MetaHeuristics

INF273

#2:Heuristics

Ahmad Hemmati

Optimization Group
Dept. of Informatics
University of Bergen

Spring Semester
2022

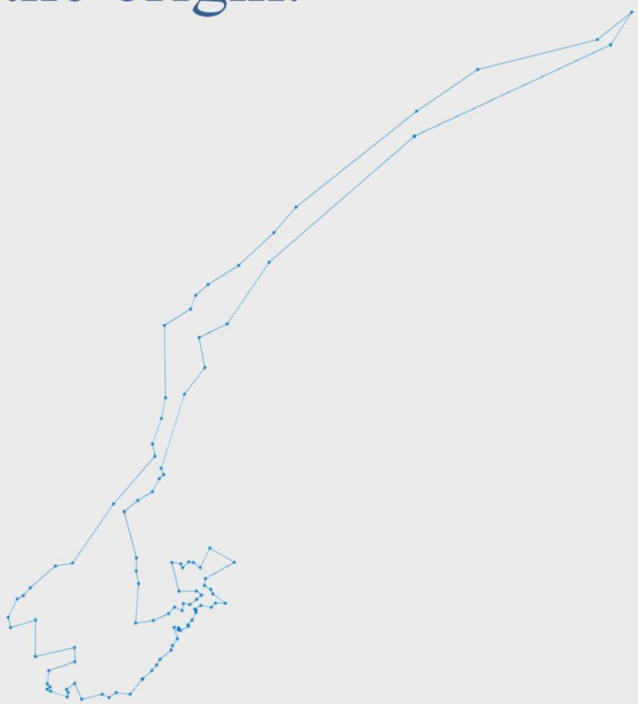
Agenda

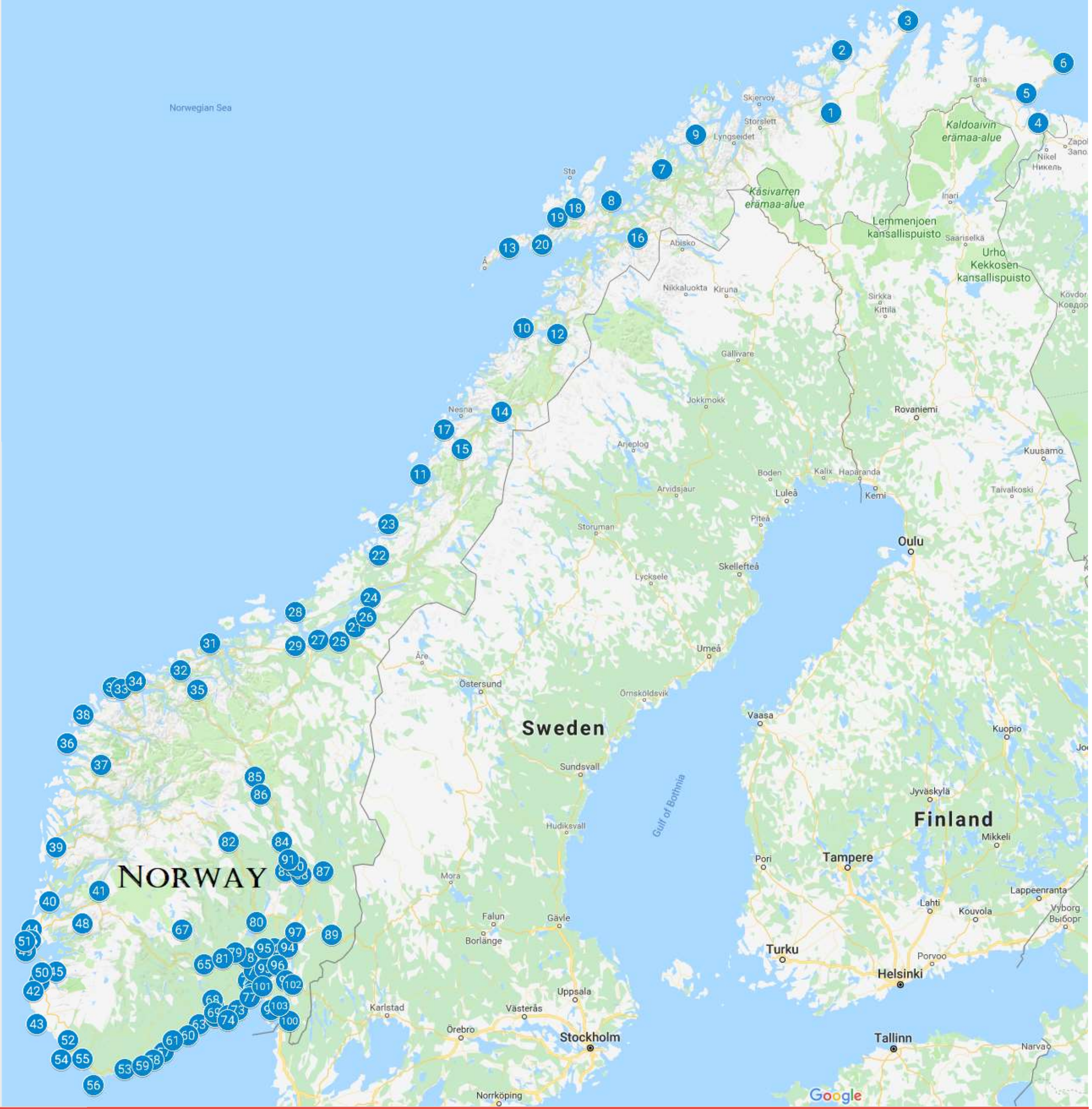
- Construction heuristics
- Improvement heuristics

- Travelling Salesman Problem
- Production Scheduling
- Knapsack Problem

Travelling Salesman Problem

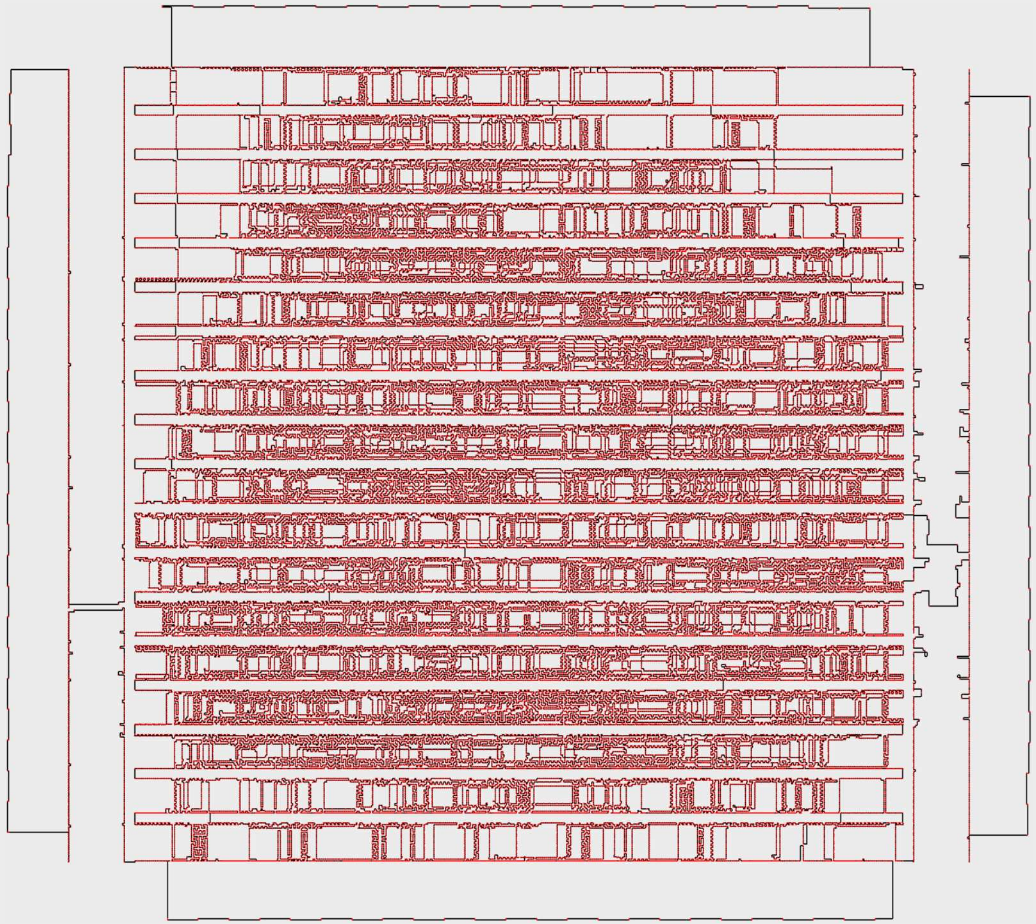
Starting from an origin node, find the minimum distance required to visit each node once and only once and return to the origin.





