNIA
www.mapsofworld.com
Nashville
TENNESSEE
Columbia
SOUTH CAROLINA
Santa Fe
ARIZONA
Phoenix
NEW MEXICO
OKLAHOMA
Oklahoma City
ARKANSAS
Little Rock
ALABAMA
Montgomery
Atlanta
GEORGIA
MISSISSIPPI
Jackson
PACIFIC OCEAN
ATLANTIC OCEAN
TEXAS
LOUISIANA
Tallahassee


Olympia
WASHINGTON
Salem
OREGON
IDAHO
Boise
MONTANA
Helena
WYOMING
Cheyenne
NORTH DAKOTA
Bismarck
SOUTH DAKOTA
Pierre
MINNESOTA
St. Paul
WISCONSIN
Madison
MICHIGAN
Lansing
L.Superior
L.Michigan
L.Huron
L.Erie
L.Ontario
IOWA
Des Moines
NEBRASKA
Lincoln
ILLINOIS
Springfield
INDIANA
Indianapolis
OHIO
Columbus
Sacramento
CALIFORNIA
Carson City
NEVADA
Salt Lake City
UTAH
Denver
COLORADO
Topeka
KANSAS
Jefferson City
MISSOURI
Frankfort
KENTUCKY
Nashville
TENNESSEE
ARIZONA
Phoenix
Santa Fe
NEW MEXICO
OKLAHOMA
Oklahoma City
TEXAS
ARKANSAS
Little Rock
MISSISSIPPI
Jackson
LOUISIANA
ALABAMA
Montgomery
Atlanta
GEORGIA
Tallahassee
Columbia
SOUTH CAROLINA
Raleigh
NORTH CAROLINA
Richmond
VIRGINIA
Charleston
WEST VIRGINIA
WASHINGTON D.C.
MARYLAND
DELAWARE
Dover
Annapolis
Harrisburg
PENNSYLVANIA
Trenton
NEW JERSEY
Albany
NEW YORK
Hartford
CONNECTICUT
RHODE ISLAND
Providence
Boston
MASSACHUSETTS
Concord
N.HAMPSHIRE
Montpelier
VERMONT
MAINE
Augusta
PACIFIC OCEAN
ATLANTIC OCEAN
www.mapsofworld.com

Topics (cont.)

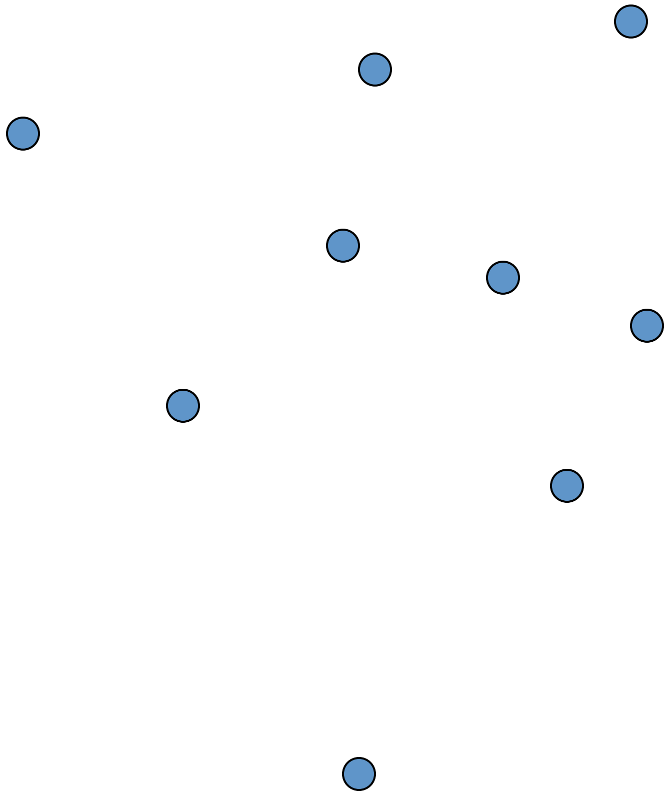
- Proximity problems
    - Nearest neighbours
    - Voronoi diagrams
- Spatial decomposition
    - Triangulations, mesh generation
- Combinatorial geometry
    - Line arrangements
    - Ham sandwiches

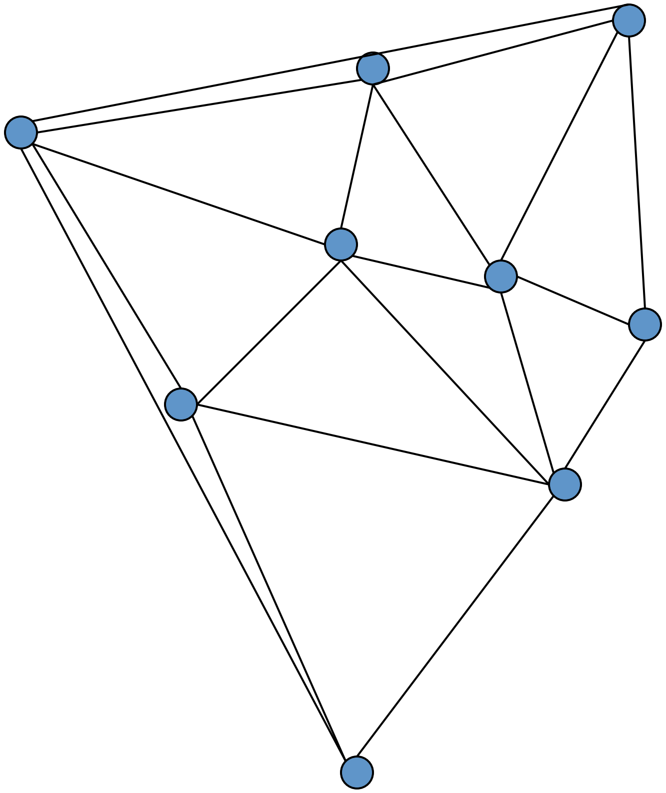




Topics (cont.)

