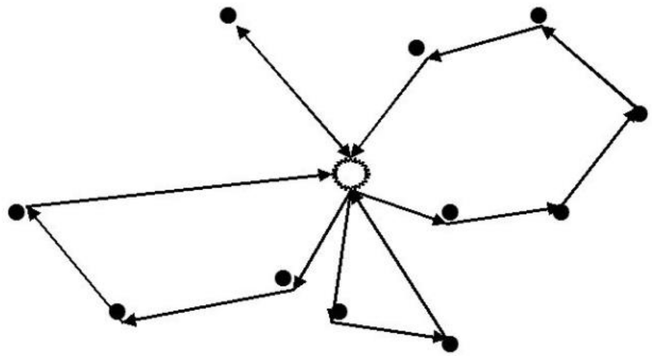
Heuristic Algorithms

章 宇 ZHANG Yu

y.zhang@swufe.edu.cn

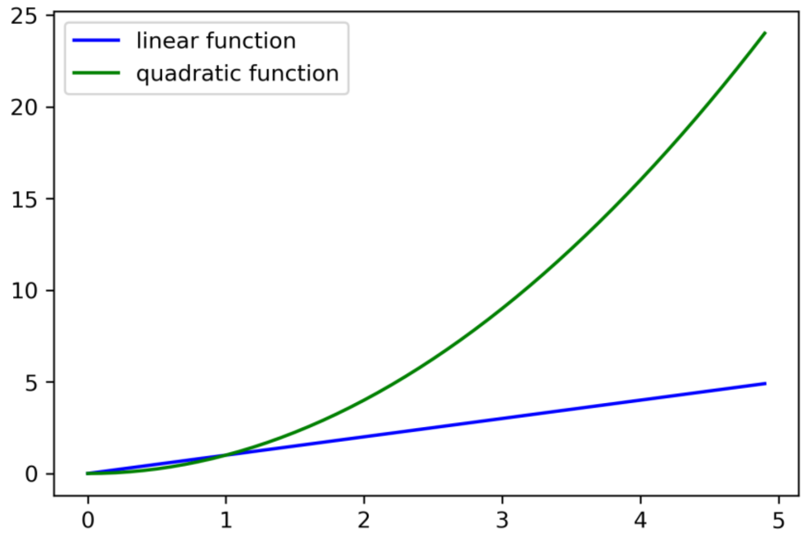
Heuristic Algorithms

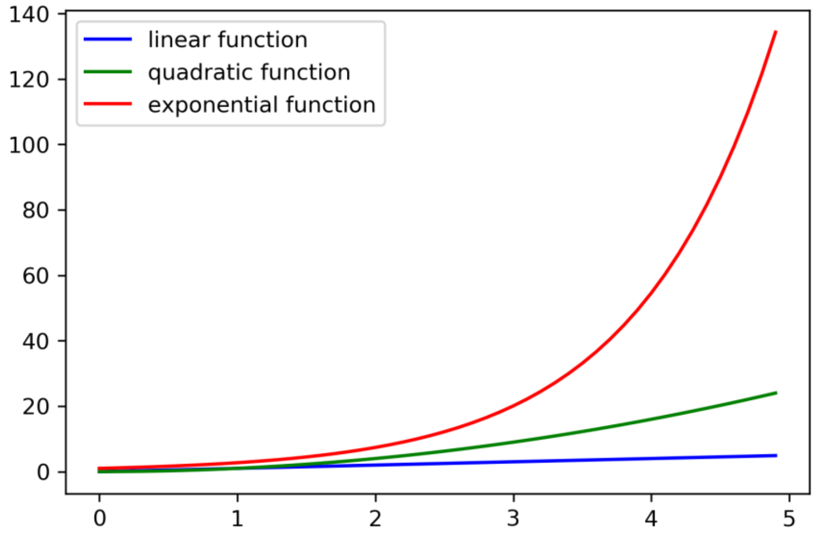
- Various Combinatorial Optimization Problems (COP) in industry
  - TSP
  - VRP
  - Scheduling
  - Packing
  - Location
  - Course timetabling
  - Portfolio optimization
  - …