Travelling Salesman Problem (TSP)

- VLSI problem (85,900 nodes)

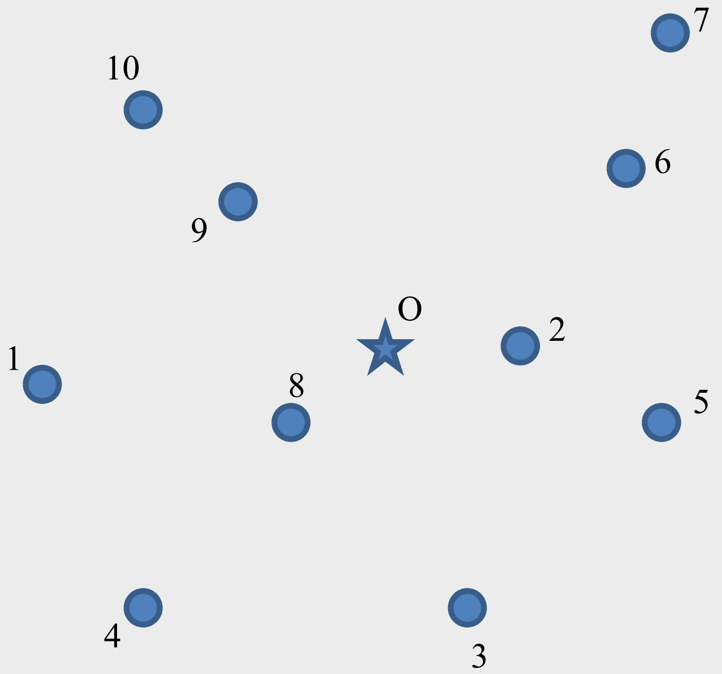


Heuristics

- Why do we use a heuristic method to solve a TSP?
  - The problem is difficult (known to be NP-Hard)
  - No polynomial time algorithm for solving it to optimality
  - Exponential in the number of cities
  - We must solve relatively "large" instances of the problem
- Heuristics aims to efficiently generate very good solutions. They do not find the optimal solution, or more precisely they do not guarantee the optimality of the found solutions.

TSP

Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0

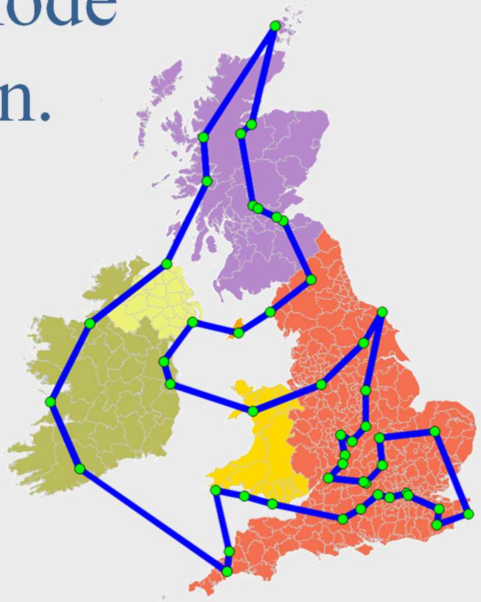


HEURISTICS

- Construction heuristics:
  - builds a solution from scratch (starting with nothing).
  - E.g. Nearest Neighbor Heuristic, and Greedy Heuristic
- Improvement heuristics (neighborhood search):
  - starts with a solution, and then tries to improve the solution, usually by making small changes in the current solution.
  - E.g. “2-opt”

TSP – Nearest Neighbor Heuristic

- Starting from an origin node, find the minimum distance required to visit each node once and only once and return to the origin.

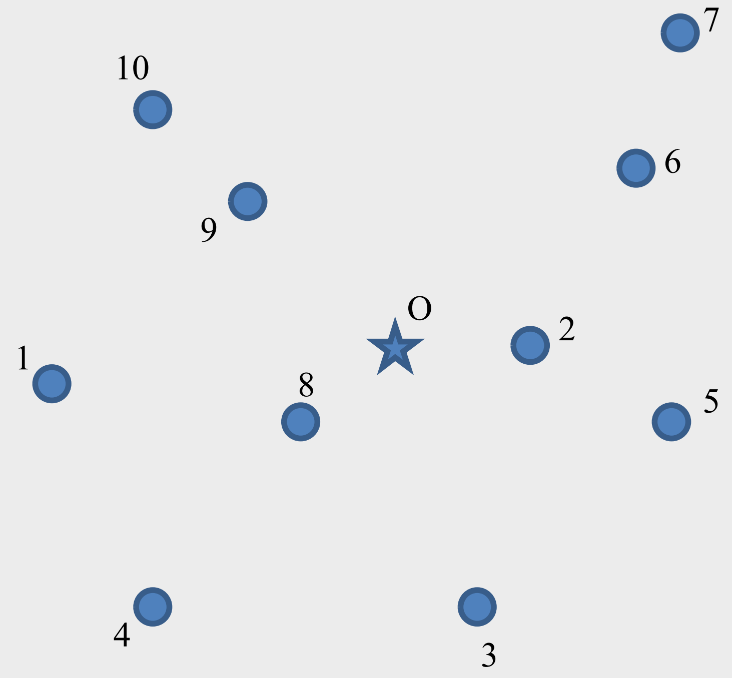


Nearest Neighbor Heuristic:

1. Select any node to be the active node
2. Connect the active node to the closest unconnected node, make that the new active node.
3. If there are more unconnected nodes go to step 2, otherwise connect to the starting node and end.

