15.053/8 April 25, 2013

The Traveling Salesman Problem and Heuristics

Quotes of the day

"Problem solving is hunting. It is savage pleasure and we are born to it."

-- Thomas Harris

"An algorithm must be seen to be believed."

-- Donald Knuth

Heuristics

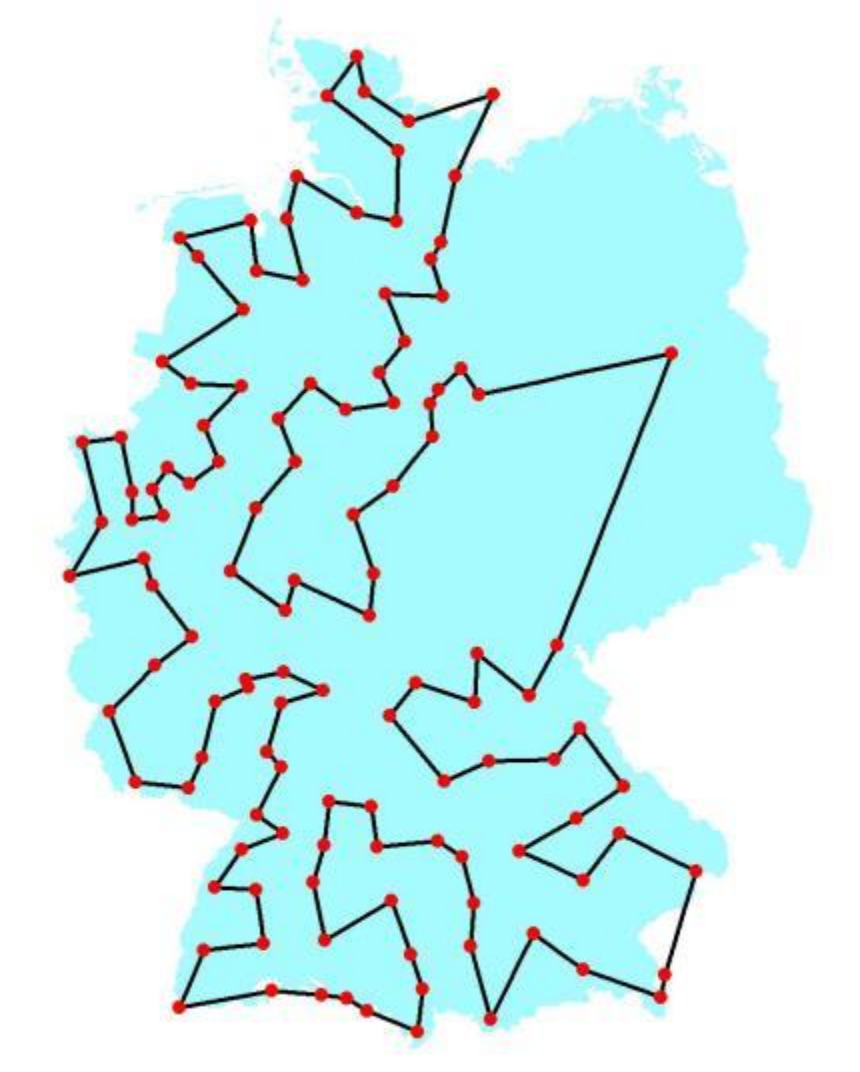
A heuristic is a technique designed for solving a problem more quickly when classic methods are too slow (from Wikipedia).

Today's lecture:

- **Heuristics illustrated on the traveling salesman problem.**
- **Design principles for heuristics**
- **Chances for practice**

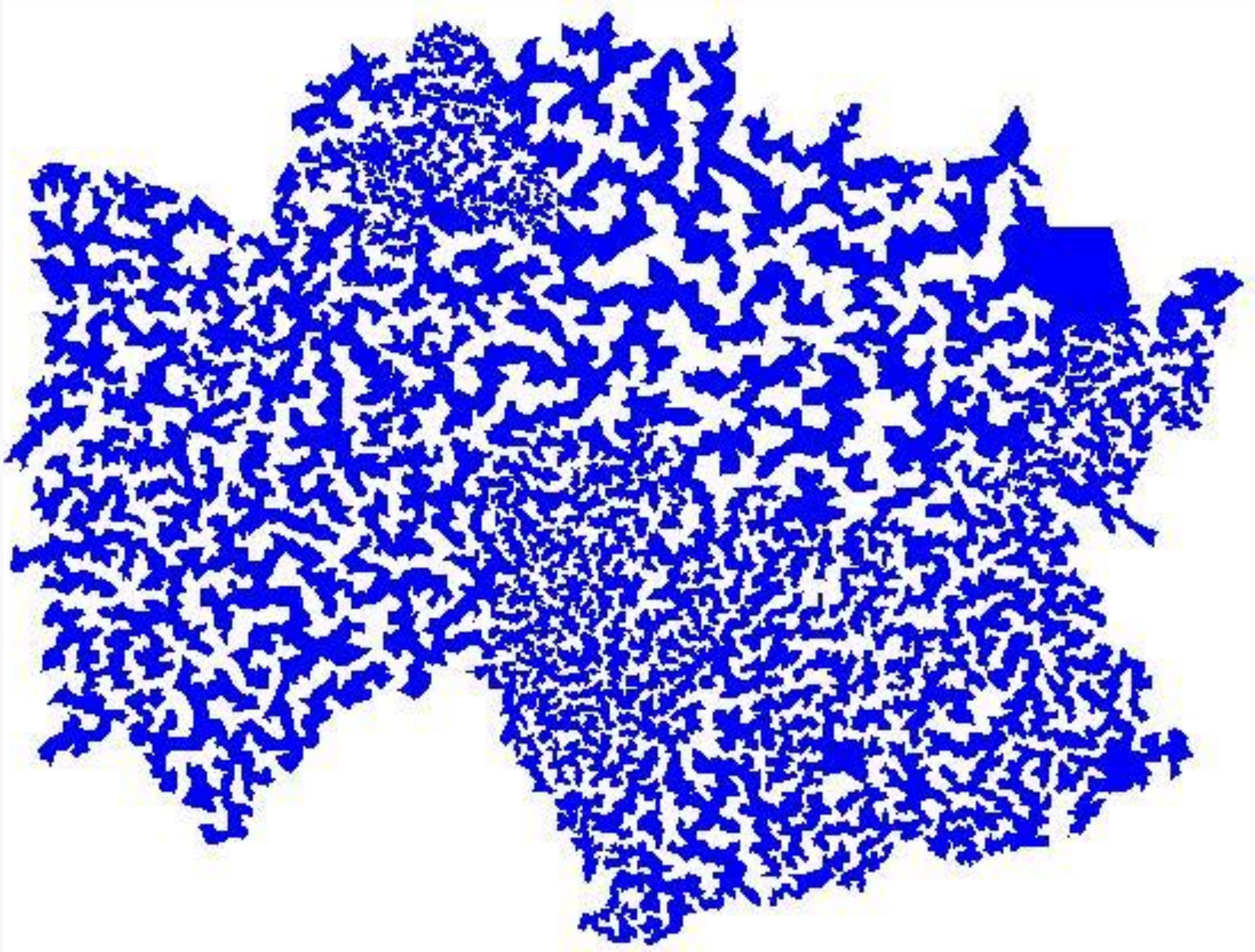
Traveling Salesperson Problem (TSP)

- Find the shortest distance tour passing through each node of the network exactly once.
- c_{ij} = distance from i to j.



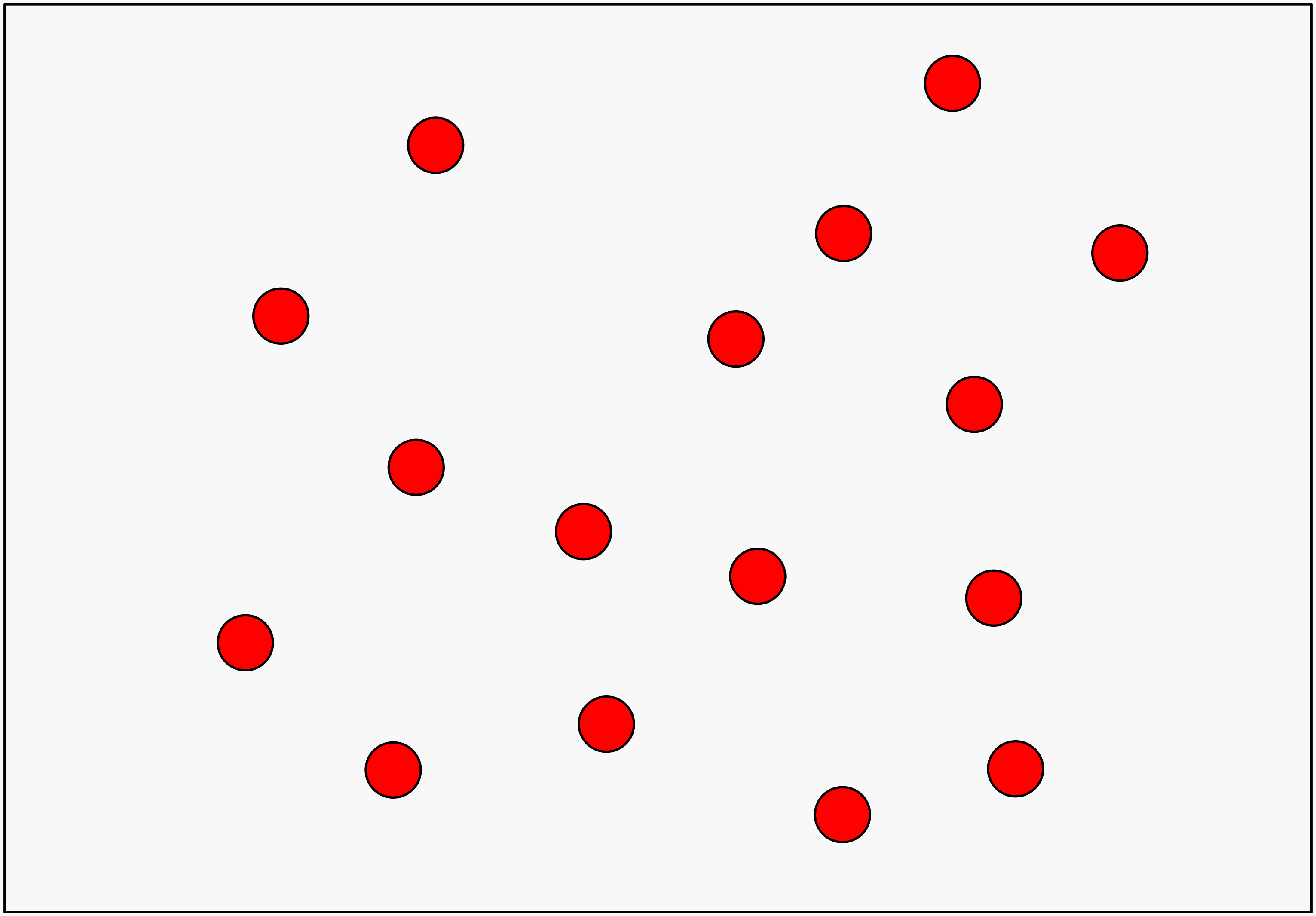
http://www.math.uwaterloo.ca/tsp/

15,112 City Optimal Tour in Germany (rotated)



Courtesy of William Cook. Used with permission.