Heuristic Algorithms

- <u>Bad news</u>: COP is generally NP-hard
  - Can be solved to optimality
  - Solution time increases exponentially
  - May not be practical
    - E.g.: solve VRP with 1000+ nodes within 5 minutes





Heuristic Algorithms

- Good news: compromised solution
  - Solve for suboptimal solution quickly
  - How? Heuristic algorithms!
- Heuristic: a procedure that determines near-optimal solutions to an optimization problem.
  - Sometimes can find optimal solution, but cannot prove its optimality
  - Widely used in industry!

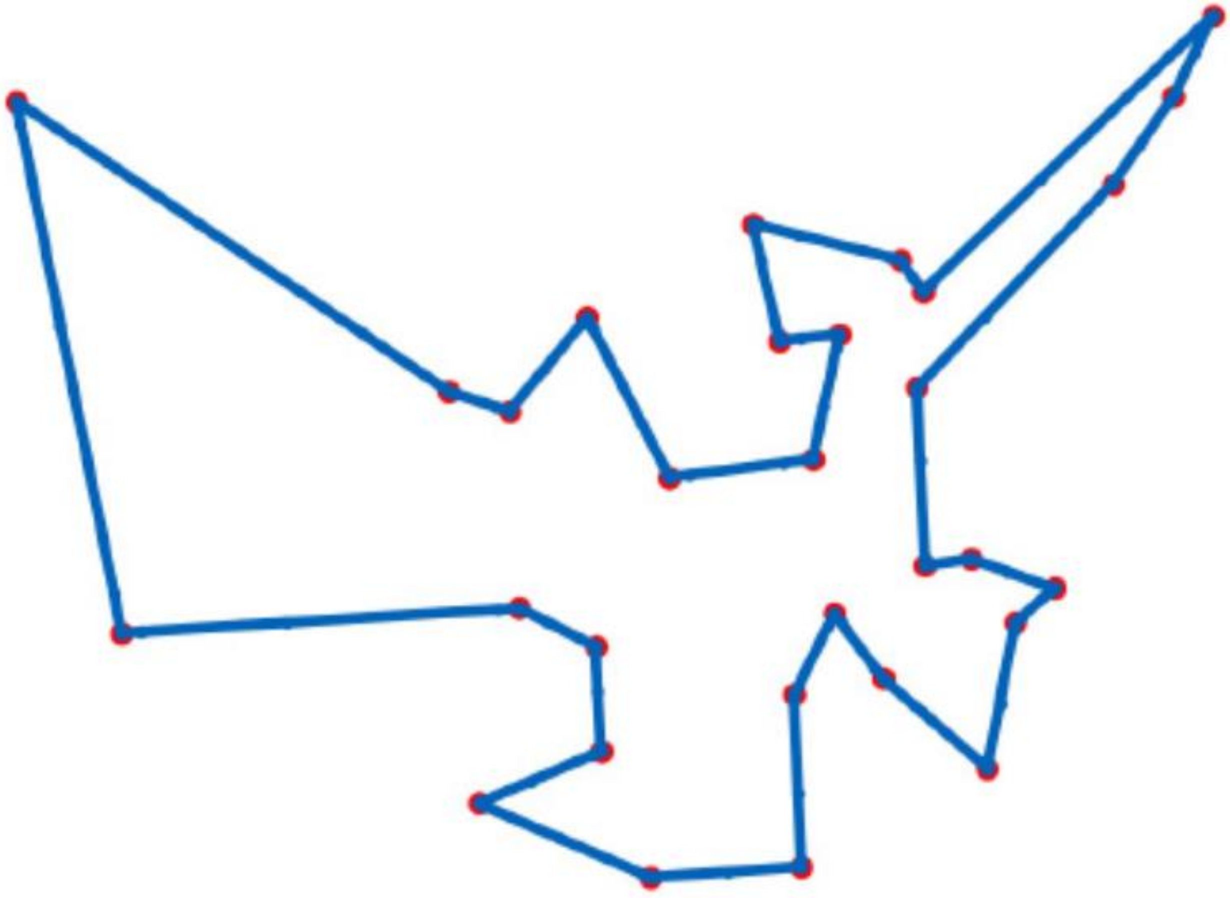
Competition

- JD
  - https://jdata.jd.com/html/detail.html?id=5

- Huawei
  - https://www.noahlab.com.hk/logistics-ranking/#/competition_details
  - https://competition.huaweicloud.com/information/1000041601/circumstance

TSP

- Problem: finding a shortest circle visiting each node exactly once.
- E.g.:

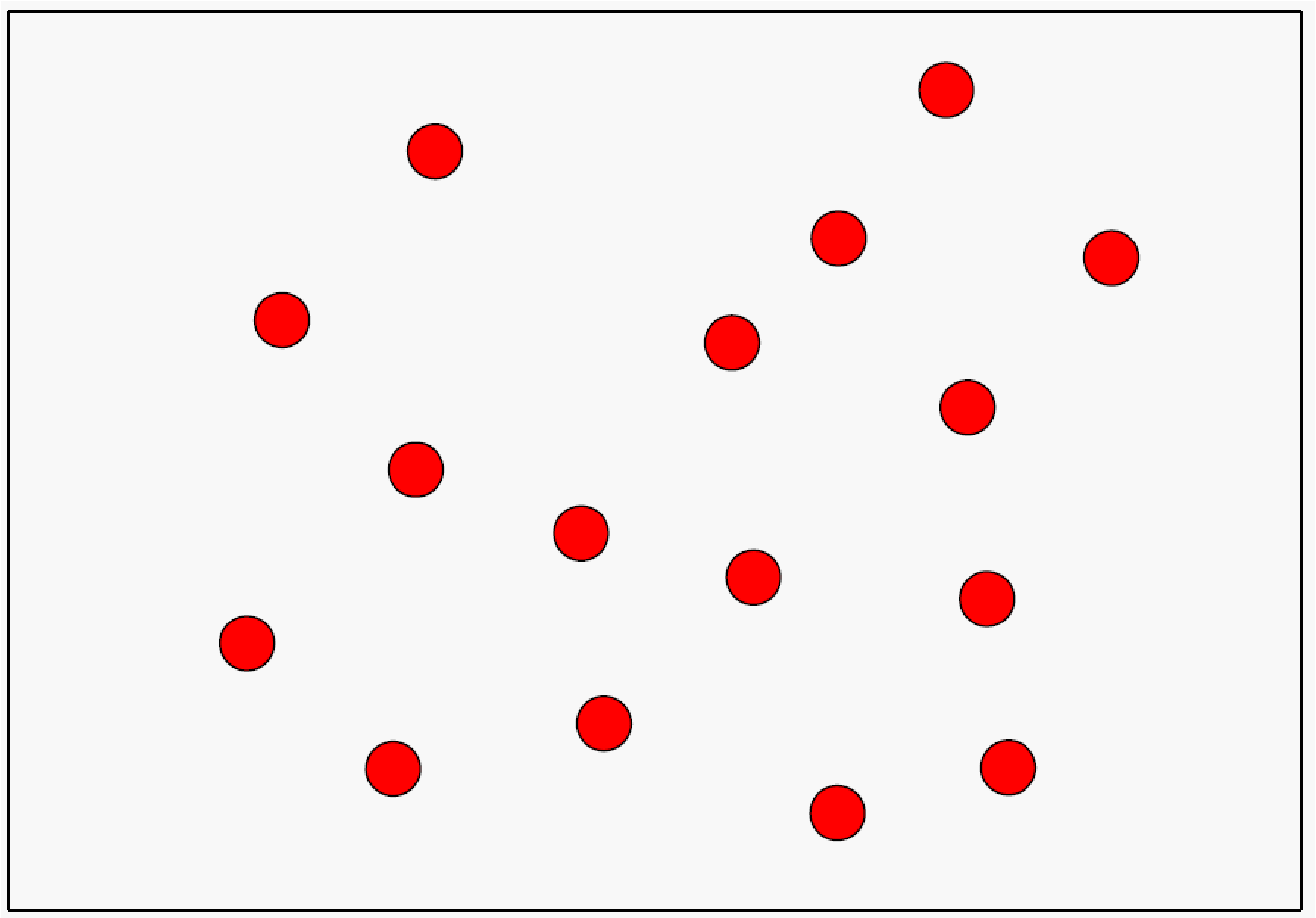


TSP

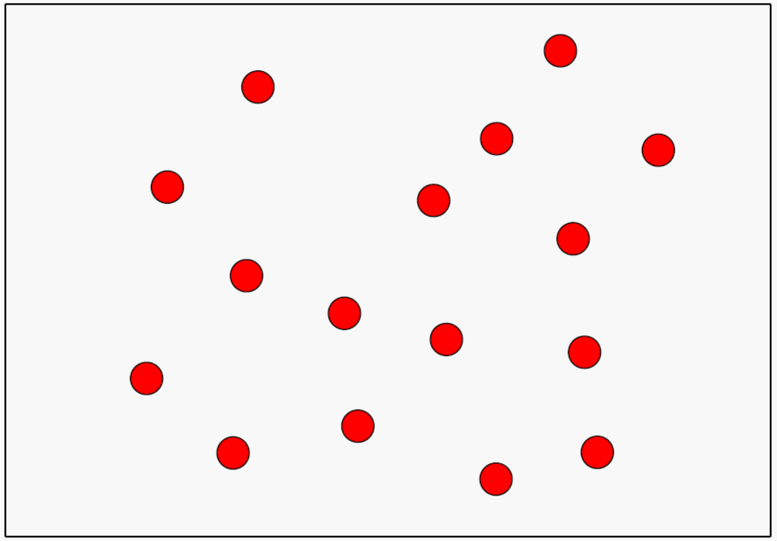
- Mona Lisa Instance
  - 100,000 nodes
  - Best known solution by heuristics:
    - Dist = 5,757,191
  - Best known lower bound
    - Dist = 5,757,084
  - Gap = 0.0019%
  - $1,000 for finding better
    - http://www.math.uwaterloo.ca/tsp/

TSP

- How to solve it? Brain storming…



TSP



- Idea 1: Enumeration
- # feasible routes?
  - $(\#\text{nodes} - 1)!$
- E.g.:
  - 101 nodes; # routes $= 100! \approx 10^{158}$
  - Super powerful computer
    - # CPUs is 10^{80} $\approx$ # atoms in the universe
    - 5.39×10^{-44} seconds (Planck time) per operation
  - $\sim 5.39 \times 10^{34}$ seconds to enumerate
  - Compare to 4.33×10^{17} s. $\approx$ age of universe

TSP

- Idea 2: Greedy Algorithm
  - Iteratively select the shortest arc

TSP

- Idea 3: Nearest neighbor
  - Iteratively select the nearest unvisited node

TSP

- Idea 4: Cheapest insertion
  - Iteratively insert a node to a subtour in a shortest possible way.

TSP

- <u>Idea 5</u>: Random insertion
  - Iteratively insert a random node to current subtour

TSP

- Guess: which performs better?
  - 1. Nearest neighbor
  - 2. Cheapest insertion

TSP

- Idea 6: 2-opt
  - Swap two edges

TSP

- Idea 7: 3-opt
  - Swap three edges

TSP

- Summary
- 1. Construction heuristics
  - Nearest neighbor
  - Cheapest insertion
  - ……
- 2. Improvement heuristics (local search)
  - 2-opt
  - 3-opt
  - ……