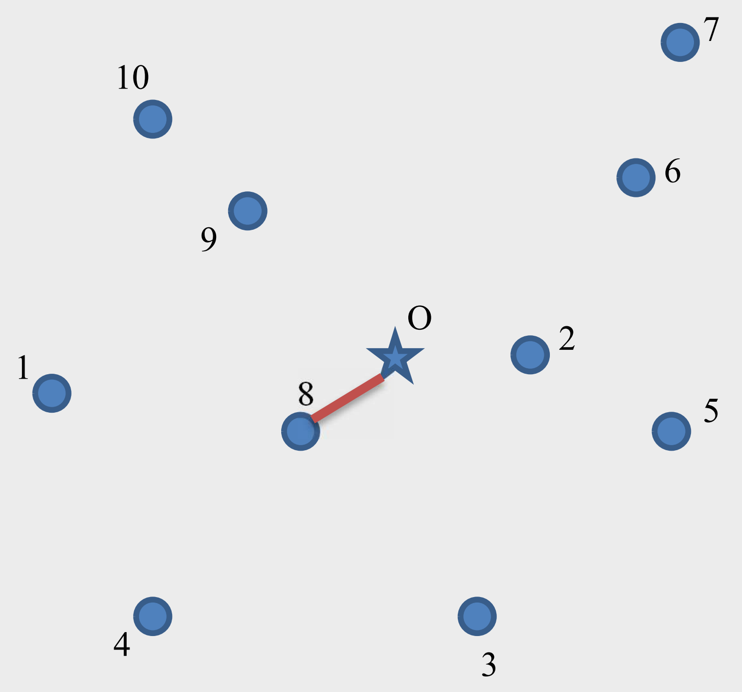
TSP – Nearest Neighbor Heuristic

Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



TSP – Nearest Neighbor Heuristic

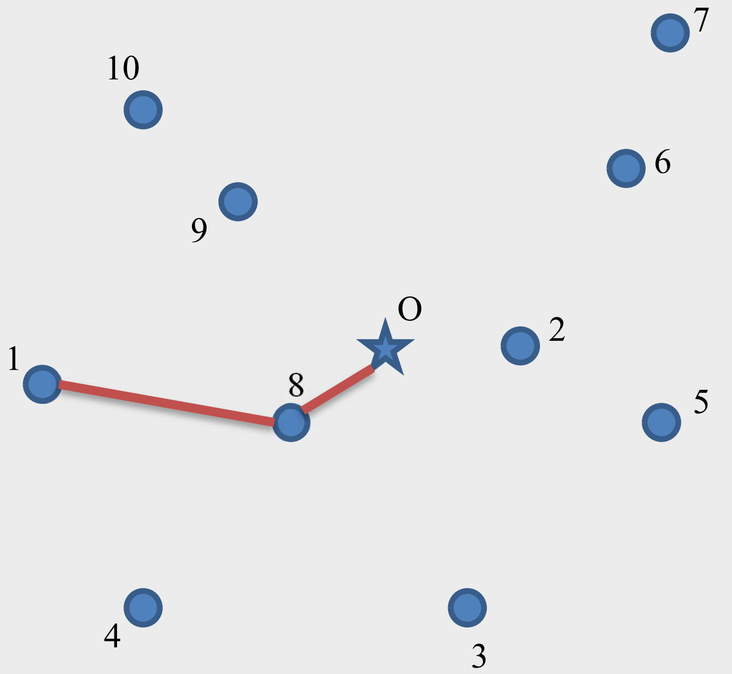
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	**7.1**	**3**	**7.3**	**8.6**	**6.3**	**7.1**	**10.8**	**2.8**	**5**	**8.6**	**0**



Tour: O-8

TSP – Nearest Neighbor Heuristic

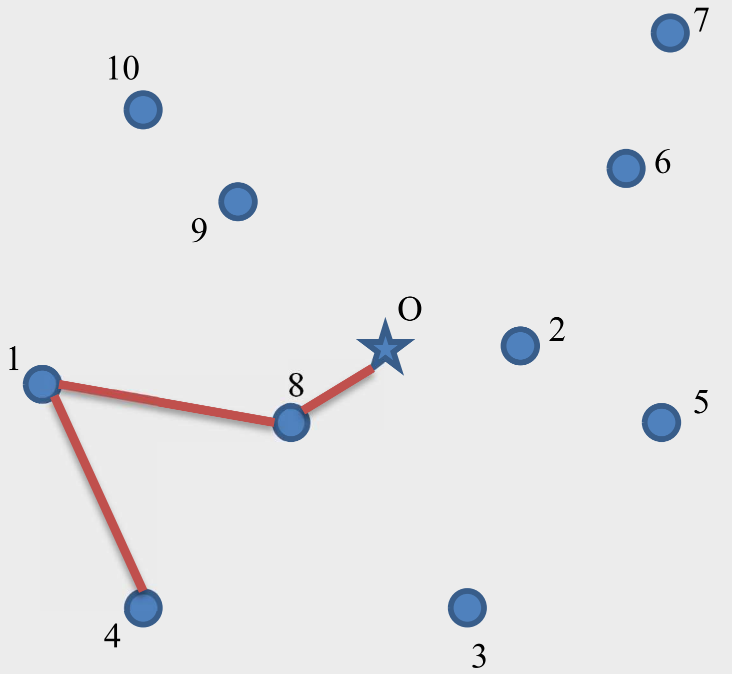
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1

TSP – Nearest Neighbor Heuristic

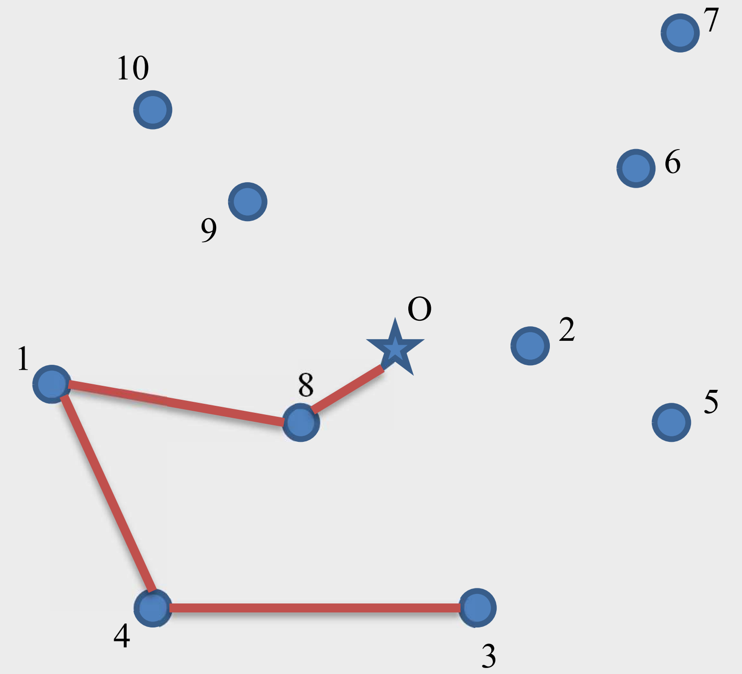
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1-4

TSP – Nearest Neighbor Heuristic

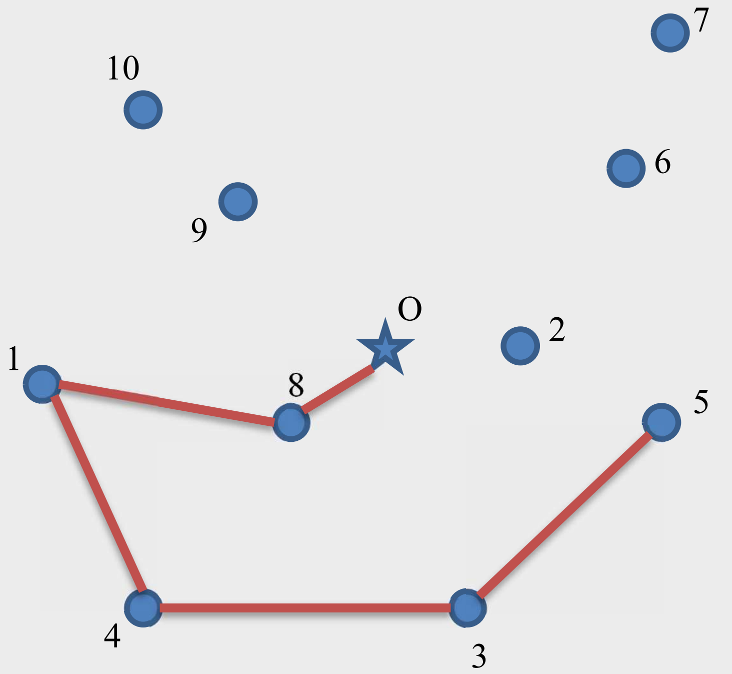
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1-4-3