Exercise: Try to find the best tour.



On solving hard problems

- How did you select your tour?
  - it relied on visualization
  - perhaps you took a holistic view
- How can we develop computer heuristics for solving hard problems?

The TSP is a hard problem

- NP-hard (also NP-complete)

Have you seen NP-hardness or NP-completeness before?

1. Yes.

2. No.

The TSP is a hard problem

- There is no known polynomial time algorithm. Cannot bound the running time as less than n^k for any fixed integer k (say k = 15).

- If there were a polynomial time algorithm, there would be a polynomial time algorithm for every NP-complete problem.

- Question: what does one do with a hard problem?

$100\ n^{15}$ vs. $n!$

Suppose that we could carry out 1 sextillion steps per second (10^{21}).

- n = number of cities. (n-1)! tours.
- compare time for $100\ n^{15}$ steps vs. n! steps.
  - Remark: $100\ n^{15}$ steps is NOT practically efficient.

-

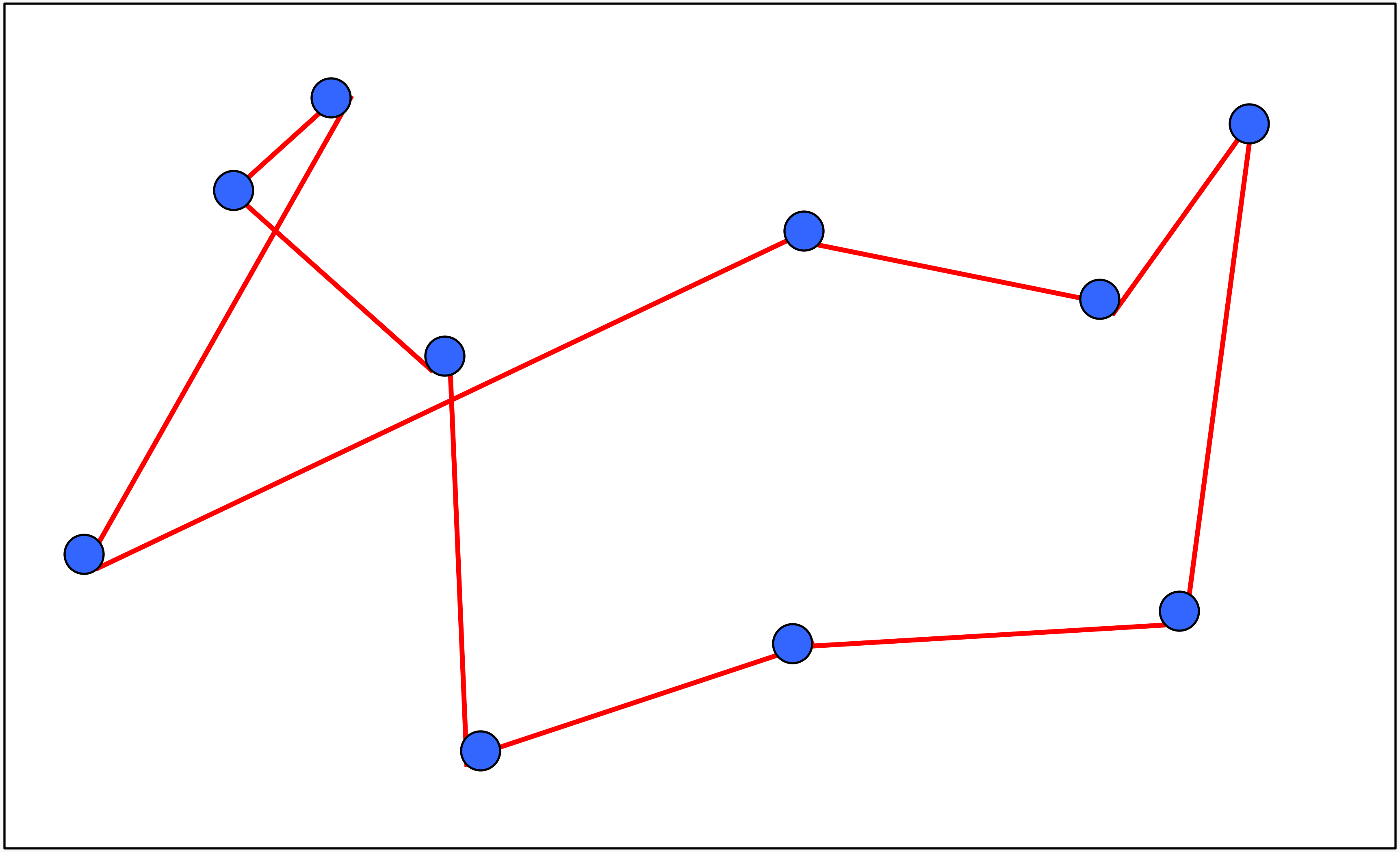
# of cities	$100\ n^{15}$ steps	n! steps
n = 20,	3.27 seconds	2.4 milliseconds
n = 25,	1.5 minutes	4.2 hours
n = 30	24 minutes	8,400 years
n = 35	4 hours	326 billion years

Two types of Heuristics

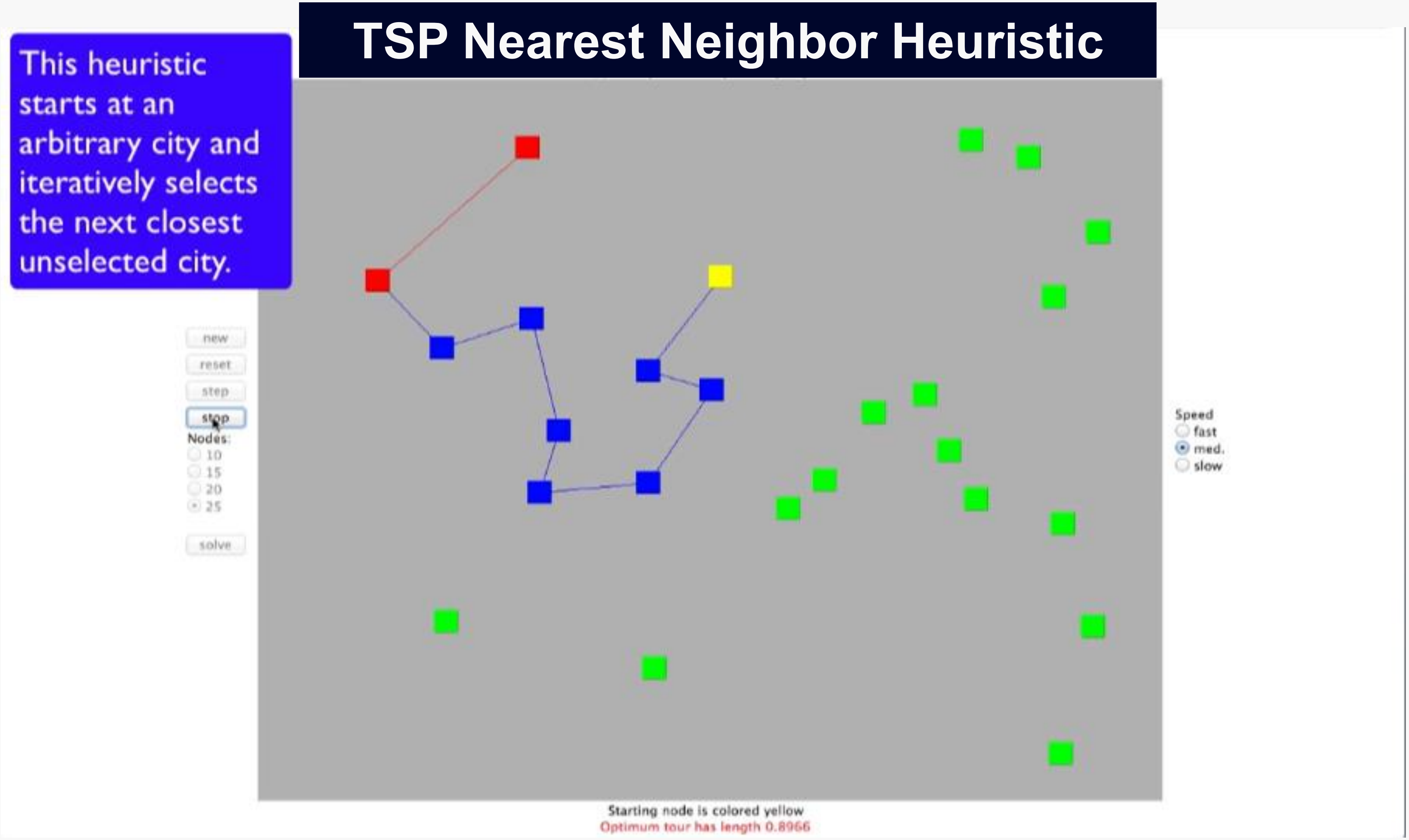
- Construction heuristics: builds a solution from scratch (starting with nothing).
  - Often called “greedy heuristics”. Each step looks good, but it doesn’t look ahead.

- Improvement heuristics (neighborhood search): starts with a solution, and then tries to improve the solution, usually by making small changes in the current solution.