TSP

- Key idea:

  construct, then **improve**

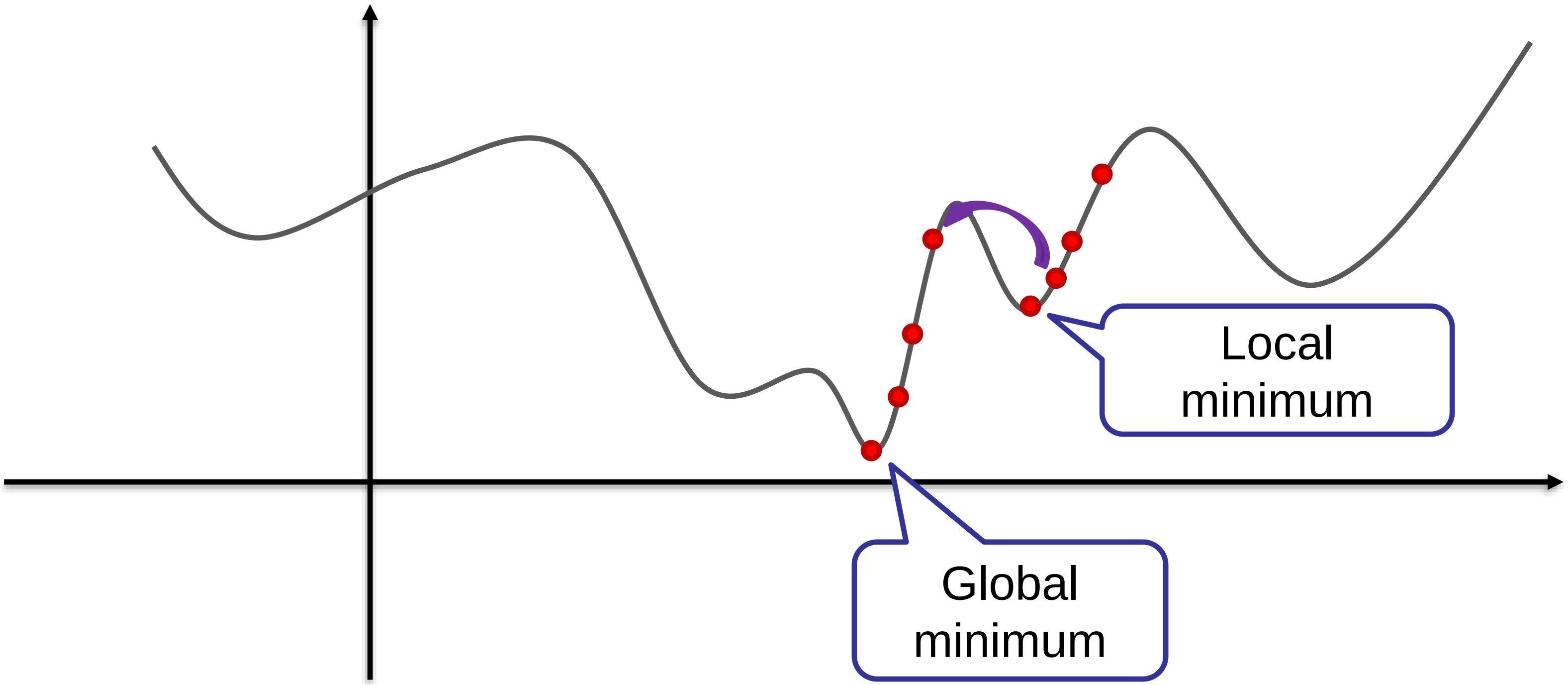
- Common wisdom**：**

  Solution quality depends largely on **improvement heuristics!**

- Improvement
  - Solution representation (node sequence)
  - Definition of neighbor (2-opt)
  - Search strategy (optimal neighbor)

TSP

- Caution: Local optimal



How to escape from local minimum?