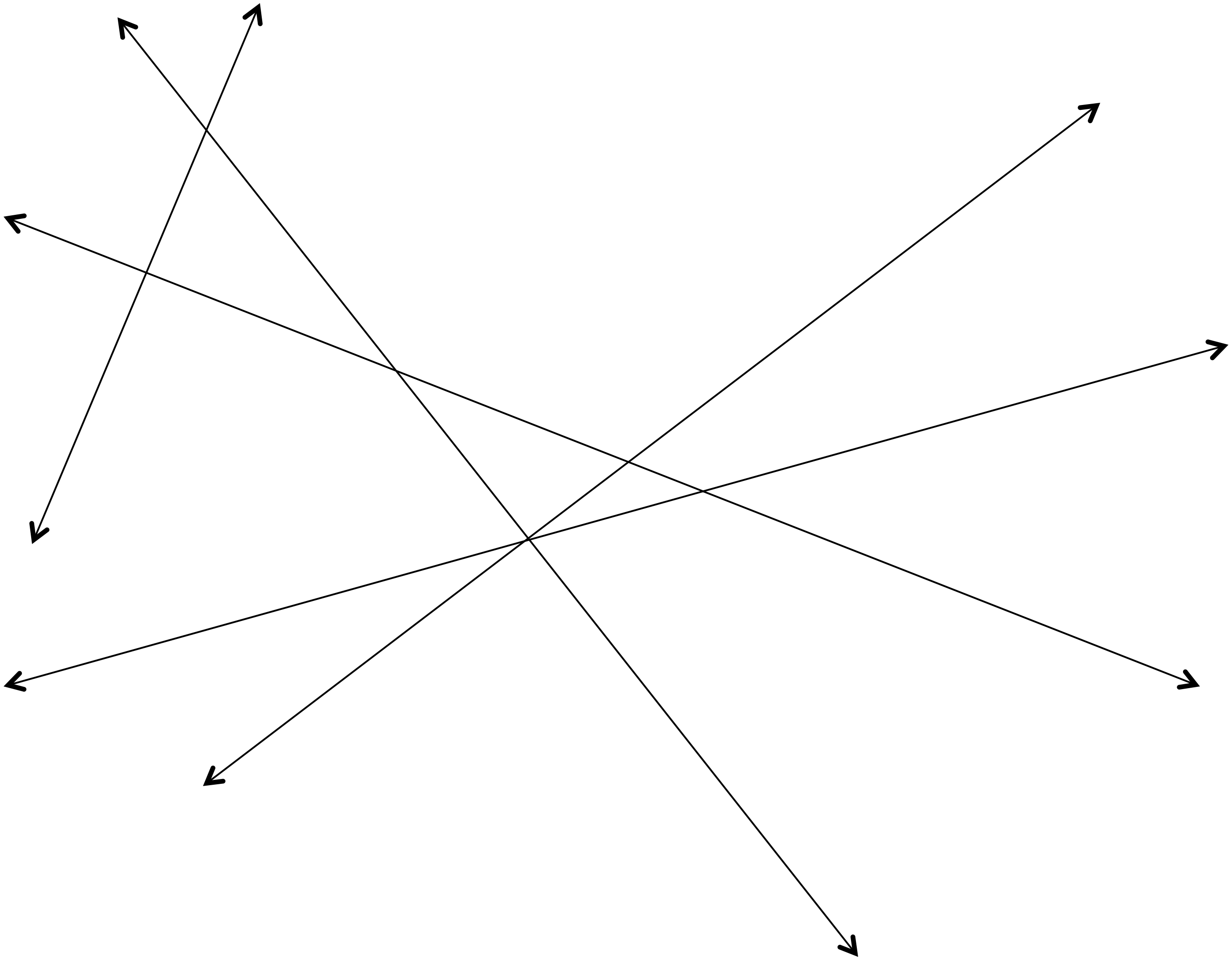
- Proximity problems
  - Nearest neighbors
  - Voronoi diagrams
- Spatial decomposition
  - Triangulations, mesh generation
- Combinatorial geometry
  - Line arrangements
  - Ham sandwiches