TSP – Nearest Neighbor Heuristic

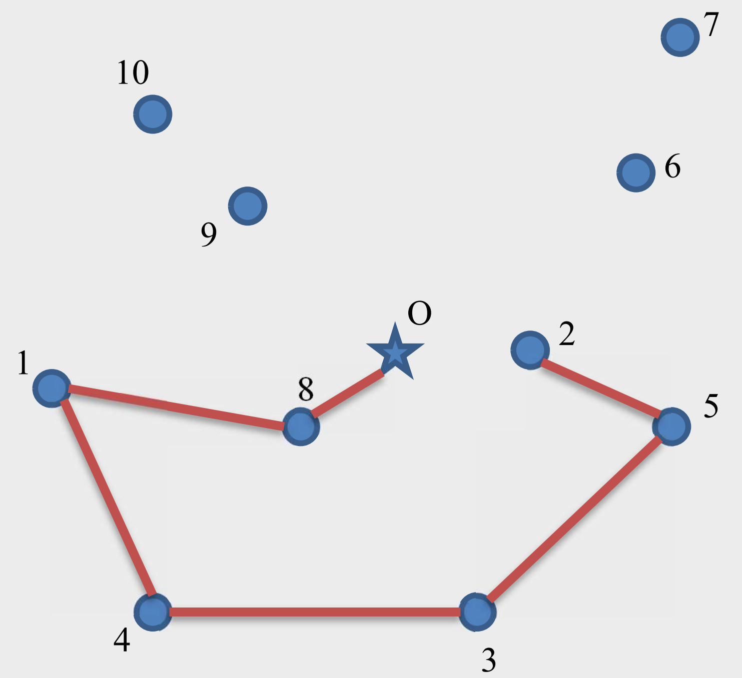
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1-4-3-5

TSP – Nearest Neighbor Heuristic

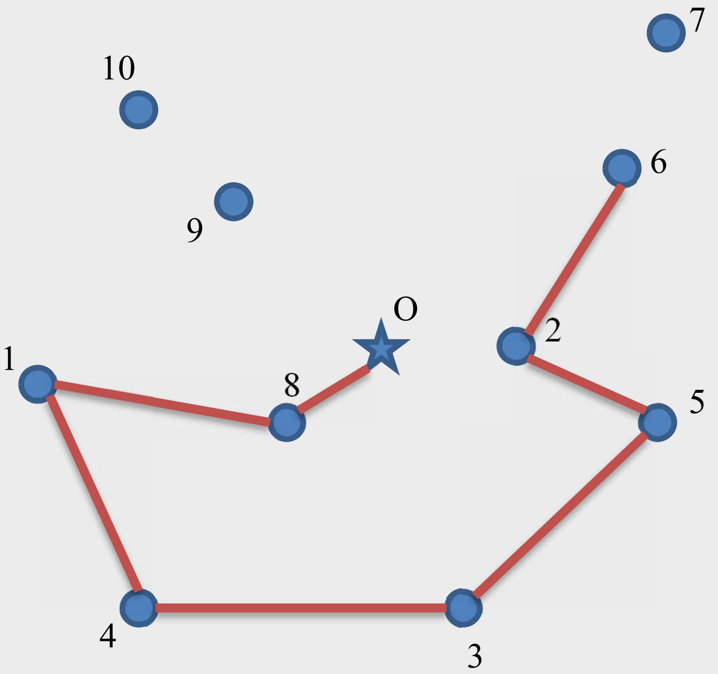
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1-4-3-5-2

TSP – Nearest Neighbor Heuristic

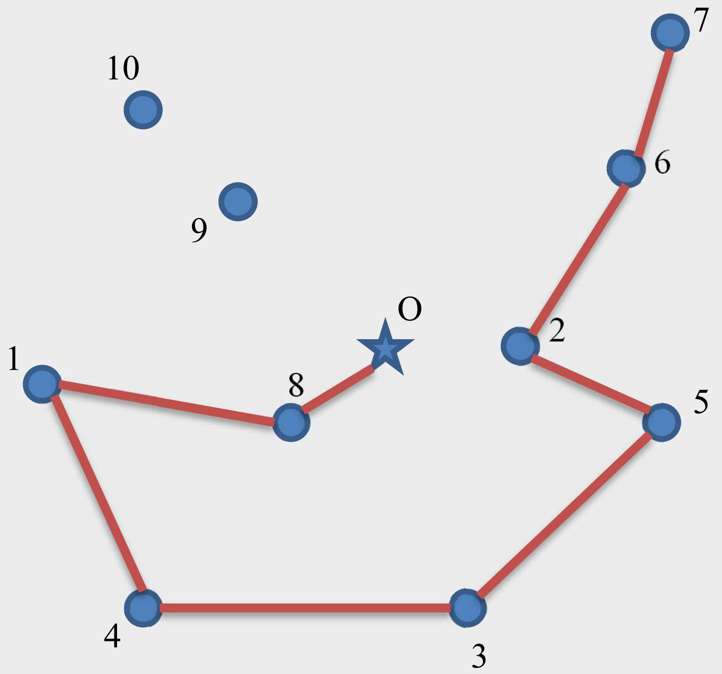
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1-4-3-5-2-6

TSP – Nearest Neighbor Heuristic

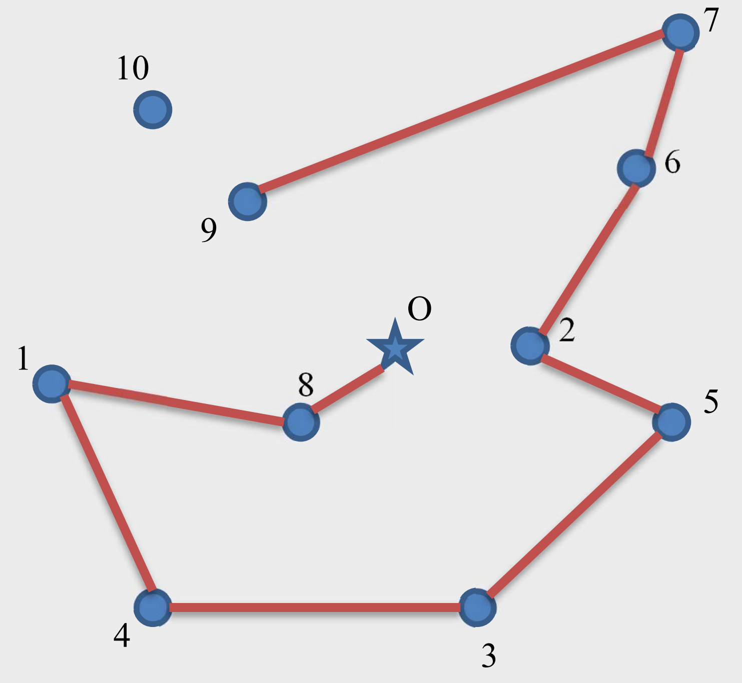
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1-4-3-5-2-6-7

TSP – Nearest Neighbor Heuristic

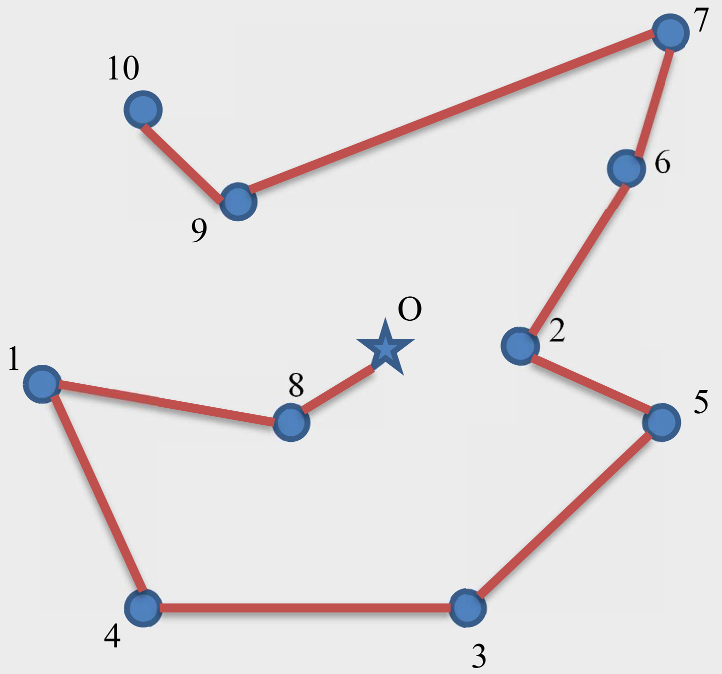
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1-4-3-5-2-6-7-9