Simulated Annealing

- Main idea: introduce randomness
  - Permits worse movement with some probability
  - Probability decreases with time
  - tends to stability
  - analogy with physical annealing

Simulated Annealing

- How to define the probability
  - law of thermodynamics
  - Probability is

$$p = e^{\frac{\Delta E}{kT}}$$

    - ΔE: difference of objective values
    - k: a positive constant
    - T: current temperature
    - Hence, $p \in (0,1)$ when $\Delta E < 0$

Simulated Annealing

- Analogy
  - Tour distance ⇔ energy level
  - Feasible solution ⇔ system state
  - Neighbore solution ⇔ change of state
  - Control parameter ⇔ temperature
  - Final solution ⇔ status when solid

Simulated Annealing

Select an initial solution $\omega \in \Omega$
Select the temperature change counter $k = 0$
Select a temperature cooling schedule, t_k
Select an initial temperature $T = t_0 \geq 0$
Select a repetition schedule, M_k, that defines the number of iterations executed at each temperature, t_k
Repeat
    Set repetition counter $m = 0$
    Repeat
        Generate a solution $\omega' \in N(\omega)$
        Calculate $\Delta_{\omega,\omega'} = f(\omega') - f(\omega)$
        If $\Delta_{\omega,\omega'} \leq 0$, then $\omega \leftarrow \omega'$
        If $\Delta_{\omega,\omega'} > 0$, then $\omega \leftarrow \omega'$ with probability $\boxed{exp(-\Delta_{\omega,\omega'}/t_k)}$
        $m \leftarrow m + 1$
    Until $m = M_k$
    $k \leftarrow k + 1$
Until stopping criterion is met

"Metropolis Criterion"

Pseudo-code: A.G. Nikolaev and S. H. Jacobson, Simulated annealing. in Handbook of Metaheuristics, 2nd Ed.