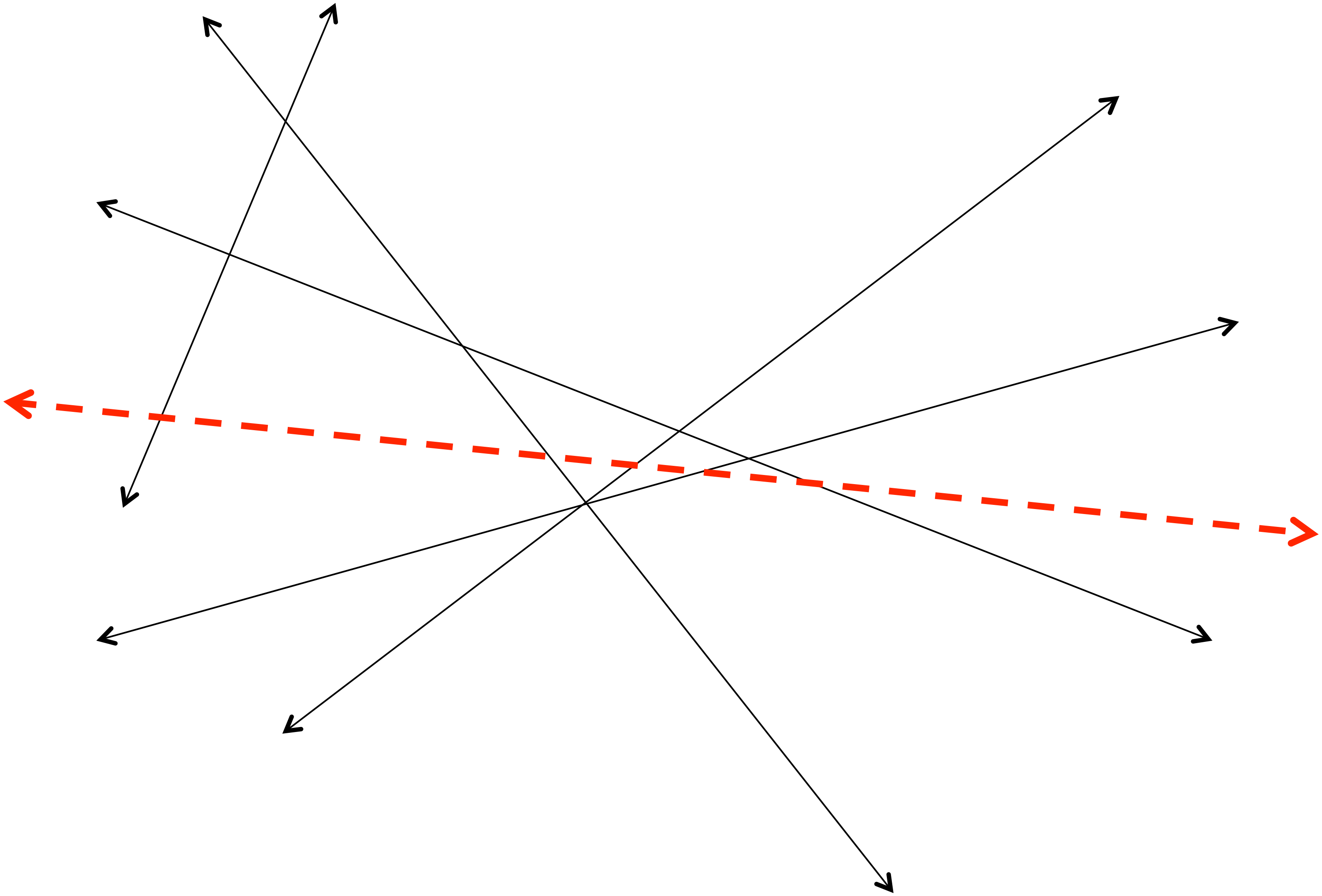




Topics (cont.)

- Proximity problems
    - Nearest neighbors
    - Voronoi diagrams
- Spatial decomposition
    - Triangulations, mesh generation
- Combinatorial geometry
    - Line arrangements
    - Ham sandwiches





Topics (cont.)

- Complexity and approximation techniques
    - Output-size sensitivity
    - Lower bounds
    - Random sampling
- Other models
    - Parallel, kinetic & streaming algorithms
    - Robustness and uncertainty

More Topics (cont.)

- “Nice” layouts of trees, wires and planar graphs
- Contact & intersection representation of graphs
- Proximity graphs
- Visibility graphs
- Geographic graphs
- Force-directed graph drawing