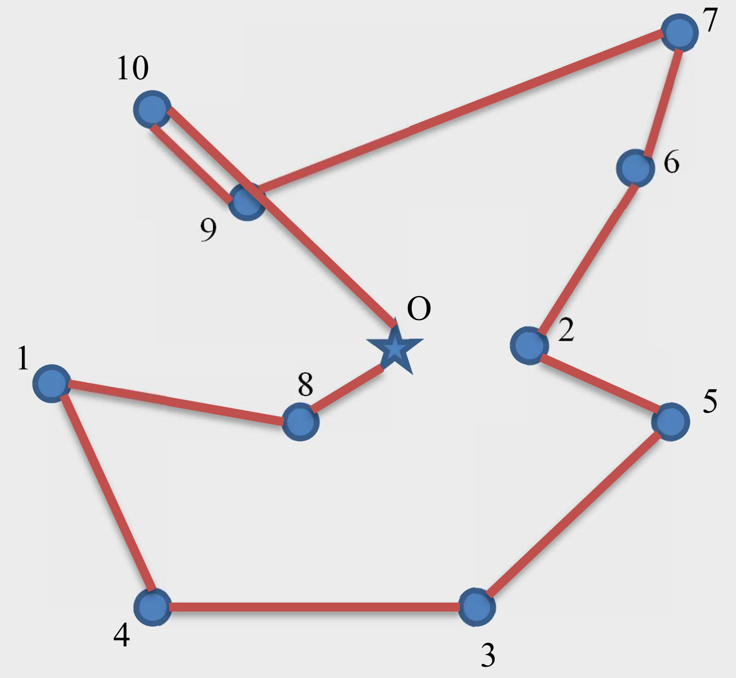
TSP – Nearest Neighbor Heuristic

Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1-4-3-5-2-6-7-9-10-O

Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1-4-3-5-2-6-7-9-10-O

Length: 63.2

Improvement Heuristic: 2-Opt

1. Identify pairs of arcs (*i*-*j* and *k*-*l*), where

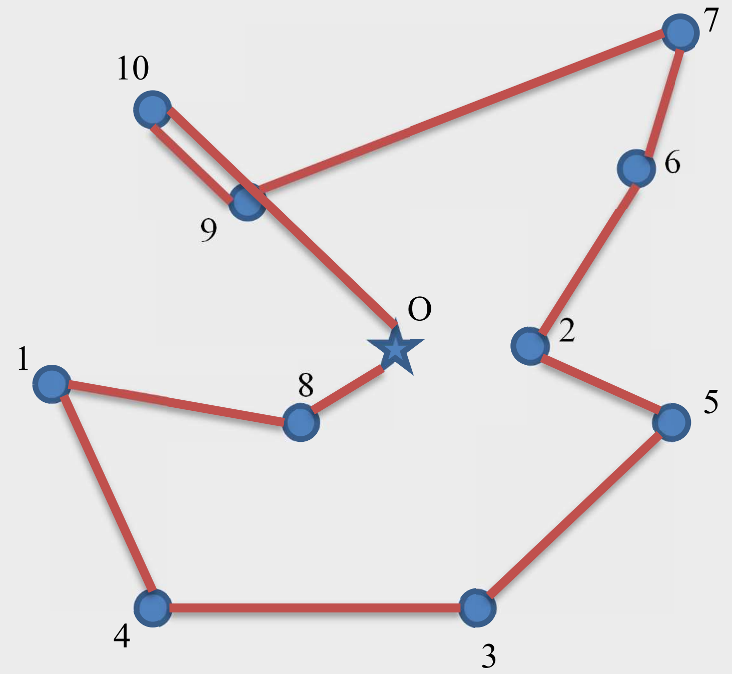
   $d(ij) + d(kl) > d(ik) + d(jl)$ (usually where they cross)

2. Select the pair with the largest difference, and re-connect the arcs (*i*-*k* and *j*-*l*)

3. Continue until there are no more crossed arcs.

TSP – 2-Opt

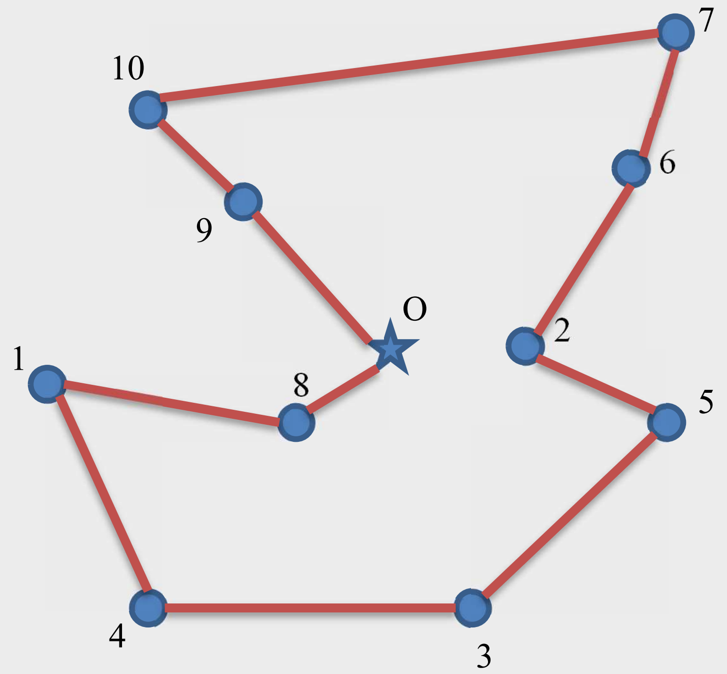
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



- Arcs 7-9 and 10-O cross
- d(79) + d(10-O) = 18.9 > d(7-10) + d(9-O) = 16.2
- Re-connect arcs 7-10 and 9-O

TSP – 2-OPT

Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0



Tour: O-8-1-4-3-5-2-6-7-10-9-O

Length: 60.5

Tour length reduces from 63.2 to 60.5

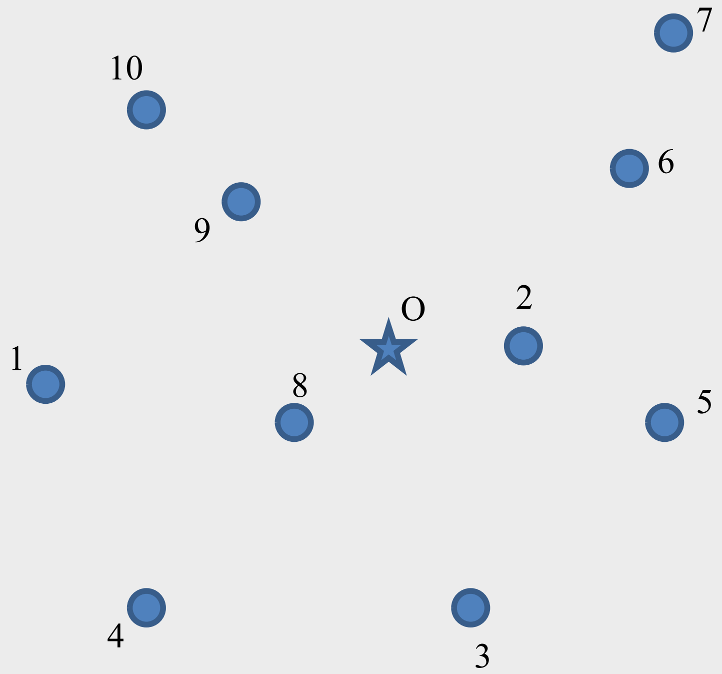
TSP – Greedy Heuristic

1. Sort all edges.
2. Select the shortest edge and add it to our tour if it doesn’t violate any of the constraints.
3. Do we have N edges in our tour? If no, repeat step 2.