An easy construction heuristic: Nearest unvisited neighbor



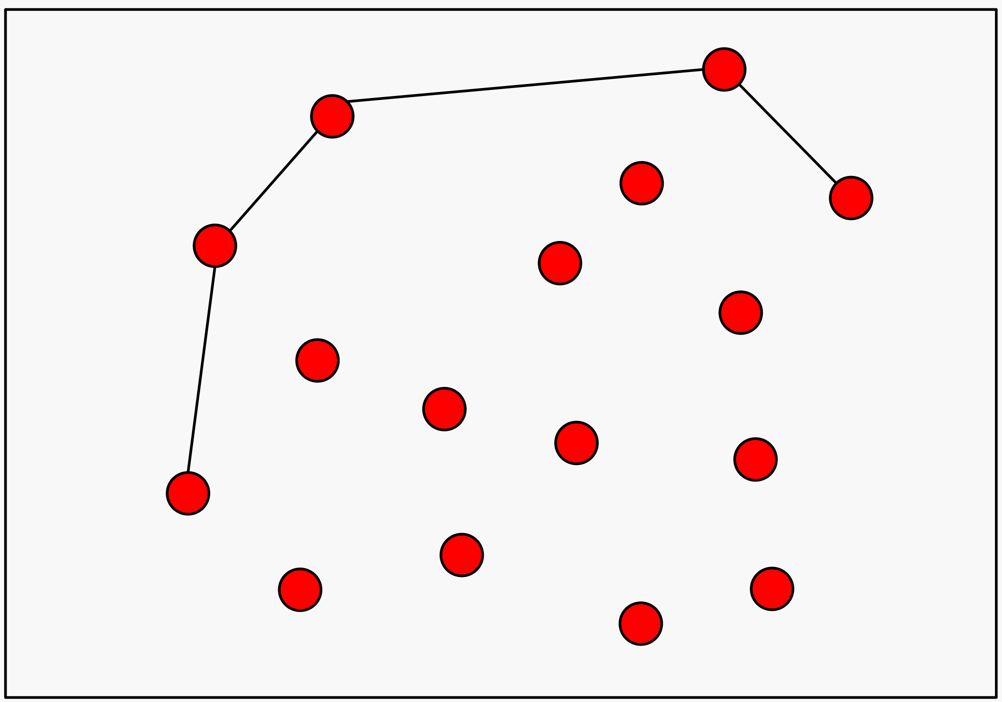
Animations from Stephen Mertens



Courtesy of Stephan Mertens. Used with permission.

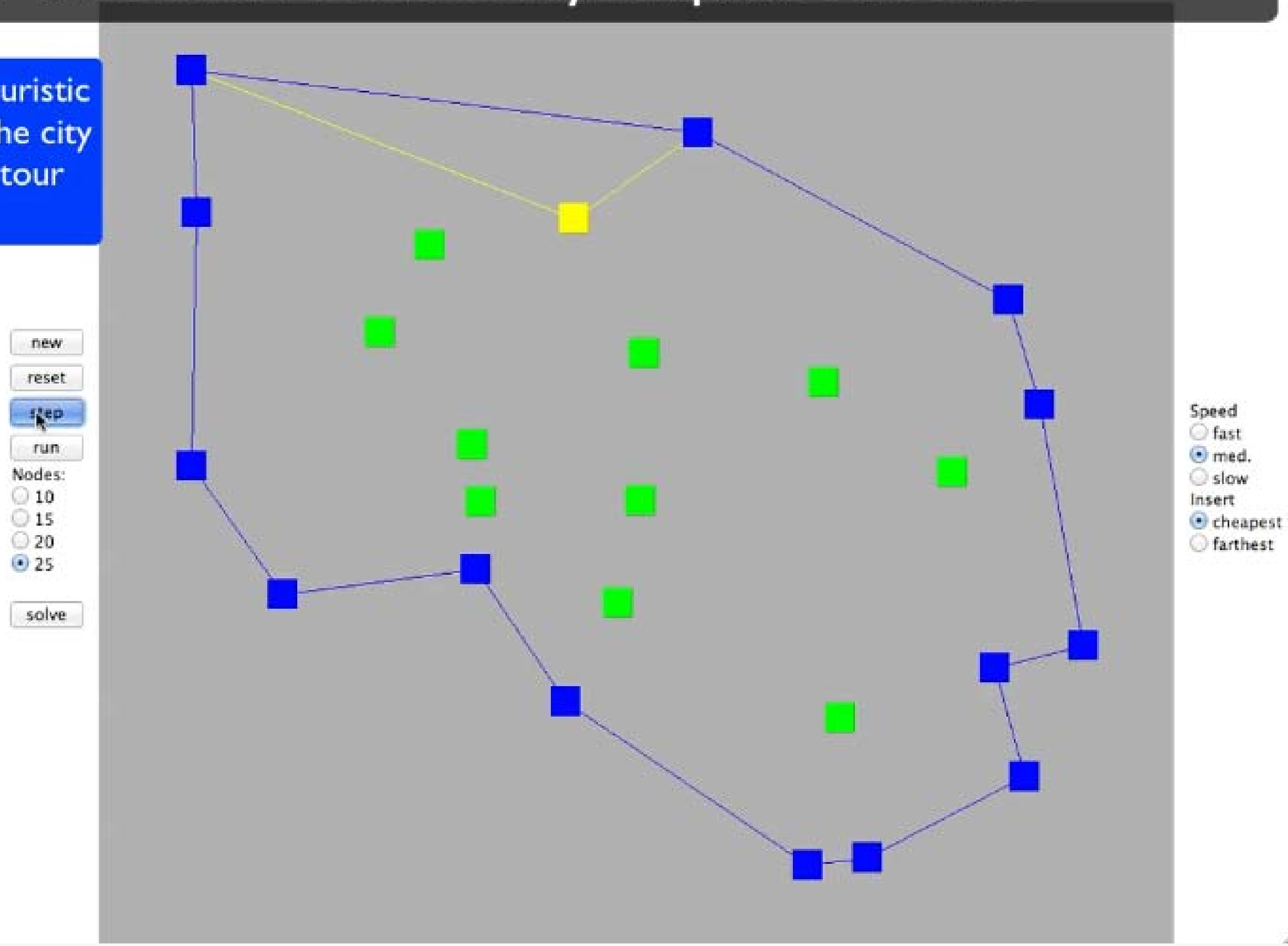
Can we do better in construction?

- **Class exercise: try to develop a heuristic in which we add one city at a time, but the next city can be added anywhere in the tour (not just the beginning or the end.)**
  - **Below is the beginning part of a tour**



TSP Insertion Heuristic by Stephan Mertens

Cheapest Insertion Heuristic
This heuristic selects the city whose addition to the tour has least cost.



Which do you believe will give shorter length tours:

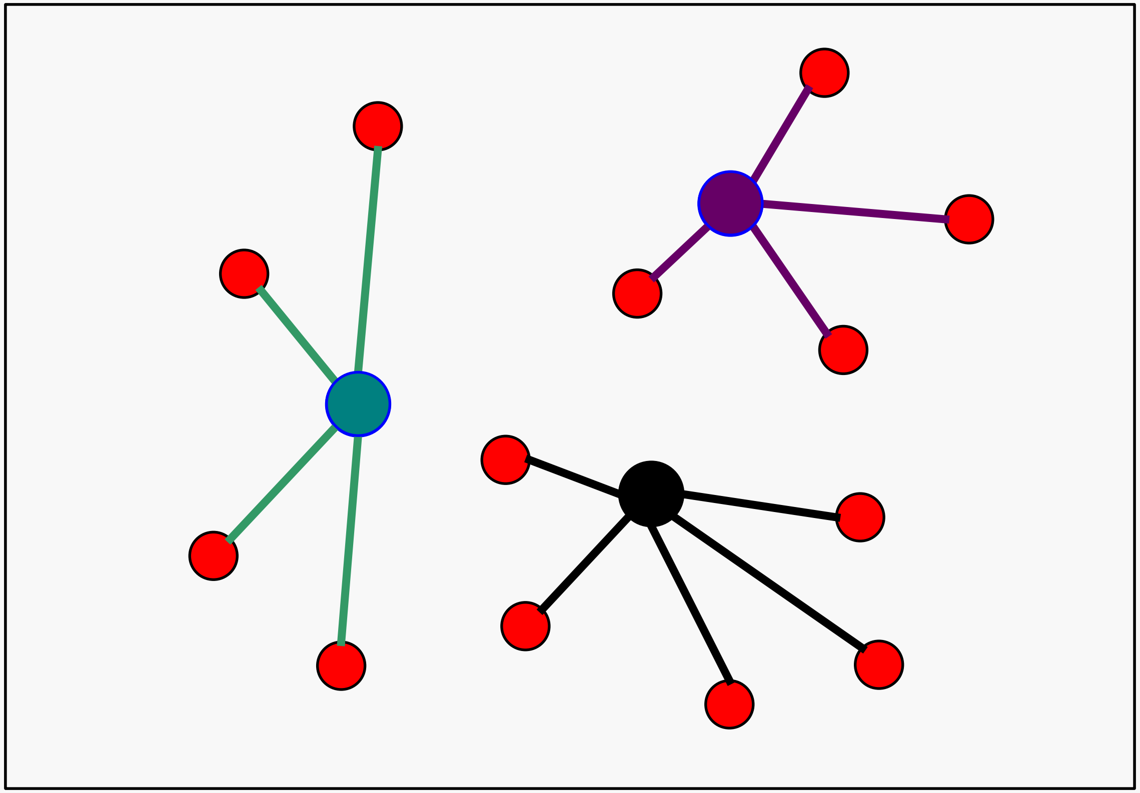
1. The nearest neighbor heuristic

2. An insertion based heuristic

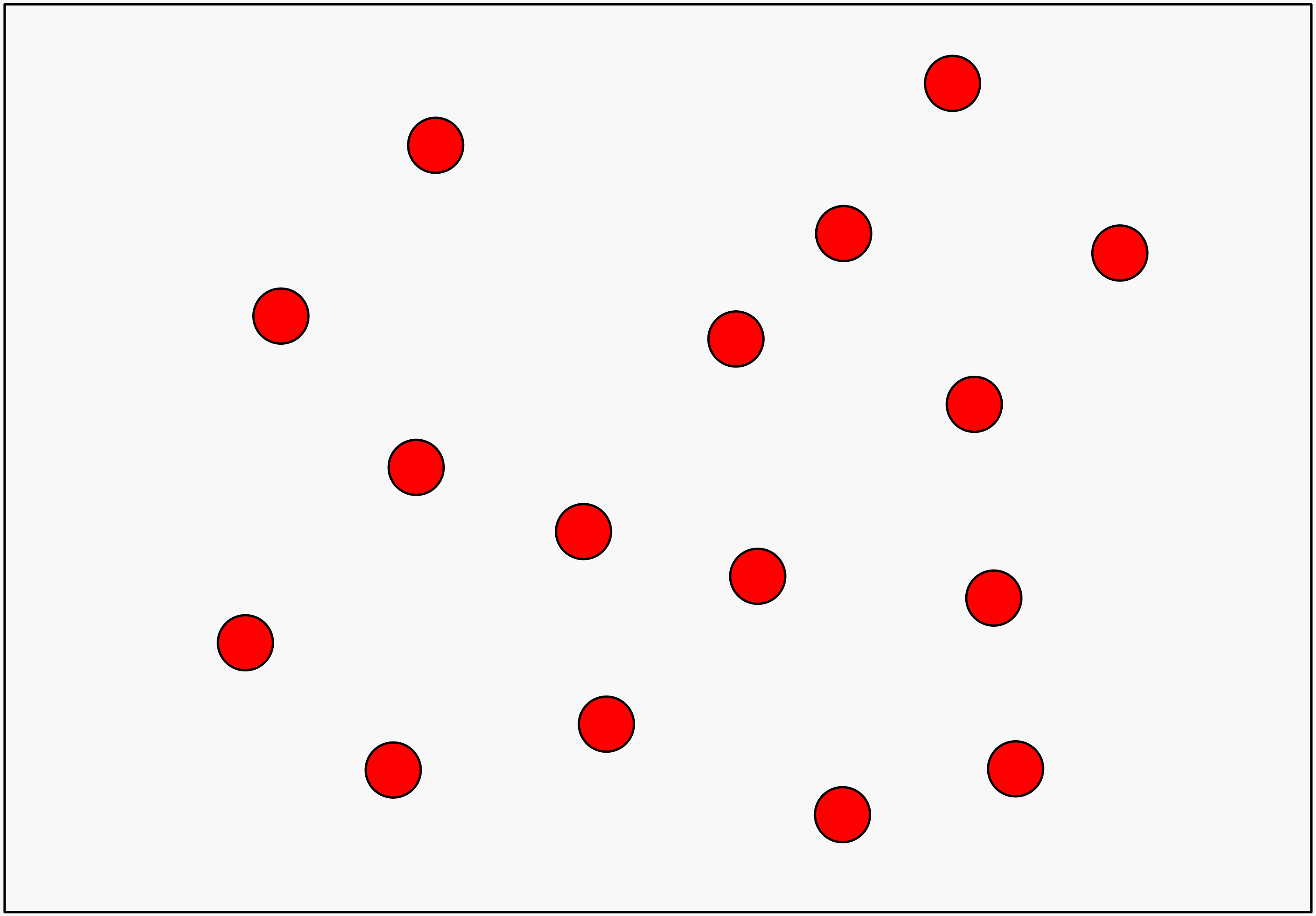
Facility location problems.

Choose K facilities so as to minimize total distance from customers to their closest facility.

- **example with three facilities**



Exercise: try developing a good solution where there are 2 facilities



Exercise: Develop a construction heuristic for the facility location problem

- Data: locations in a city.
- c_{ij} = distance from i to j

Mental Break

Improvement heuristics

- Improvement heuristics start with a feasible solution and look for an improved solution that can be found by making a very small number of changes.
  - This will be made more formal

- Two TSP tours are called 2-adjacent if one can be obtained from the other by deleting two edges and adding two edges.

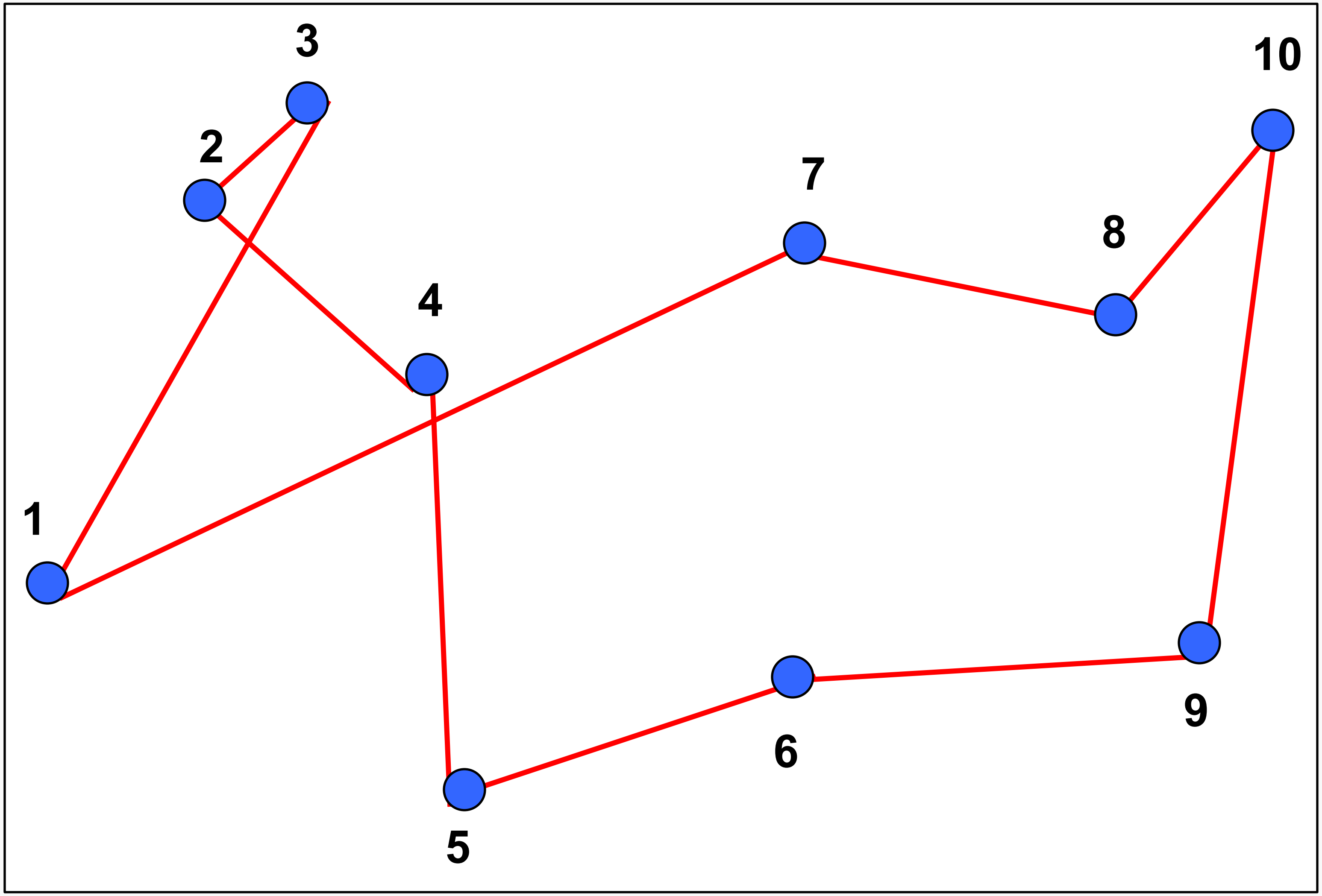
2-opt neighborhood search

- A TSP tour T is called 2-optimal if there is no 2-adjacent tour to T with lower cost than T.

- 2-opt heuristic. Look for a 2-adjacent tour with lower cost than the current tour. If one is found, then it replaces the current tour. This continues until there is a 2-optimal tour.

An improvement heuristic: 2 exchanges.

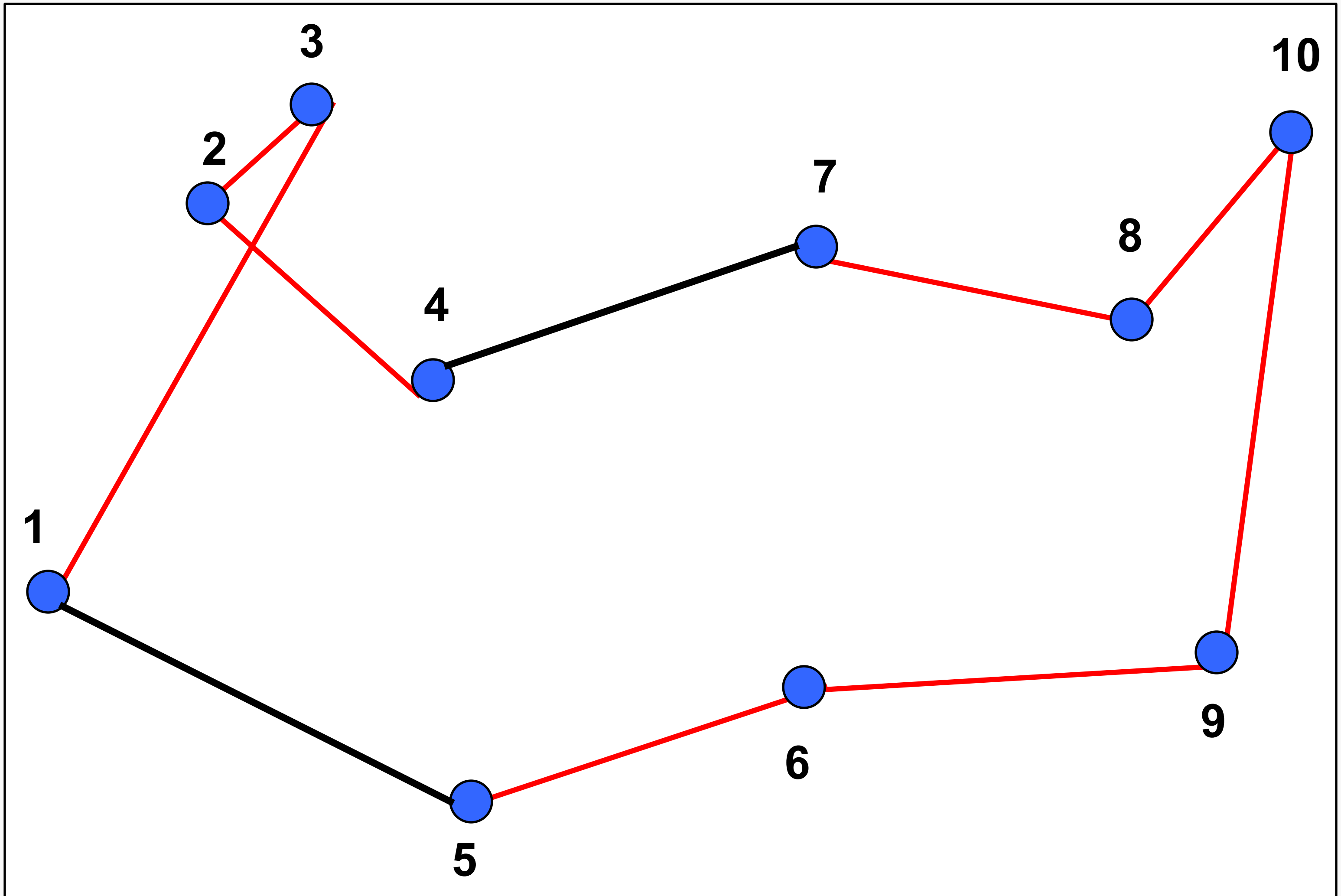
Look for an improvement obtained by deleting two edges and adding two edges.