Graph drawing

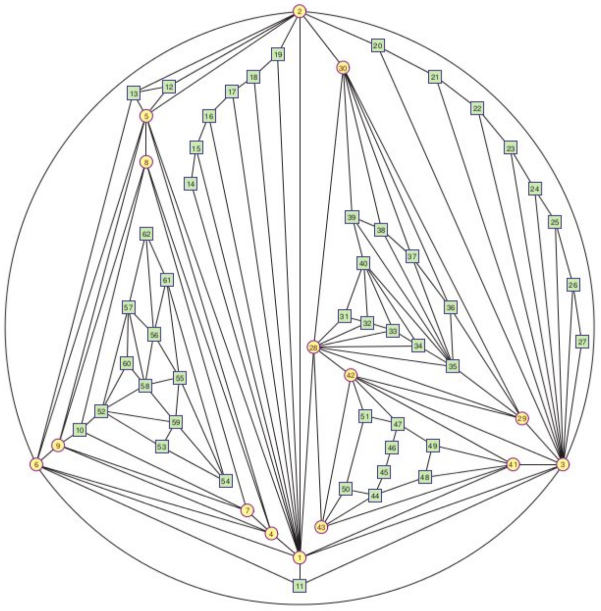
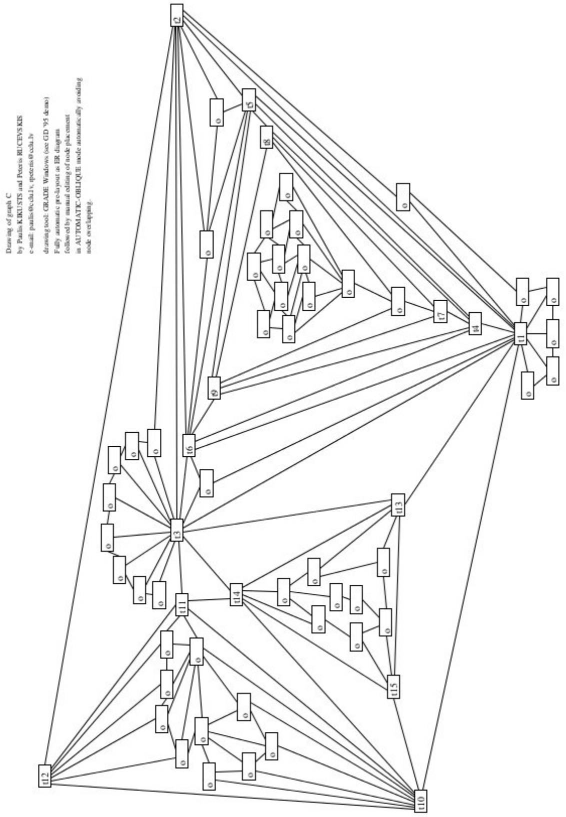
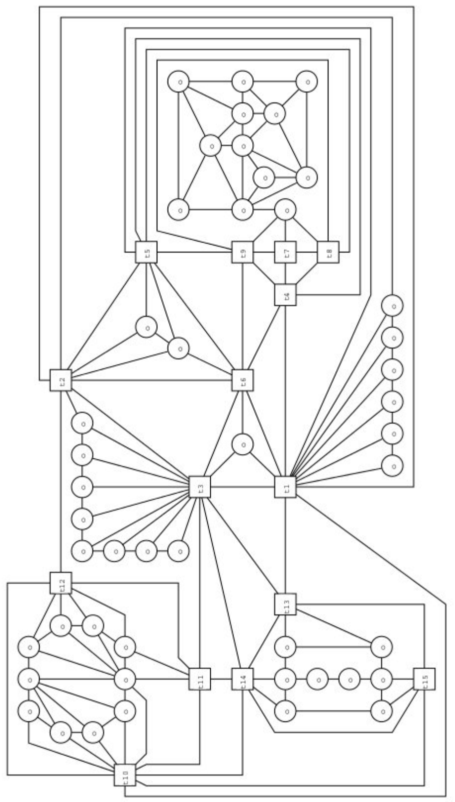
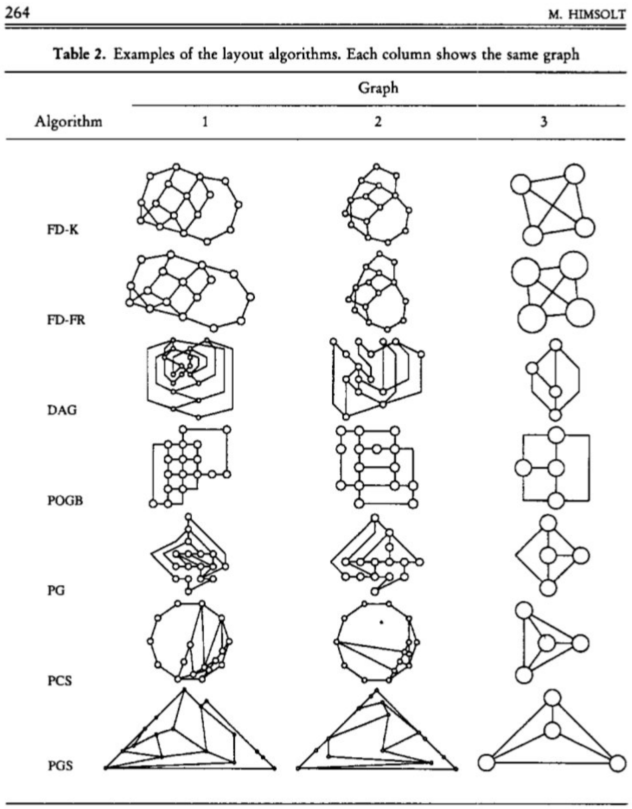


Figure 8: Joint winner, Graph C.


Drawing of graph C
by Paulis KIKUSTS and Peteris RUCEVSKIS
e-mail: paulis@cclu.lv, rpeteris@cclu.lv
drawing tool: GRADE Windows (see GD '95 demo)
Fully automatic pre-layout as ER diagram
followed by manual editing of node placement
in AUTOMATIC-OBLIQUE mode automatically avoiding
node overlapping.



