07.01.19

Constructive heuristics

- Nearest neighbour heuristic
- Savings heuristic

Improvement heuristics

- K-opt (K = 2)

Implement these algos. (You may implement more algos if you want. These threes are mandatory)

Experiment

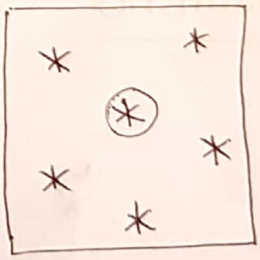
Chapter 3

Formulate to graph search problem

State space graph

Transition function

Chapter 4



Candidate sol^n.

x $N(x) = \{ \quad \}$

Neighbourhood function.

* For k-opt, when the number of members of neighborhood ~~search space~~ is too large, we may search filtered neighbor than all the neighbors.

Euclidean TSP

50

x y

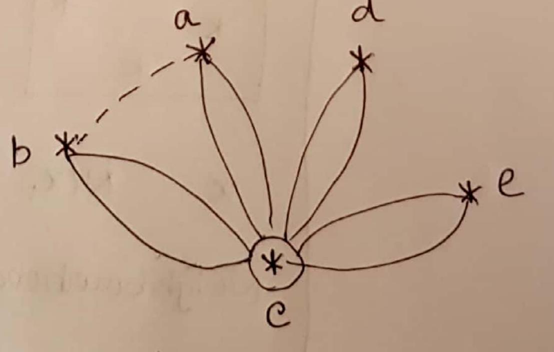
. .

. .

. .

* For nearest neighbor heuristic, experiment by choosing close neighbors than choosing only nearest neighbor.

Savings heuristics



tours

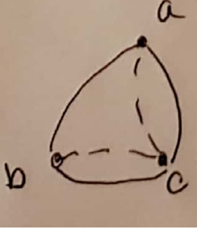
c – b – c

c – a – c

c – d – c

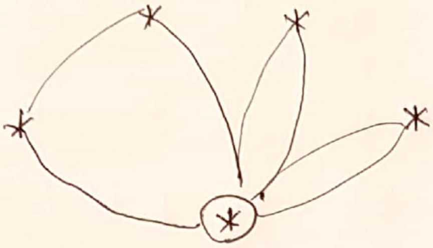
c – e – c

By merging ab,



savings are maximized,

$d_{ab} - (d_{ac} + d_{bc})$



* For savings heuristics, experiment by keeping top 3/5 savings instead of just the topmost.

Experiments

→ the solution found by constructive heuristics can be the starting point of improvement heuristics.

For nearest neighbor heuristics

→ Take top 3 neighbors instead of best one.

For 2-opt

→ Take top 10 or 20.. pairs instead of all pairs.