T
1
3
2
4
5
6
9
10
8
7

After the two exchange

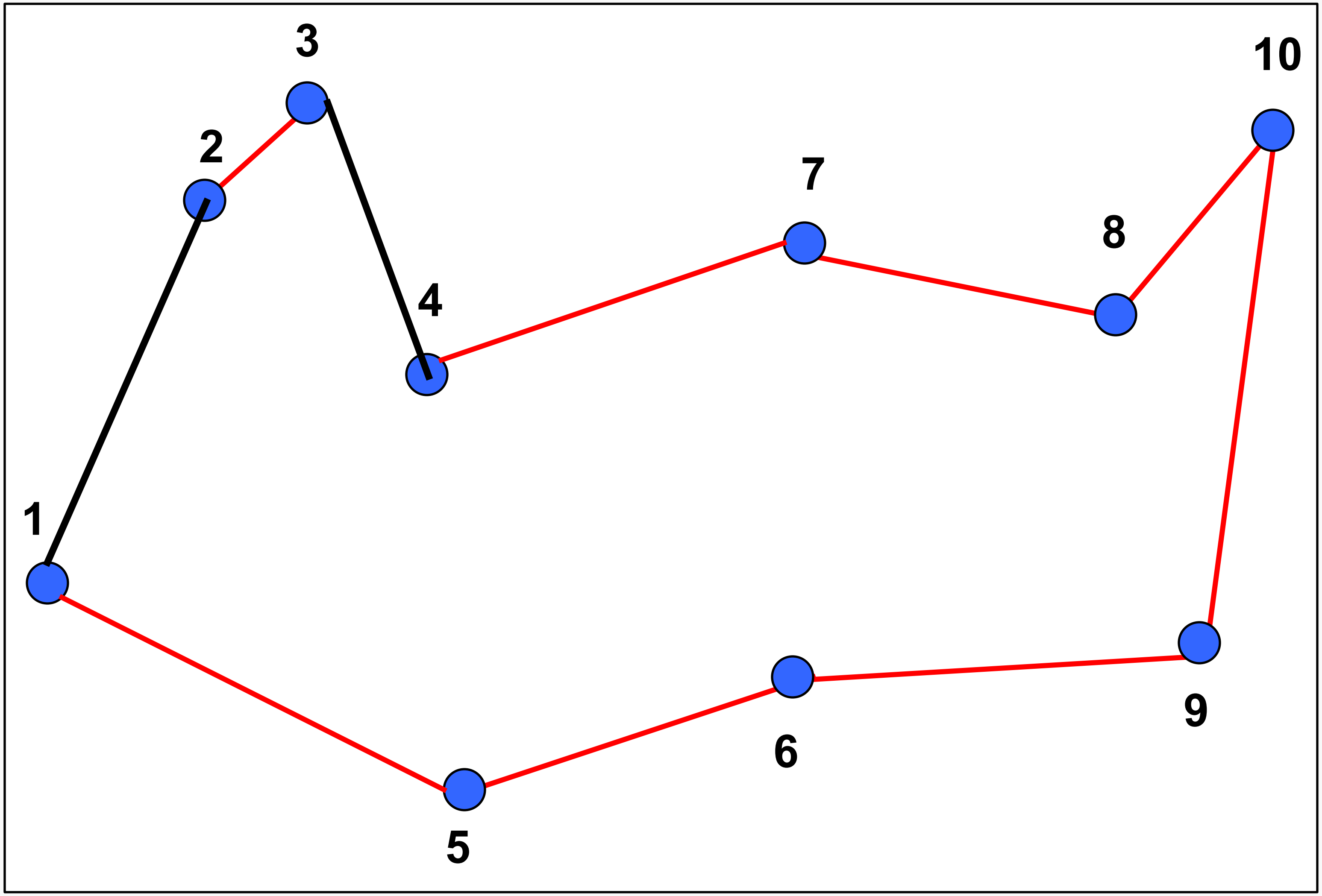
Deleting arcs (4,7) and (5, 1) flips the subpath from node 7 to node 5.



T_1	T_2
1	1
3	3
2	2
4	4
5	7
6	8
9	10
10	9
8	6
7	5

After the two exchange

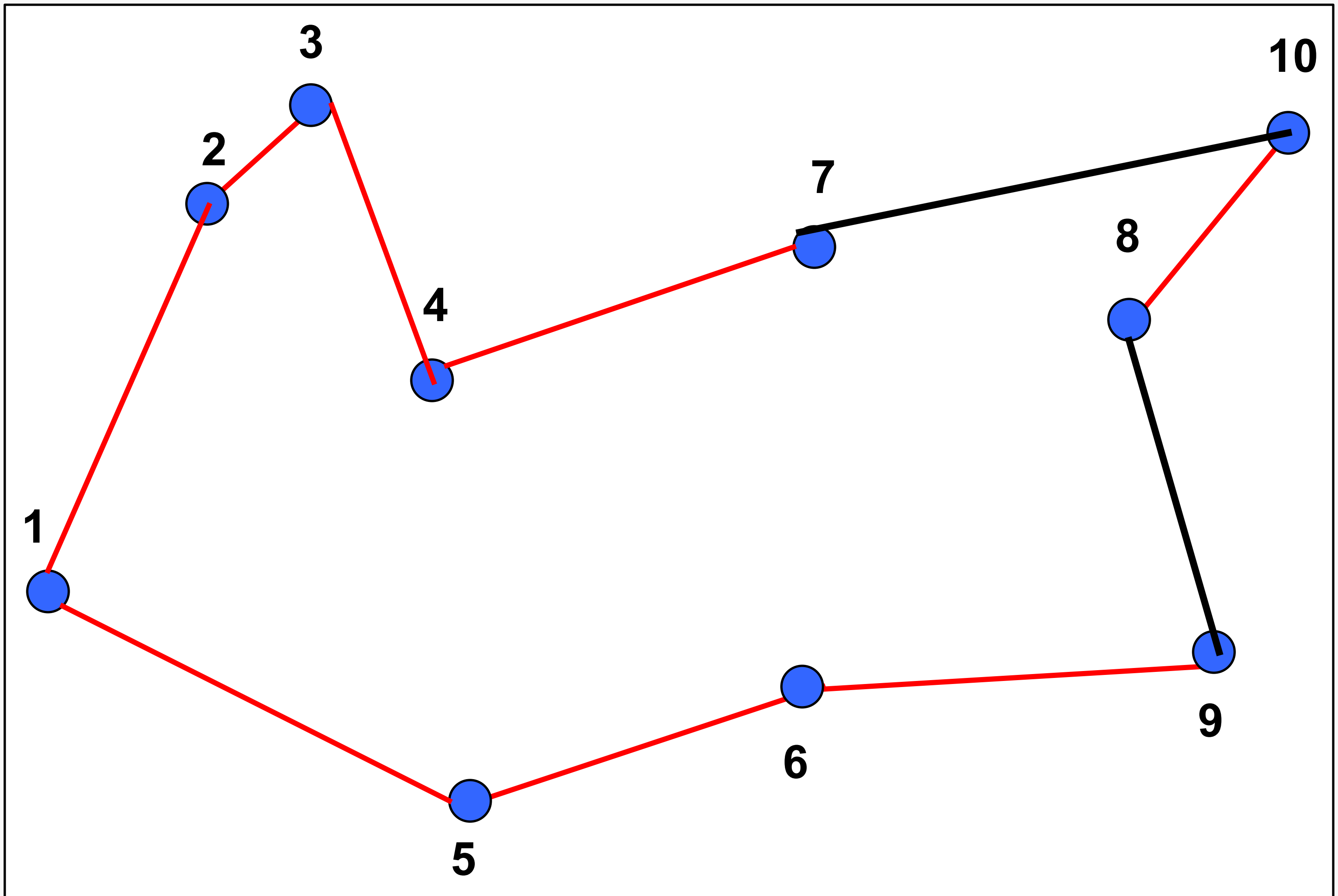
Deleting arcs (1,3) and (2, 4) flips the subpath from 3 to 2.