Planar graph drawing

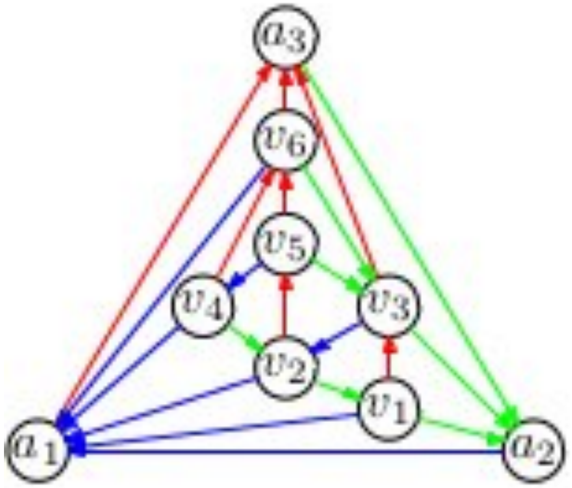


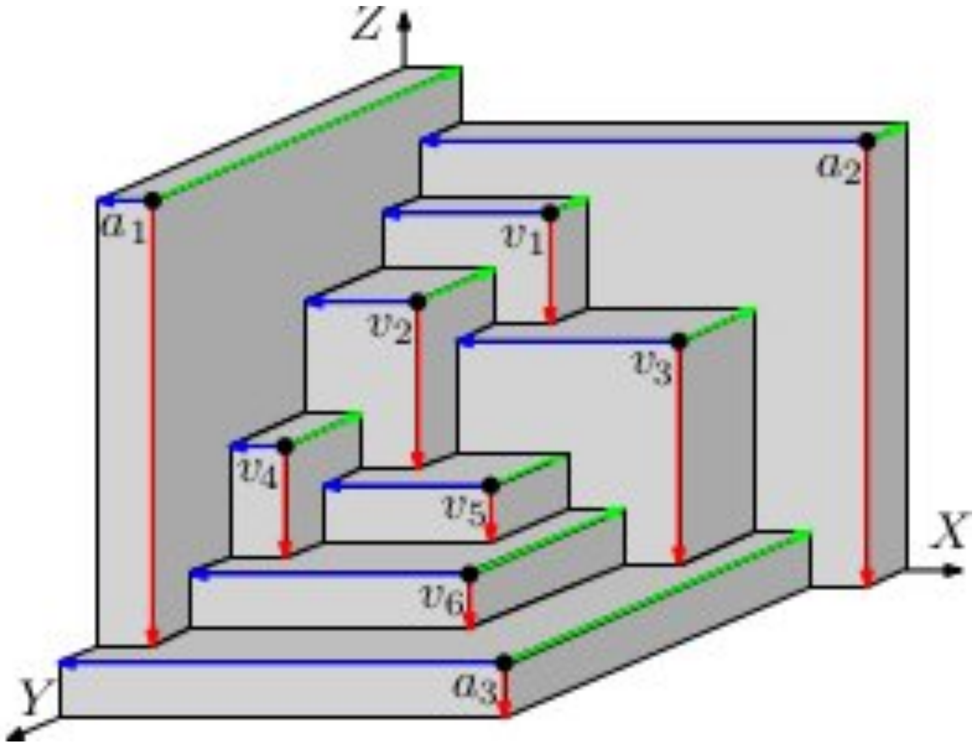
264 M. HIMSOLT

Table 2. Examples of the layout algorithms. Each column shows the same graph

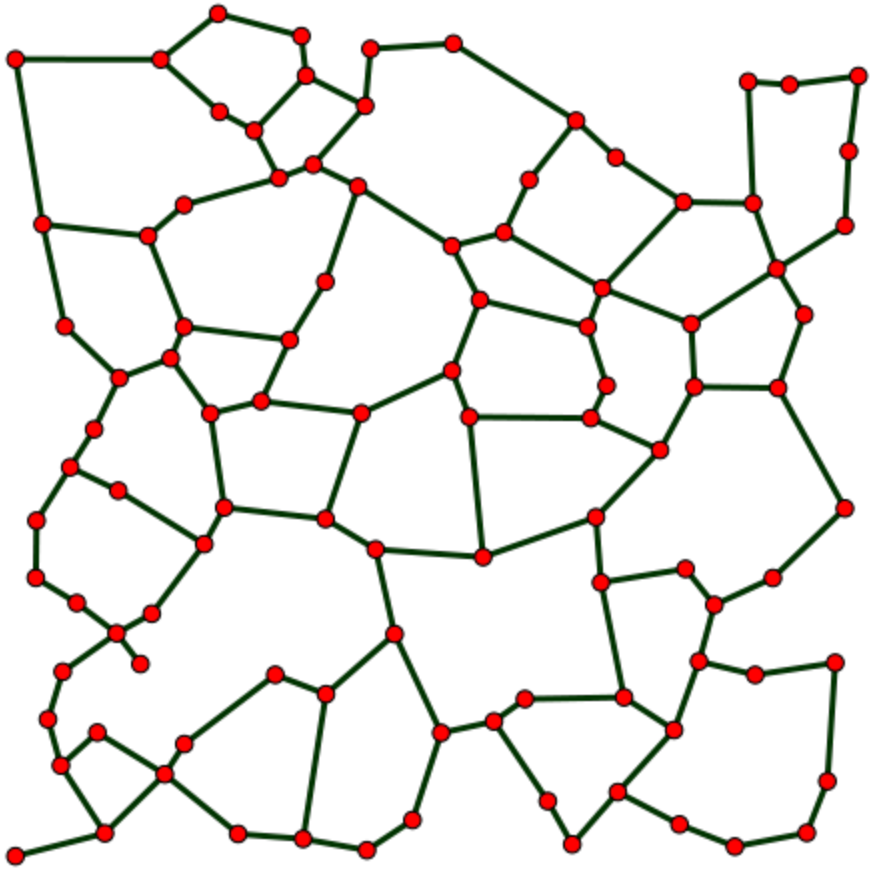
	Graph		
Algorithm	1	2	3
FD-K			
FD-FR			
DAG			
POGB			
PG			
PCS			
PGS			