TSP – Greedy Heuristic

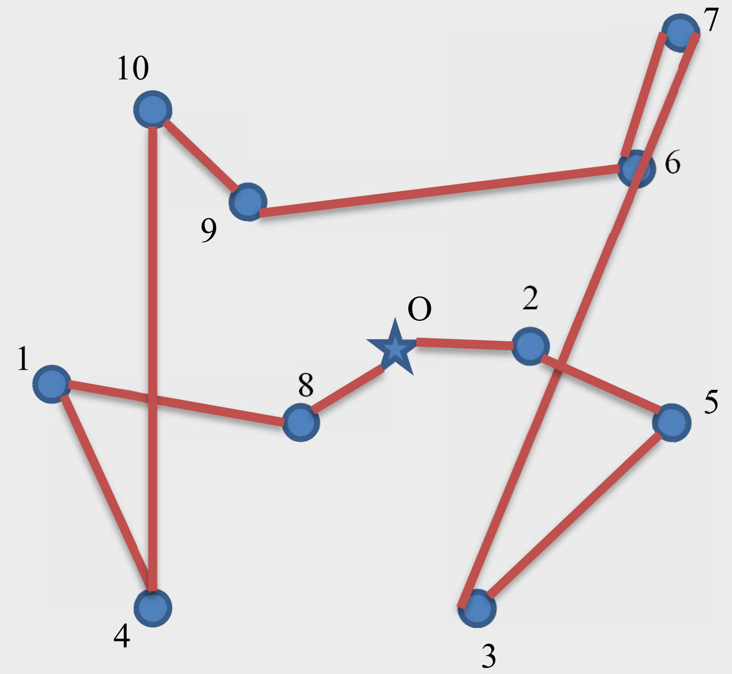
Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0

(i-j)	D_{ij}
(O-8)	2.8
(2-O)	3
(5-2)	3.6
(10-9)	3.6
(6-7)	4.1
(O-9)	5
(8-1)	5.1
(2-6)	5.4
(2-8)	5.4
(4-8)	5.8
(9-8)	6.1
(1-4)	6.3
(O-5)	6.3
(3-5)	6.4
(8-3)	6.4
(9-1)	6.4
(4-3)	7
(7-4)	19.4



TSP – Greedy Heuristic

Dis.	1	2	3	4	5	6	7	8	9	10	O
1	0	10	10.8	6.3	13	13.4	16.4	5.1	6.4	8.2	7.1
2	10	0	7.1	10.6	3.6	5.4	9.5	5.4	7.2	10.6	3
3	10.8	7.1	0	7	6.4	12.4	16.5	6.4	12.1	15.7	7.3
4	6.3	10.6	7	0	12.1	15.6	19.4	5.8	11.2	14	8.6
5	13	3.6	6.4	12.1	0	7.1	11	8	10.8	14.2	6.3
6	13.4	5.4	12.4	15.6	7.1	0	4.1	9.9	8.1	10.2	7.1
7	16.4	9.5	16.5	19.4	11	4.1	0	13.6	10.3	11.2	10.8
8	5.1	5.4	6.4	5.8	8	9.9	13.6	0	6.1	9.5	2.8
9	6.4	7.2	12.1	11.2	10.8	8.1	10.3	6.1	0	3.6	5
10	8.2	10.6	15.7	14	14.2	10.2	11.2	9.5	3.6	0	8.6
O	7.1	3	7.3	8.6	6.3	7.1	10.8	2.8	5	8.6	0

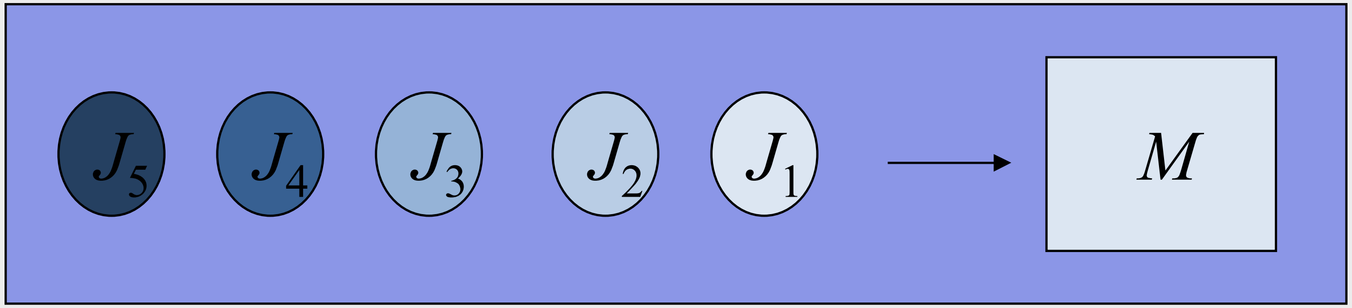


Tour: O-8-1-4-10-9-6-7-3-5-2-O

Length: 73.5

PRODUCTION SCHEDULING – SINGLE MACHINE

- ATM queue
- Small shops with one cashier



Example:

- 5 jobs
- Single machine

Job number	Processing Time	Due Date
1	11	61
2	29	45
3	31	31
4	1	33
5	2	32

Objectives

- The objective to be minimized is always a function of the completion times of the jobs (C_j), which, of course, depend on the schedule

- *Lateness:* $L_j = C_j - d_j$

Due date of job *j*

- *Tardiness:* $T_j = max\ (C_j - d_j, 0) = max\ (L_j, 0)$

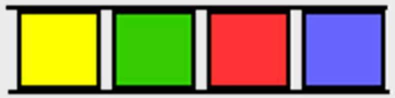
- *Unit Penalty:* $U_j = \begin{cases} 1 & \text{if } C_j > d_j \\ 0 & \text{otherwise} \end{cases}$

Objectives

- *Makespan* (C_{max}): completion time of the last job
- *Maximum Lateness* (L_{max}): worst violation of the due dates
- *Total completion time* ($\sum C_j$): flow time
- *Total tardiness* ($\sum T_j$)
- *Total number of tardy jobs* ($\sum U_j$)

Priority Sequencing Rules

- First come, first served (FCFS)
  - Queuing at airport



- Last come, first served (LCFS)
  - In a warehouse where items are stacked upwards, the unit on top is taken to fulfill an order

Priority Sequencing Rules

- Earliest due date (EDD)
  - The job with earliest due date is first, the one with the next earliest due date is second, and so on
- Shortest processing time (SPT)
  - The job with shortest processing time is first, the one with the next shortest processing time is second, and so on
- Longest processing time (LPT)
  - Multiprocessor scheduling in computer science

Example:

- 5 jobs
- Single machine

Job number	Processing Time	Due Date
1	11	61
2	29	45
3	31	31
4	1	33
5	2	32

Job number	Processing Time	Due Date
1	11	61
2	29	45
3	31	31
4	1	33
5	2	32

Job	Completion Time	Due Date	Tardiness
1	11	61	0
2	40	45	0
3	71	31	40
4	72	33	39
5	74	32	42
Total	268		121

First Come First Serve

Flow time
=268

Total tardiness
=121

No. of tardy jobs=3

Job number	Processing Time	Due Date
1	11	61
2	29	45
3	31	31
4	1	33
5	2	32

Shortest Processing Time

Job	Processing Time	Completion Time	Due Date	Tardiness
4	1	1	33	0
5	2	3	32	0
1	11	14	61	0
2	29	43	45	0
3	31	74	31	43
Total		135		43

Flow time =135

Total tardiness =43

No. of tardy jobs=1

PRIORITY SEQUENCING RULES

Job number	Processing Time	Due Date
1	11	61
2	29	45
3	31	31
4	1	33
5	2	32

Job	Processing Time	Completion Time	Due Date	Tardiness
3	31	31	31	0
5	2	33	32	1
4	1	34	33	1
2	29	63	45	18
1	11	74	61	13
Total		235		33

Earliest Due Date

Flow time
=235

Total tardiness
=33

No. of tardy jobs=4

Priority Sequencing Rules

Rule	Total in-process	Total Tardiness	Number of Tardy Jobs
FCFS	268	121	3
LCFS	176	34	3
EDD	235	33	4
SPT	135	43	1
LPT	309	107	4