T_2	T_3
1	1
3	2
2	3
4	4
7	7
8	8
10	10
9	9
6	6
5	5

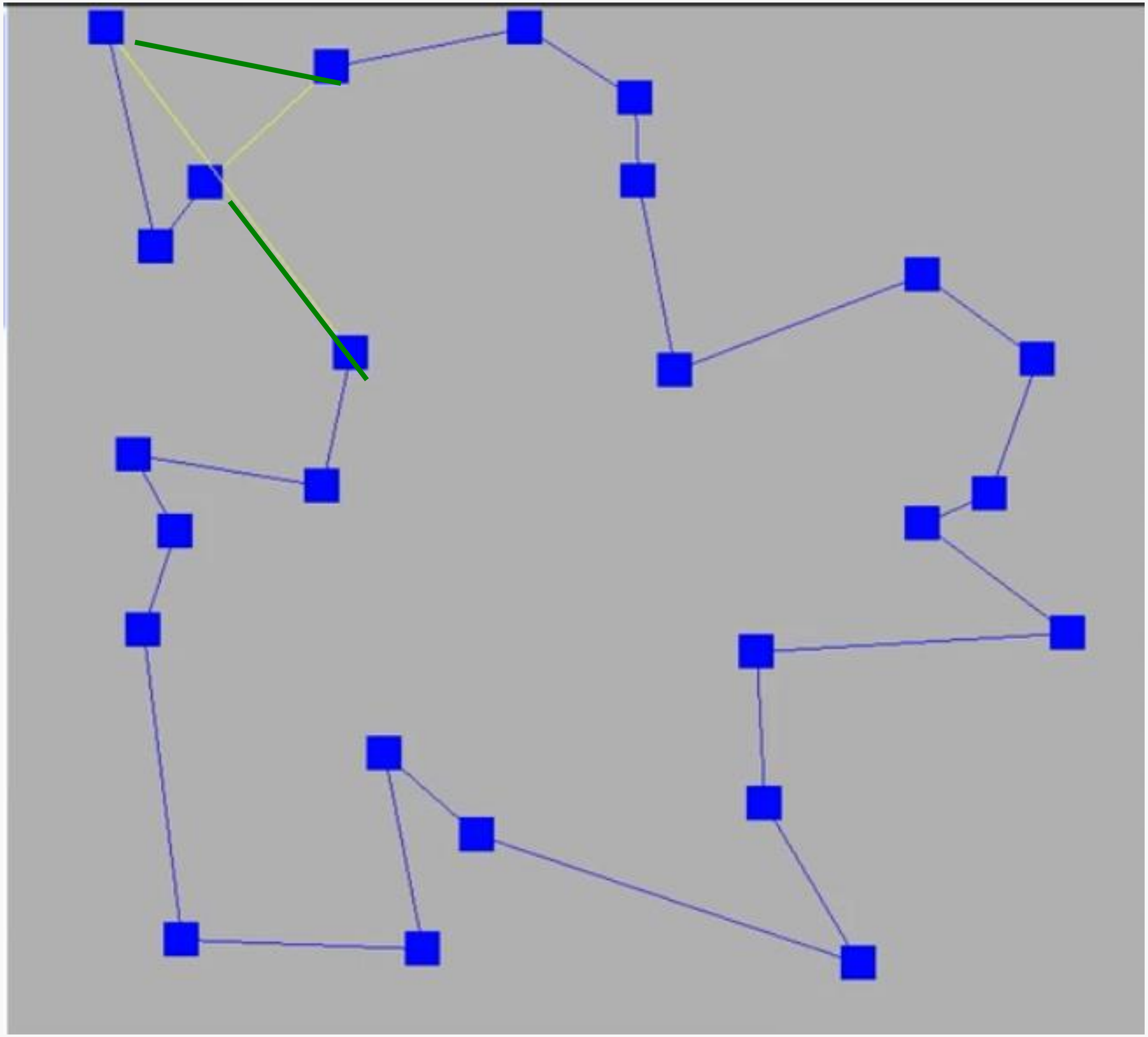
After the final improving 2-exchange

Deleting arcs (7,8) and (10, 9) flips the subpath from 8 to 10.



T_3	T_4
1	1
2	2
3	3
4	4
7	7
8	10
10	8
9	9
6	6
5	5

2-exchange heuristic (also called 2-opt)



3-opt neighborhood

- **Two TSP tours are called <u>3-adjacent</u> if one can be obtained from the other by deleting three edges and adding three edges.**

- **A TSP tour T is called 3-optimal if there is no 3-adjacent tour to T with lower cost than T.**

- **3-opt heuristic. Look for a 3-adjacent tour with lower cost than the current tour. If one is found, then it replaces the current tour. This continues until there is a 3-optimal tour.**

On Improvement Heuristics

Improvement heuristics are based on searching a "neighborhood ." Let N(T) be the neighborhood of tour T.

In this case, the N(T) consists of all tours that can be obtained from T deleting two arcs and inserting two arcs.

Improvement heuristic:

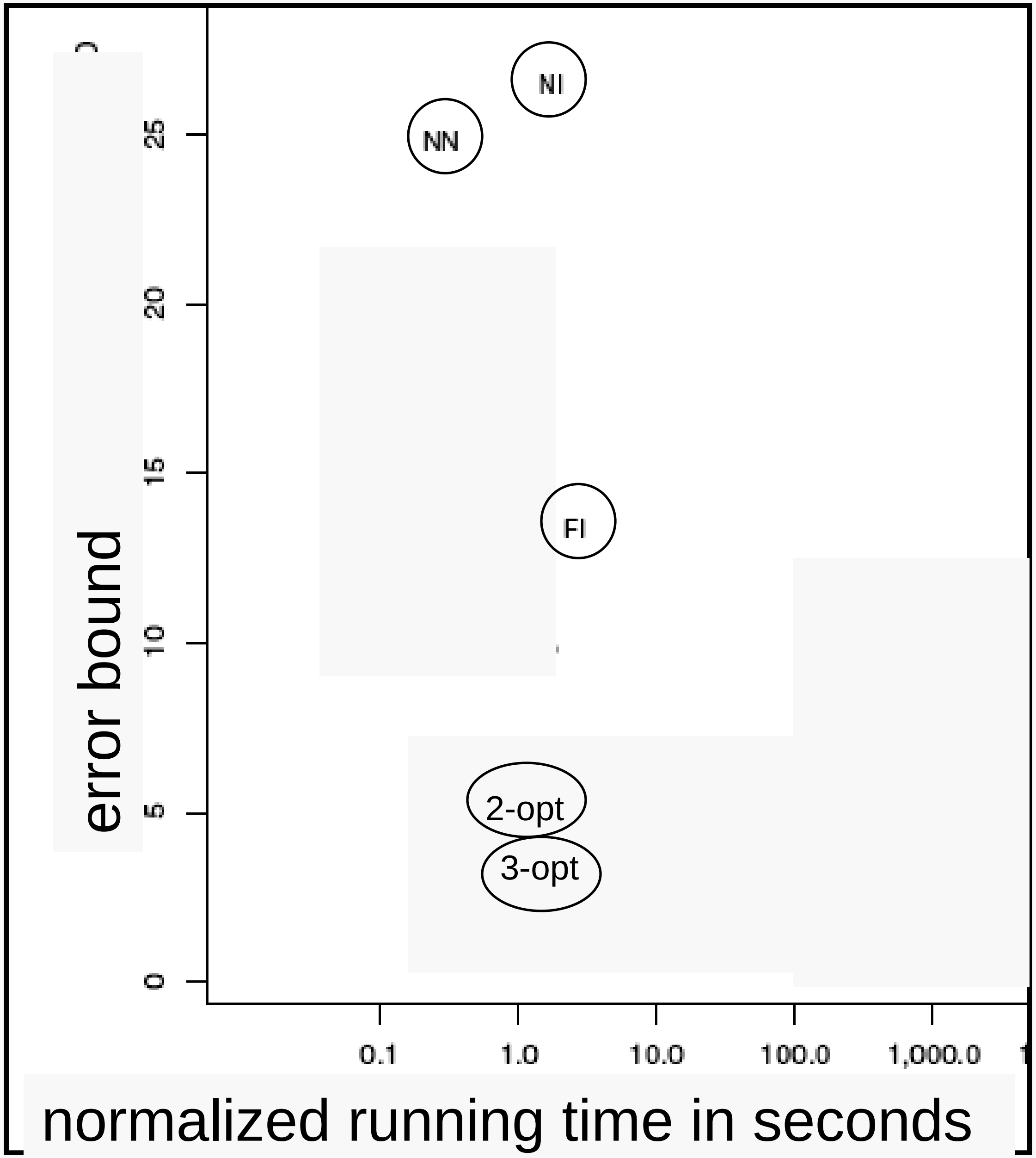
start with tour T

if there is a tour T' ∈ N(T) with c(T') < c(T), then

replace T by T' and repeat

otherwise, quit with a *locally optimal solution*.

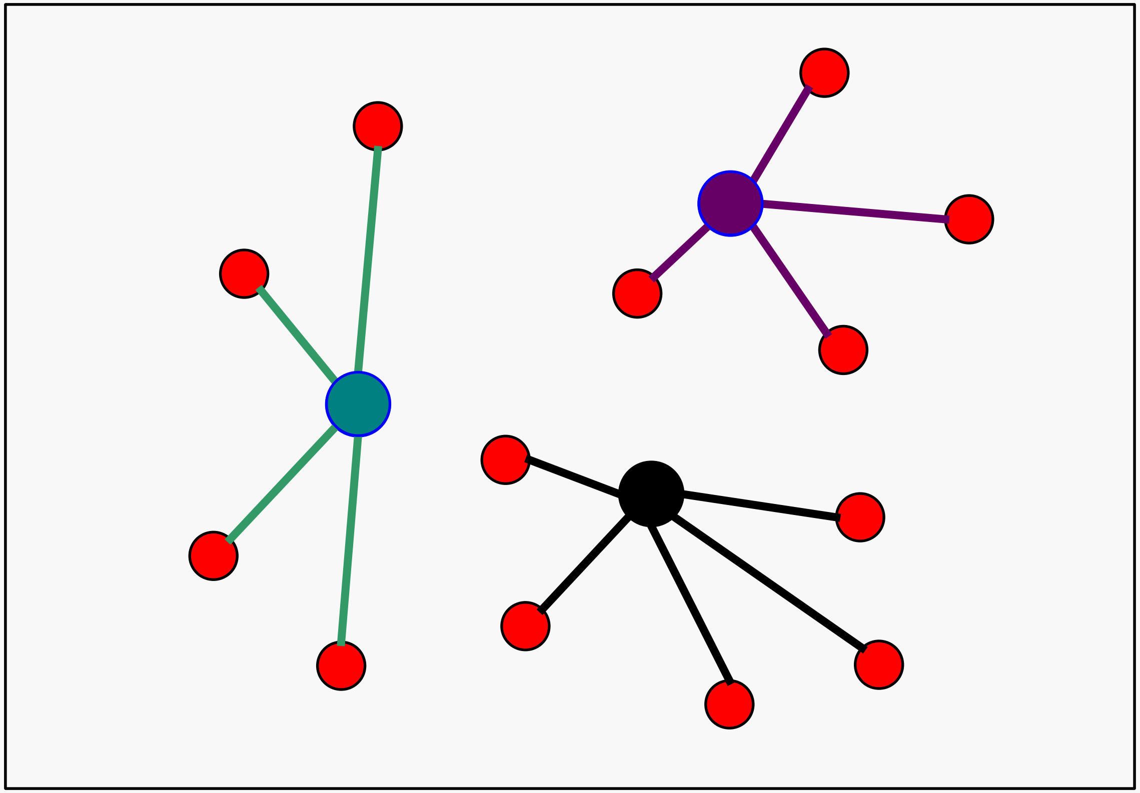
How good are improvement heuristics?