Contact representation

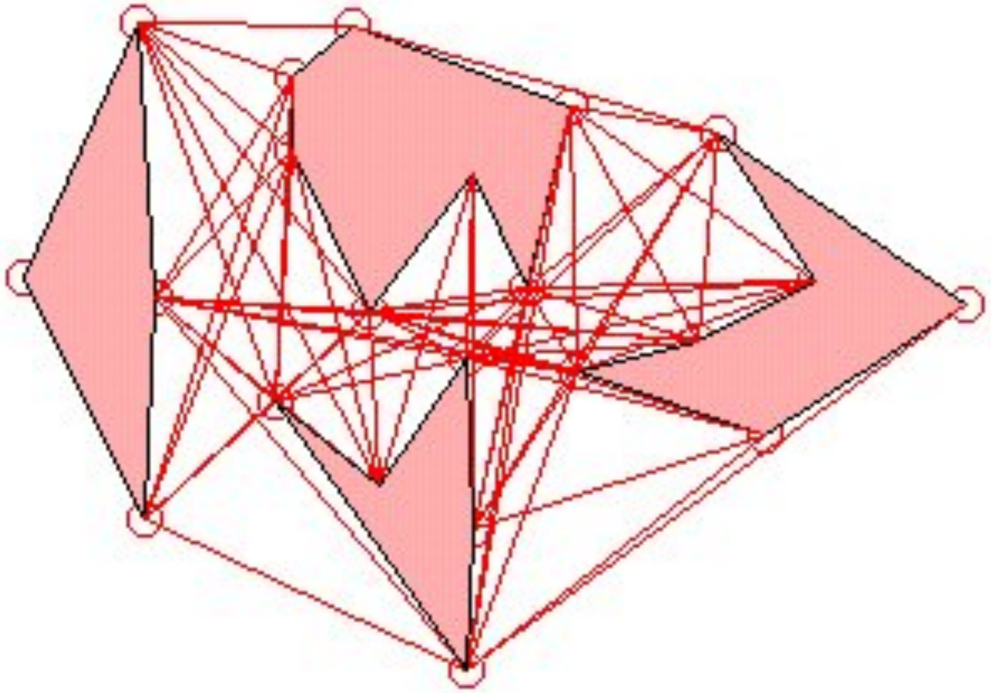




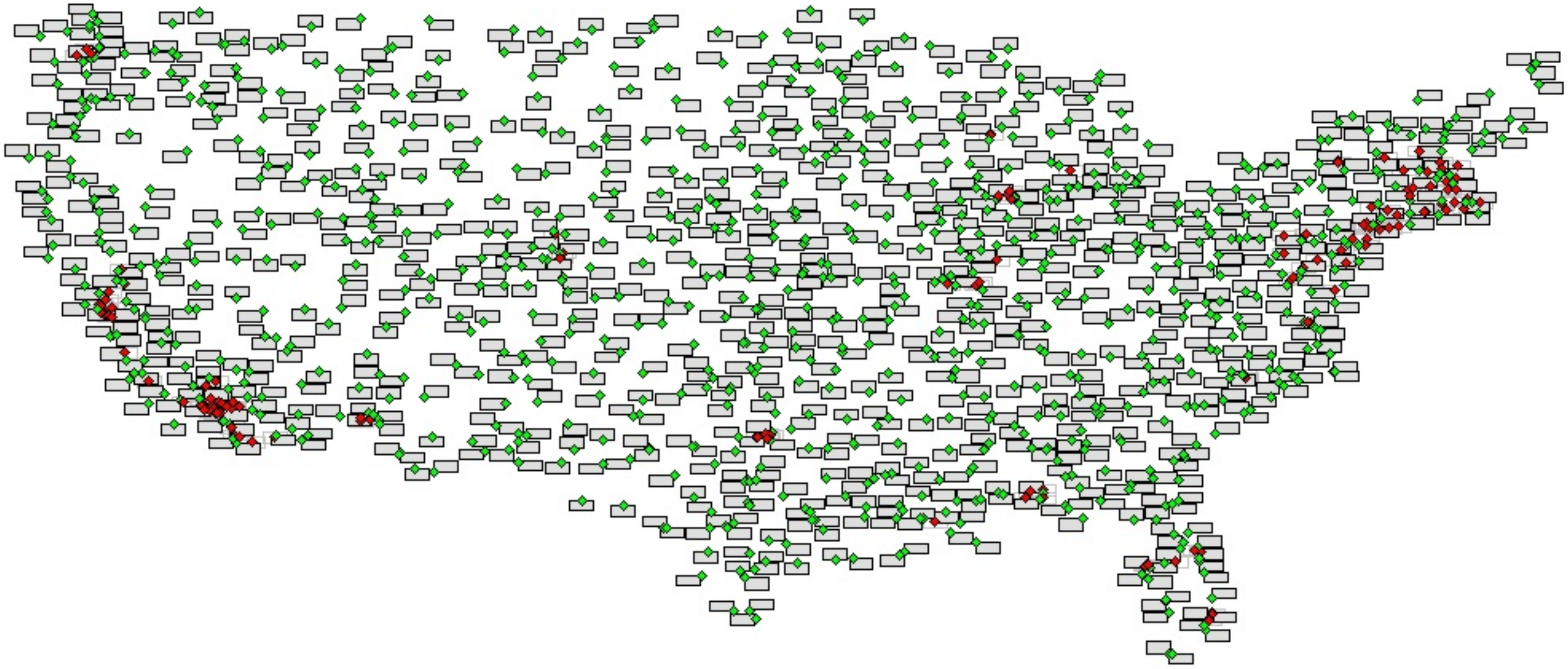
Relative Neighborhood Graph



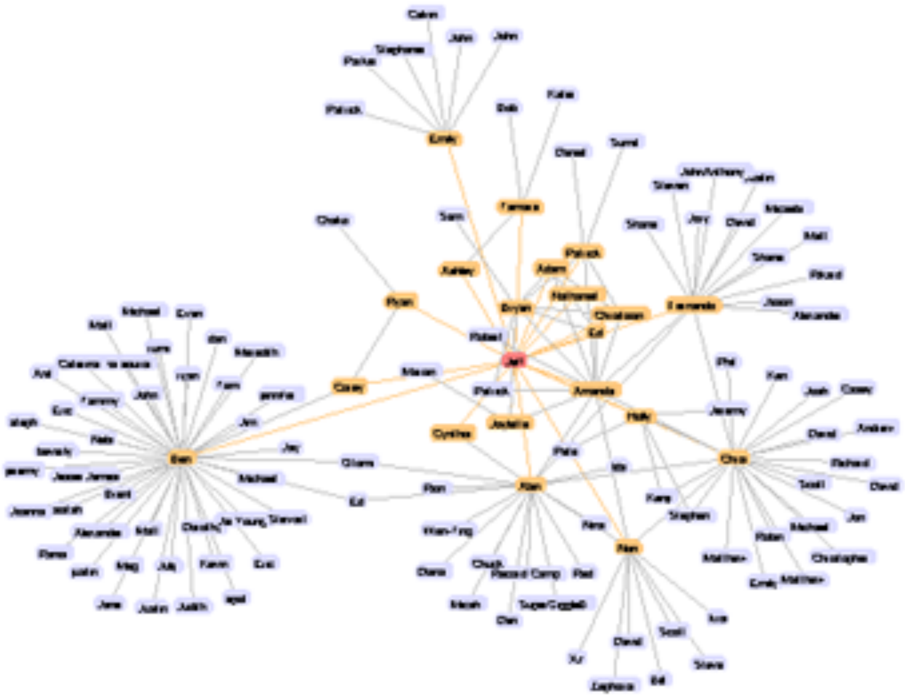