FCFS: First come, first served

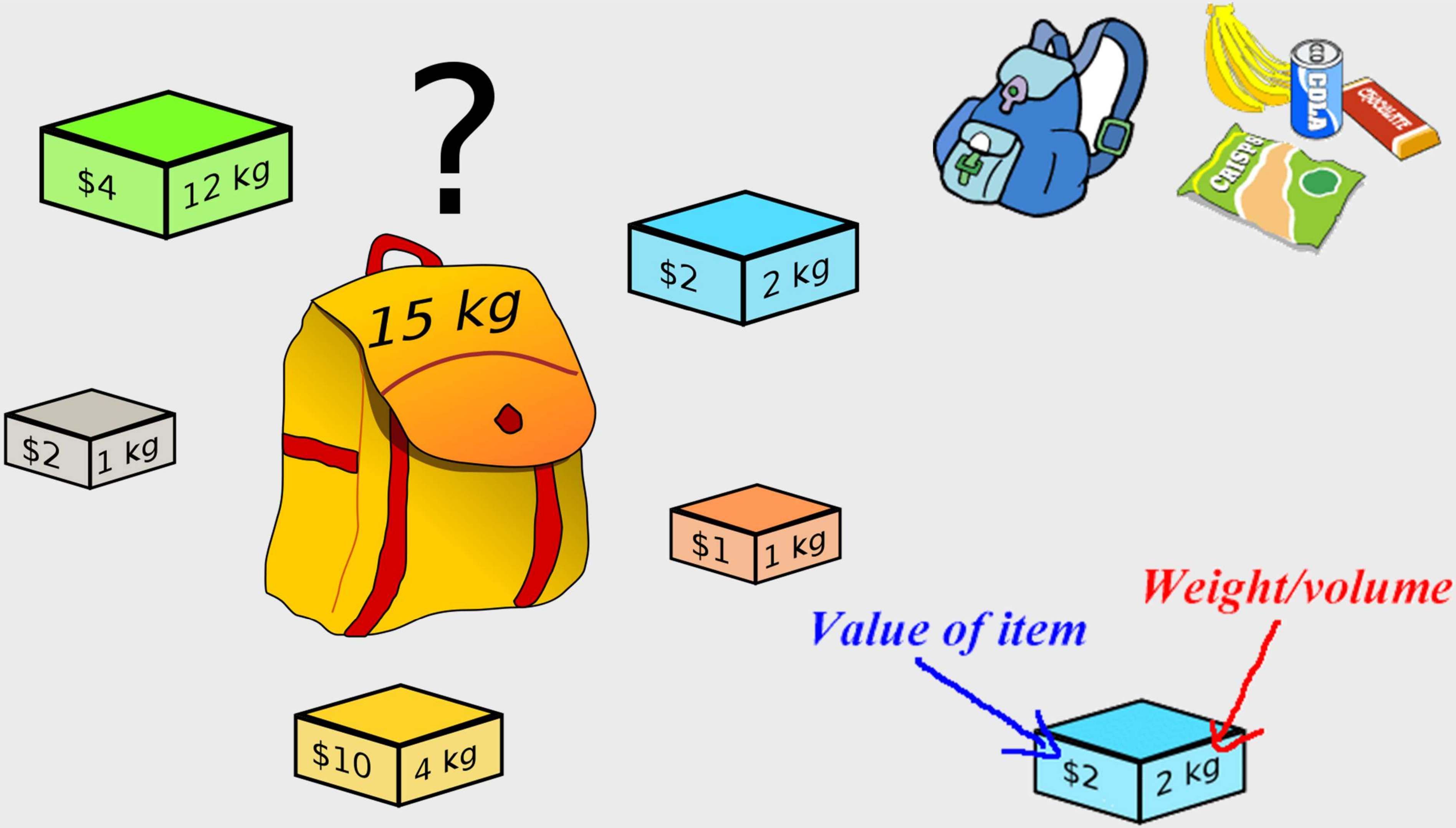
LCFS: Last come, first served

EDD: Earliest due date

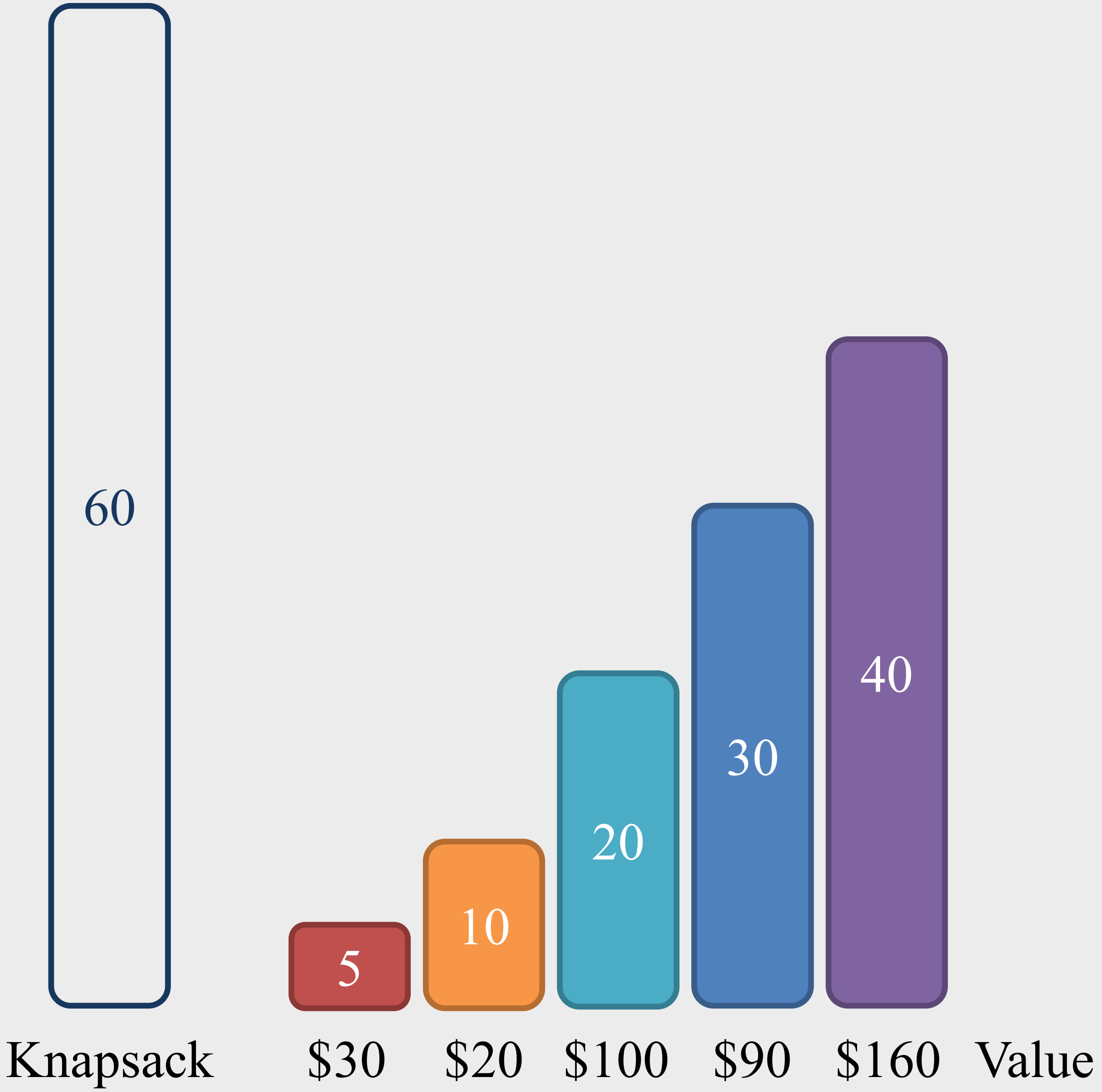
SPT: Shortest processing time

LPT: Longest processing time

Knapsack Problem



Knapsack Problem



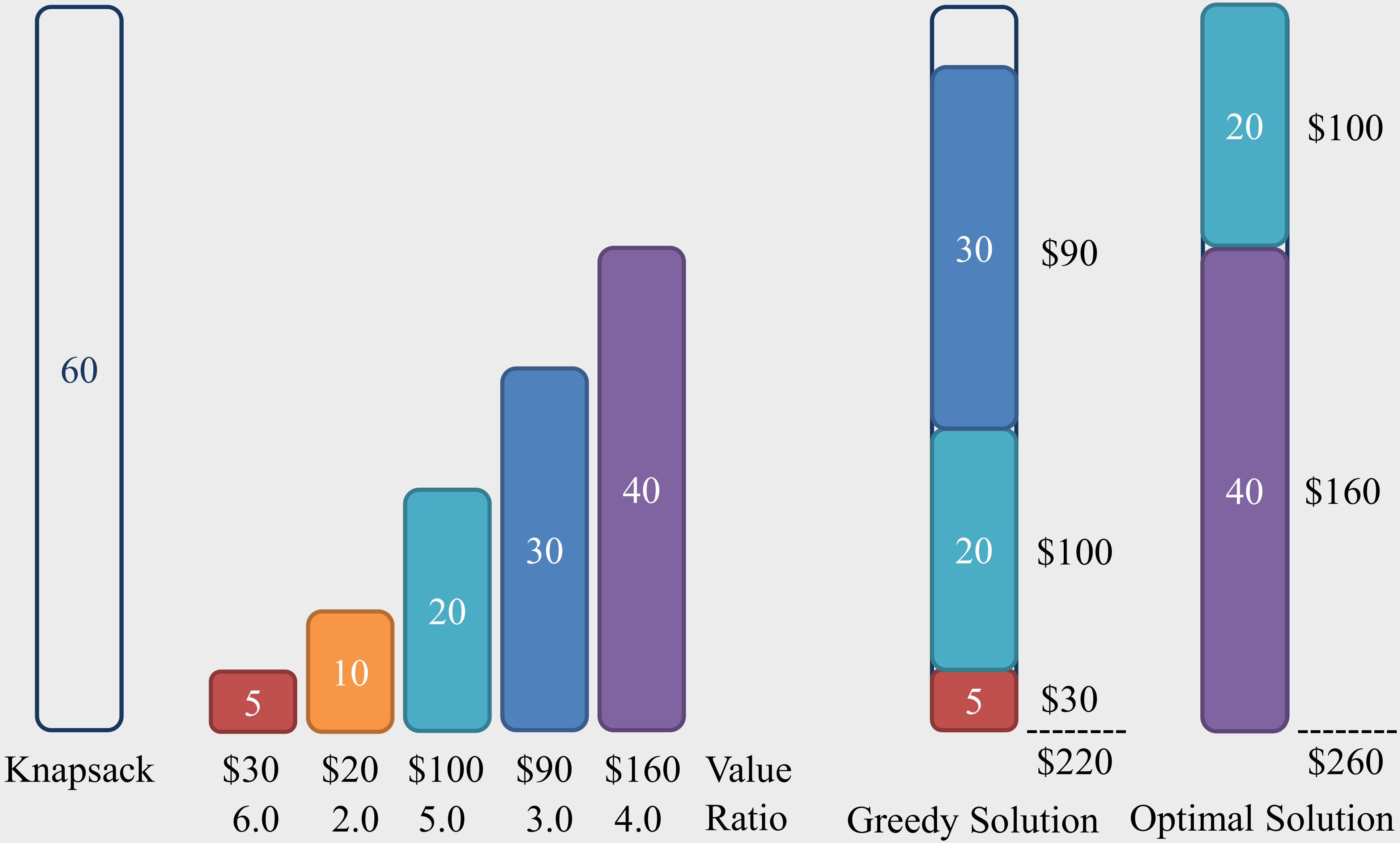
Knapsack Problem

- Greedy approximation algorithm