Implementers had to be very clever to achieve these running times.

Facility location problems.

Class exercise. Suppose we want to solve a facility location problem with 3 facilities. Design a neighborhood search heuristic.



Using Randomization

- **An important idea in algorithm development: randomization**

- **Randomization in neighborhood improvement: a way of using the same approach multiple times and getting different answers. (Then choose the best).**

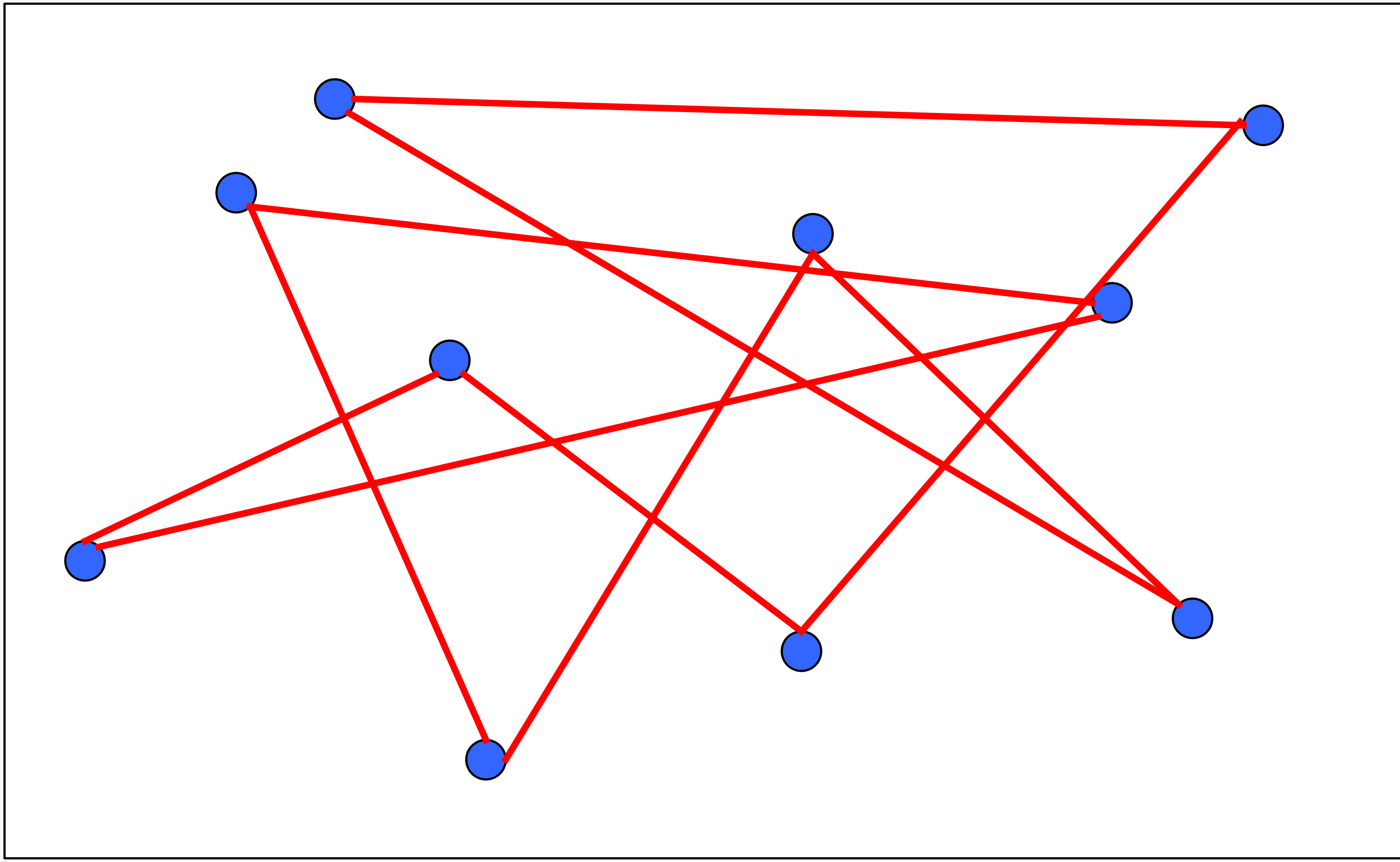
- **Simulated Annealing: randomization in order to have an approach that is more likely to converge to a good solution**

On the use of randomization

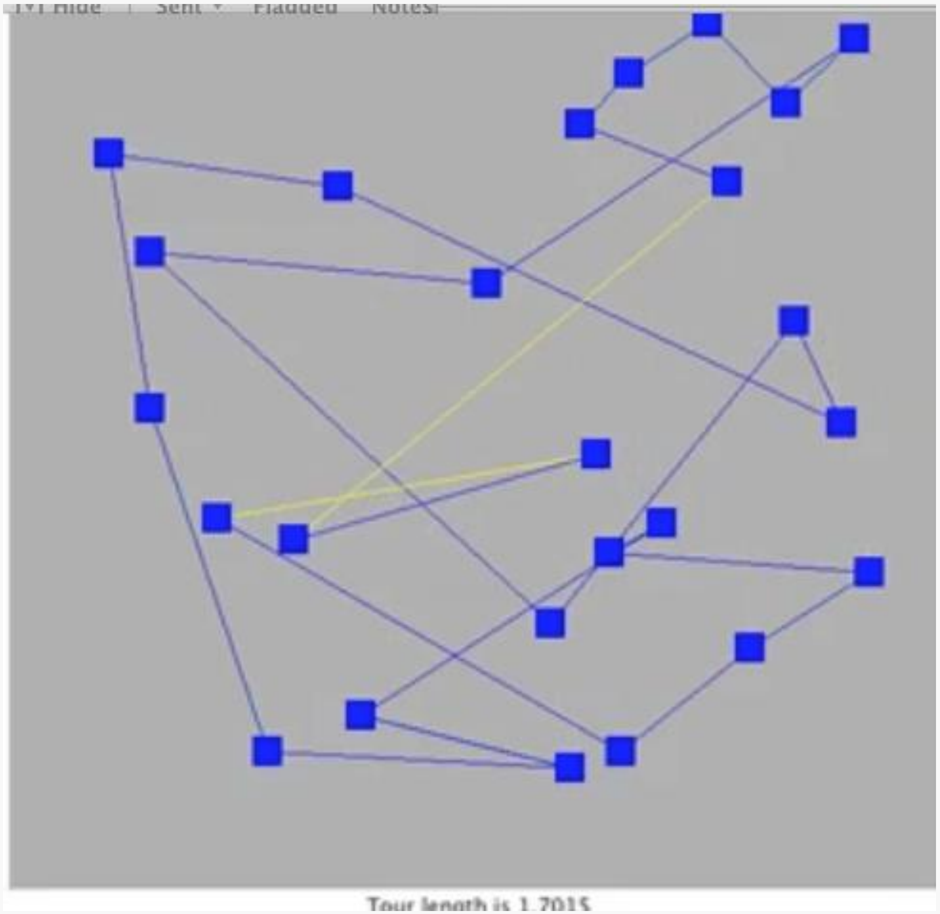
- **Remark: 2-exchanges will behave differently depending on the starting solution.**

- **Randomization based heuristic:**
- **Start with a random tour**
- **use the 2-exchange neighborhood until obtaining a local optimum.**

One difficulty: random tours are terrible.



2-opt heuristic starting from a random solution.



Another use of randomization

- Replace the nearest neighbor tour with the following: at each iteration, visit either the closest neighbor or the second or third closest neighbors. Choose each with 1/3 probability.

- This generates a random tour that is "pretty good" and may be a better starting point than a totally random tour.

Other approaches to heuristics

- The metaphor based approach to the design of heuristics
  – simulated annealing
  – genetic algorithms
  – neural networks
  – ant swarming
- That is, look for something that seems to work well in nature, and then try to simplify it so that it is practical and helps solve optimization problems.

Simulated Annealing

A randomization heuristic based on neighborhood search that permits moves that make a solution worse. It is based on an analogy with physical annealing.

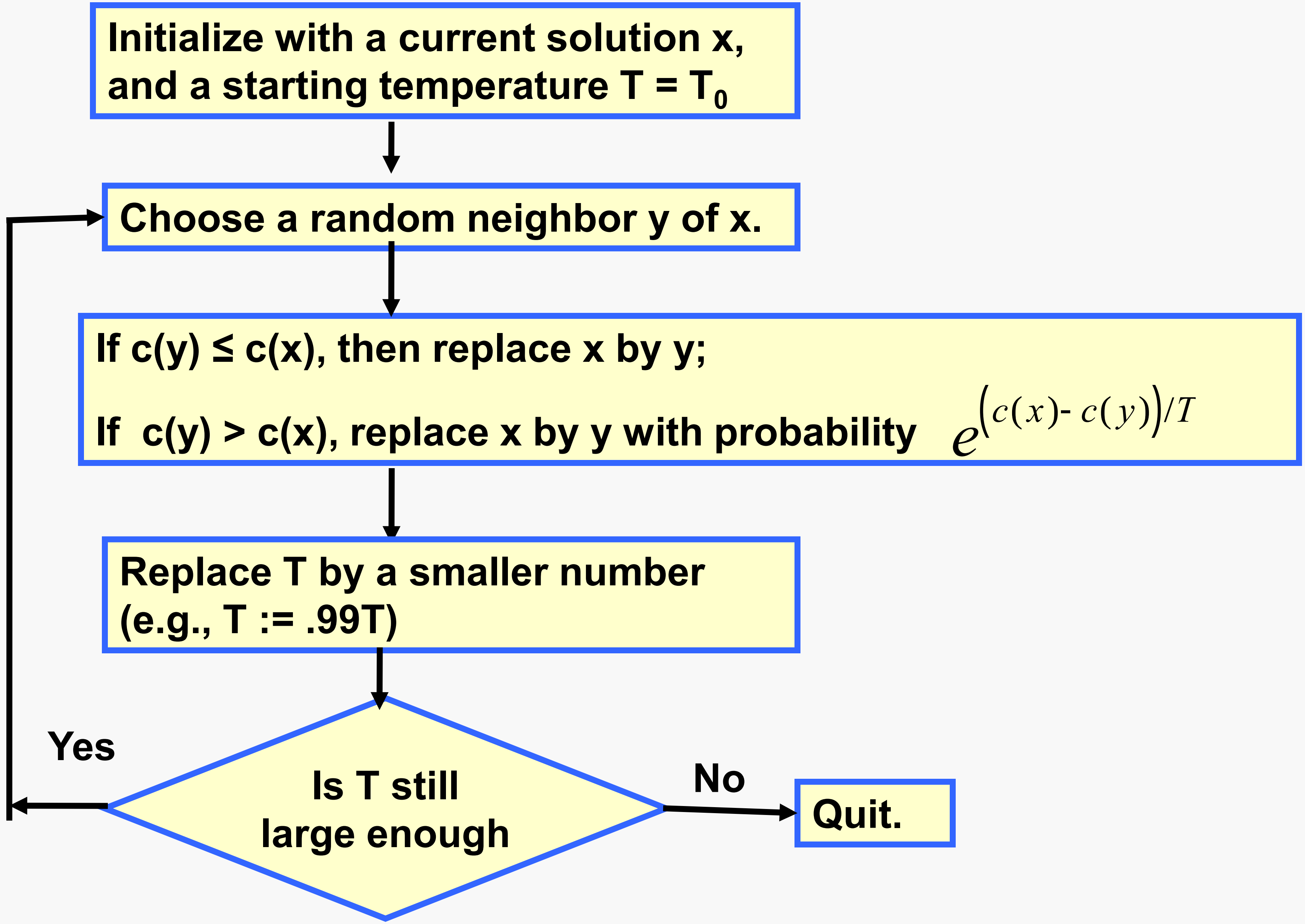
Photo removed due to copyright restrictions.