Visibility Graph



Map labelling



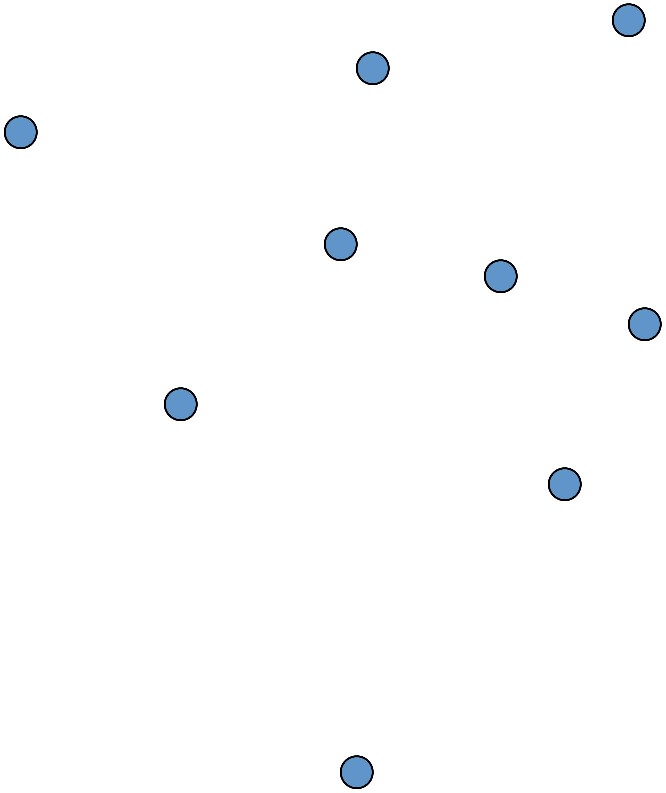
Force-directed layout



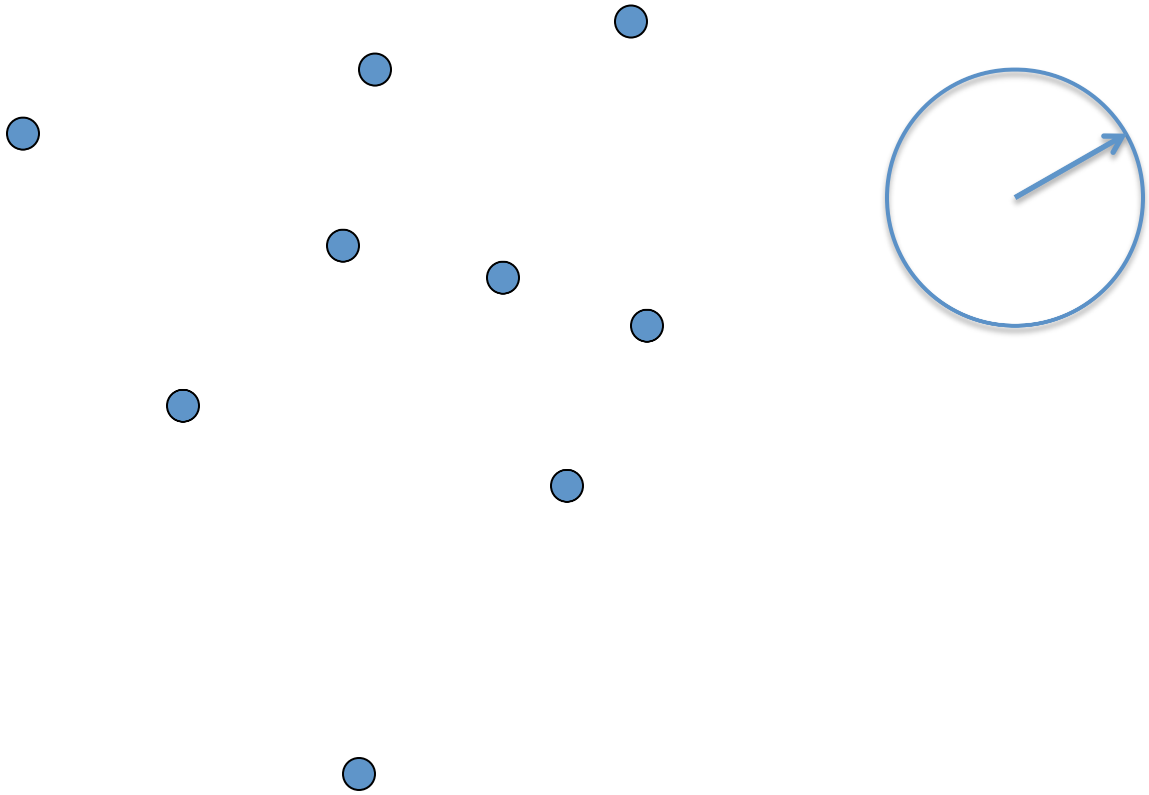
Topics (cont.)