Simulated Annealing

- Illustration

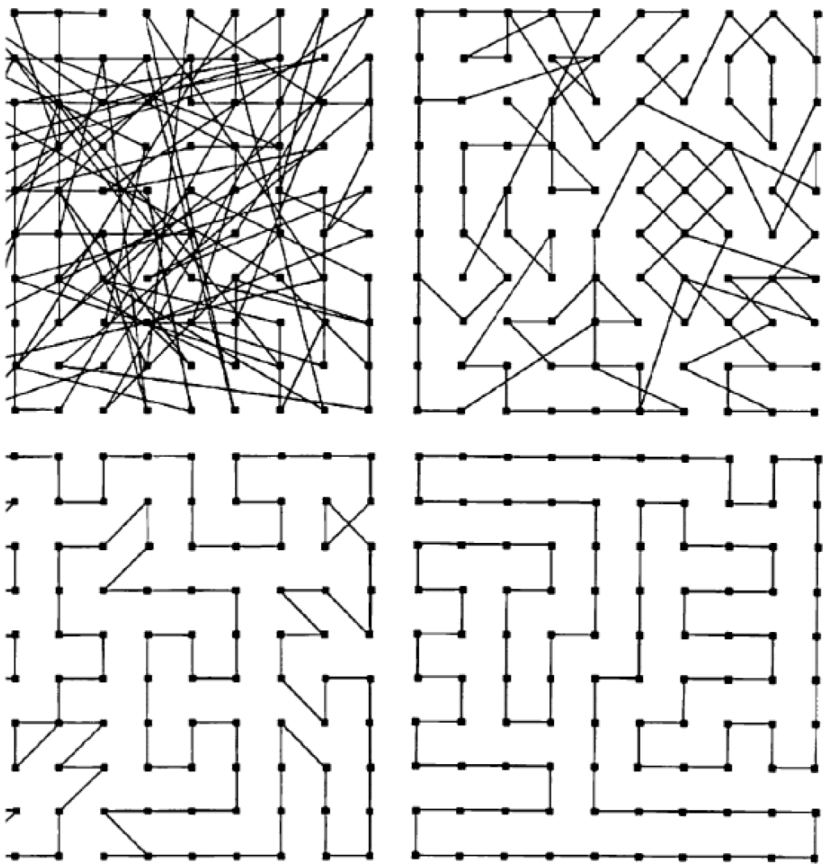
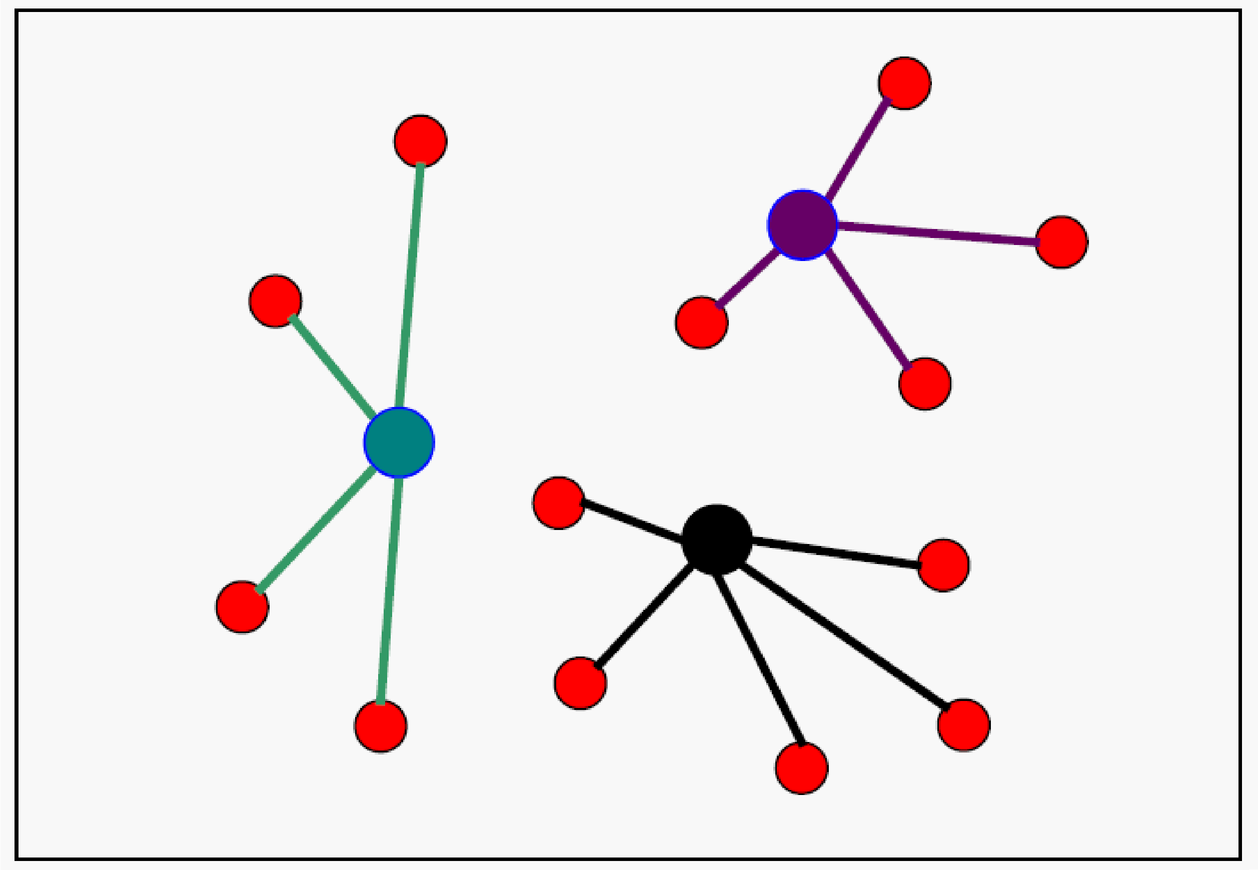


Image:E. K. Burke and G. Kendall, Search Methodologies, Introductory Tutorials in Optimization and Decision Support Techniques. Springer, 2014.

Facility Location

- Problem: Determine the facilities to open and assign customer to facilities to minimize the fixed and variable costs



Facility Location

- Question: Design a heuristic algorithm