A glass annealing oven.
www.carbolite.com

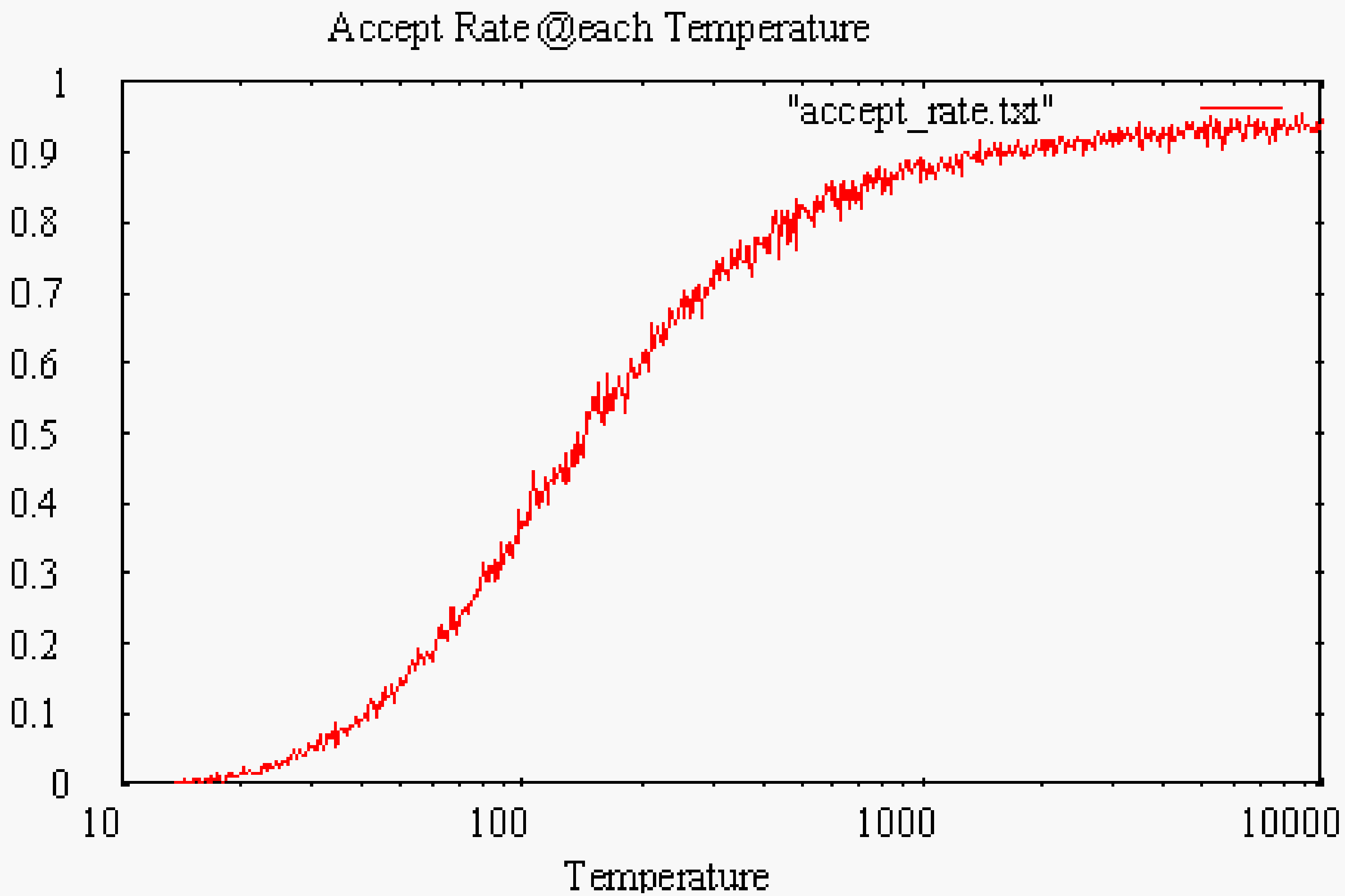
To take a hot material and have it reach a low energy state, one should cool it slowly.

See Eglese, R. W. "Simulated Annealing: A tool for Operational Research." *European Journal of Operational Research* 46 (1990) 271-281. (PDF)

Simulated Annealing: a variation on local search.



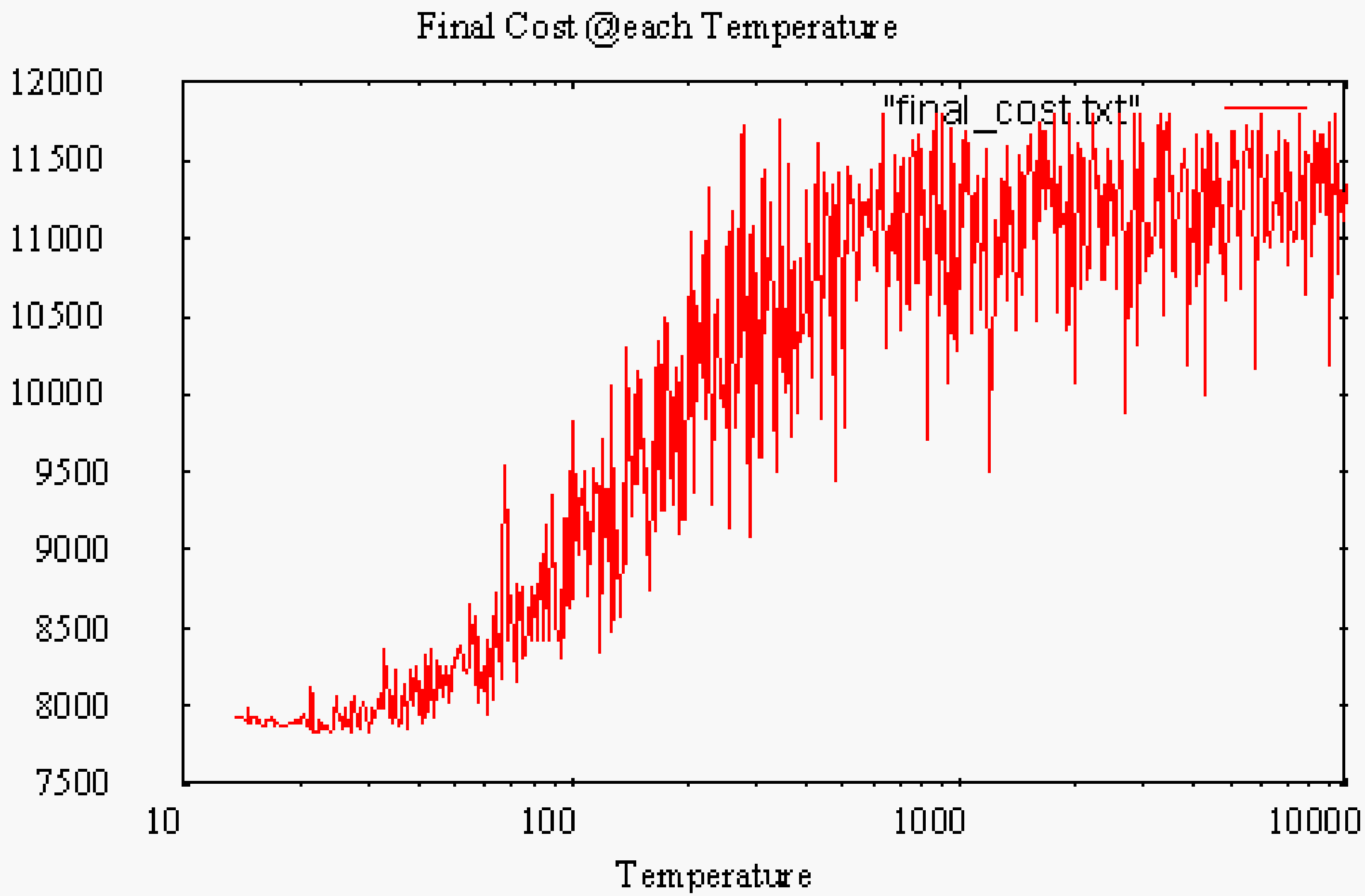
A typical acceptance rate as a function of temp



A network design problem for wireless networks.

Source: Fang, M. Ch. 3 in "Layout Optimization for Point-to-Multi-Point Wireless Optical Networks Via Simulated Annealing & Genetic Algorithm." Senior Project Report, University of Bridgeport, 2000.

A typical graph of Temp vs. cost.



A network design problem for wireless networks.

Source: Fang, M. Ch. 3 in "Layout Optimization for Point-to-Multi-Point Wireless Optical Networks Via Simulated Annealing & Genetic Algorithm." Senior Project Report, University of Bridgeport, 2000.

Summary

- Two types of heuristics
- Use of randomization
- Simulated annealing
- Goal of this lecture: let you get started in solving hard problems.