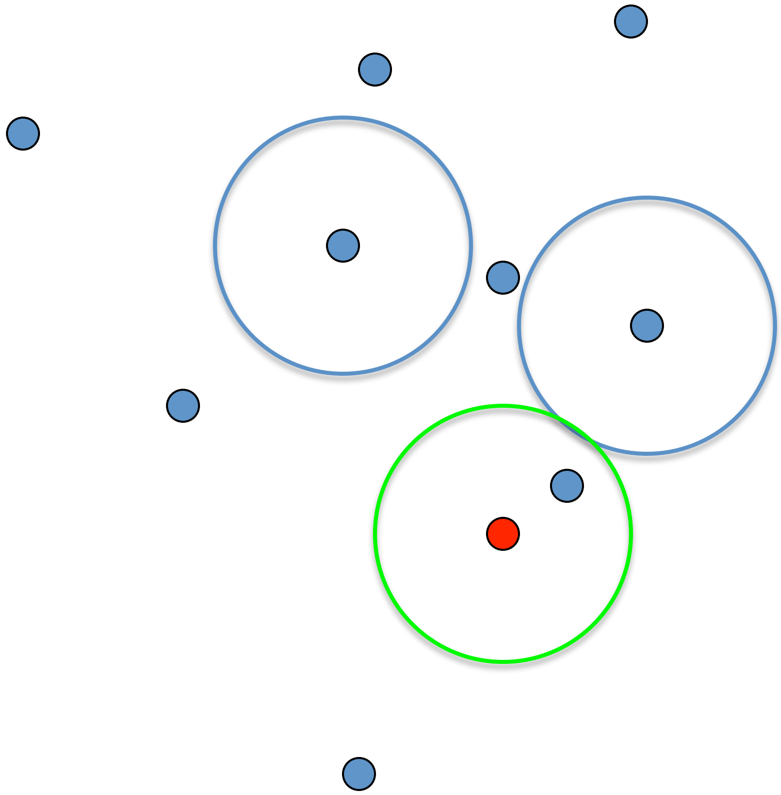
- Applications
  - Geographic information systems
  - Guarding & sensor networks
  - Motion planning
  - Manufacturing
  - …

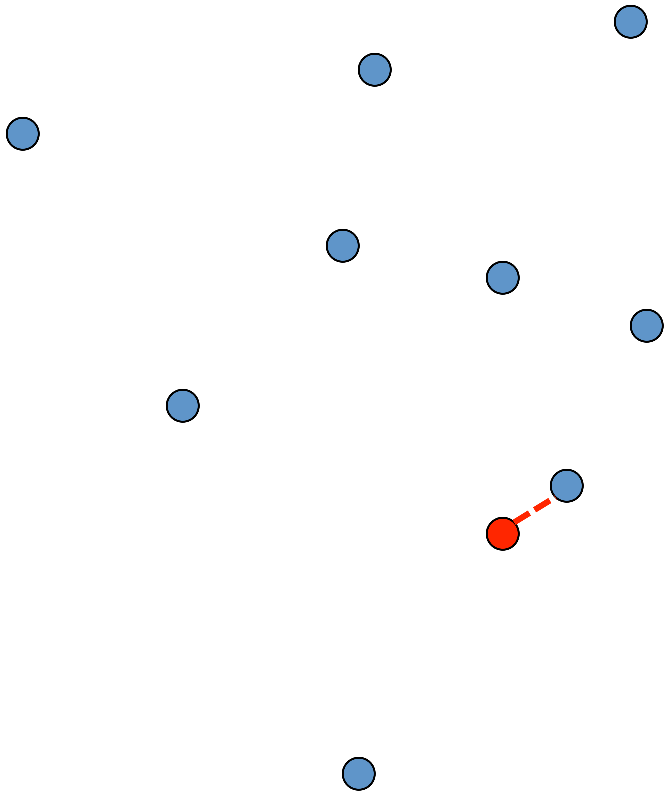
Case study: finding all near neighbours



Case study: finding all near neighbours







Reading

MountNotes Chapters 1-4,7,8