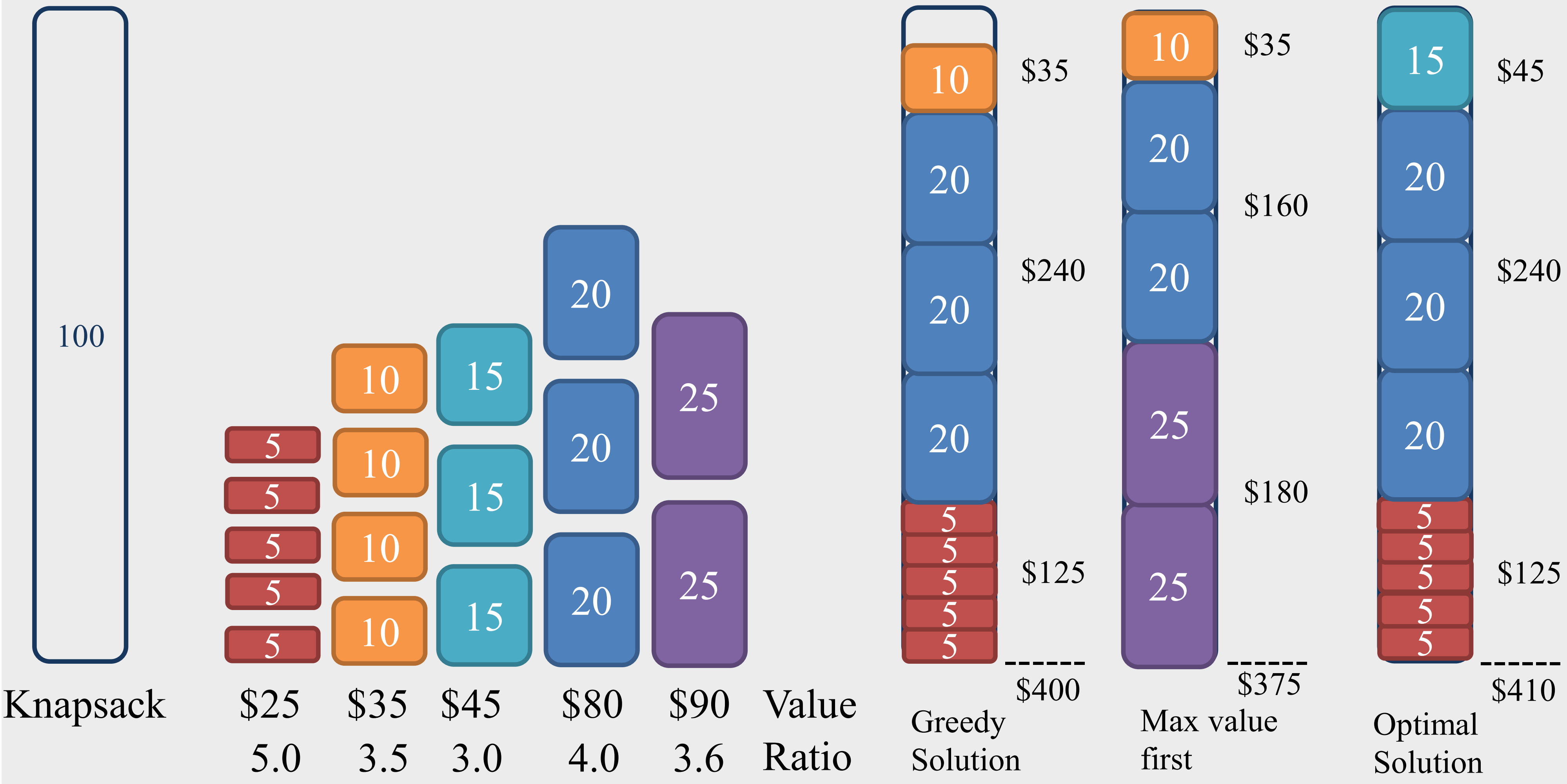
- Proposed by George Dantzig

1. Sort the items in decreasing order of value per unit of weight, V_i/W_i.
2. Insert them into the sack, starting with as many copies as possible of the first kind of item until there is no longer space in the sack for more.

Knapsack Problem



Knapsack Problem



LECTURE #3:

COMPONENTS OF METAHEURISTICS