DM841

Discrete Optimization

Vehicle Routing
Construction Heuristics

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

Construction Heuristics for CVRP

- TSP based heuristics
- Saving heuristics (Clarke and Wright)
- Insertion heuristics
- Cluster-first route-second
  - Sweep algorithm
  - Generalized assignment
  - Location based heuristic
  - Petal algorithm
- Route-first cluster-second

Cluster-first route-second seems to perform better than route-first
(Note: distinction construction heuristic / iterative improvement
is often blurred)

Construction heuristics for TSP

They can be used for route-first cluster-second or for growing multiple tours subject to capacity constraints.

- Heuristics that Grow Fragments
  - Nearest neighborhood heuristics
  - Double-Ended Nearest Neighbor heuristic
  - Multiple Fragment heuristic (aka, greedy heuristic)
- Heuristics that Grow Tours
  - Nearest Addition
  - Farthest Addition
  - Random Addition
  - Nearest Insertion
  - Farthest Insertion
  - Random Insertion
  - Clarke-Wright saving heuristic
- Heuristics based on Trees
  - Minimum spanning tree heuristic
  - Christofides' heuristics

(But recall! Concorde: http://www.tsp.gatech.edu/)

[Bentley, 1992]

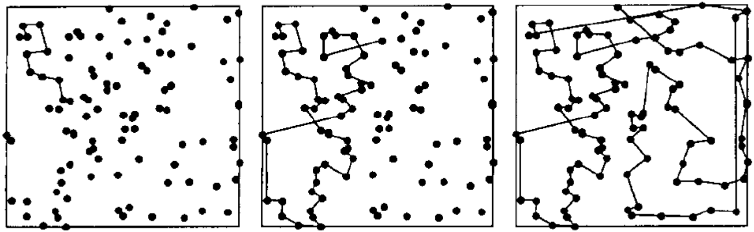


Figure 1. The Nearest Neighbor heuristic.

NN (Flood, 1956)

1. Randomly select a starting node
2. Add to the last node the closest node until no more nodes are available
3. Connect the last node with the first node

Running time $O(N^2)$

[Bentley, 1992]

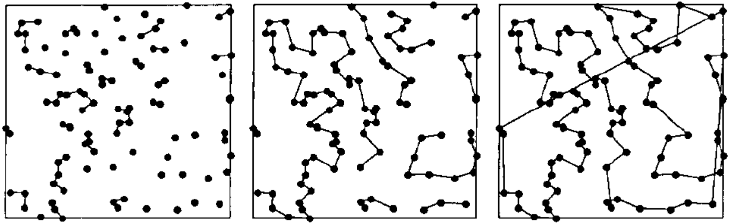


Figure 5. The Multiple Fragment heuristic.

Add the cheapest edge provided it does not create a cycle.

[Bentley, 1992]

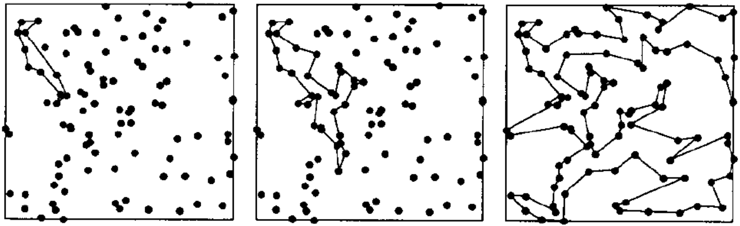


Figure 8. The Nearest Addition heuristic.

NA

1. Select a node and its closest node and build a tour of two nodes
2. Insert in the tour the closest node v until no more node are available

Running time $O(N^3)$

[Bentley, 1992]

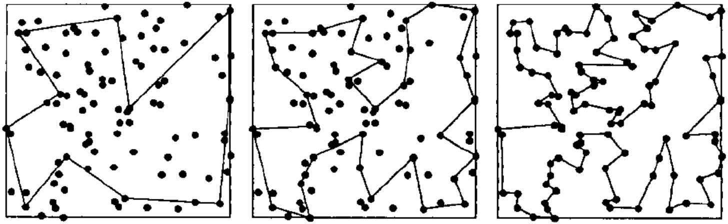


Figure 11. The Farthest Addition heuristic.

FA

1. Select a node and its farthest and build a tour of two nodes
2. Insert in the tour the farthest node v until no more node are available

FA is more effective than NA because the first few farthest points sketch a broad outline of the tour that is refined after.

Running time $O(N^3)$

[Bentley, 1992]

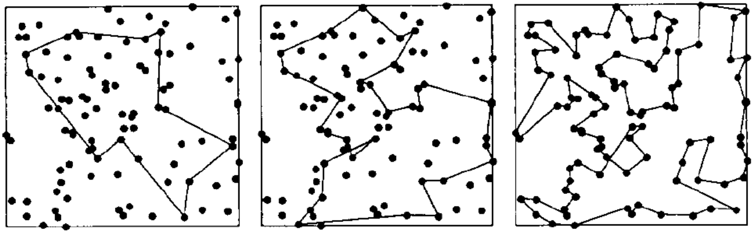


Figure 14. The Random Addition heuristic.

[Bentley, 1992]

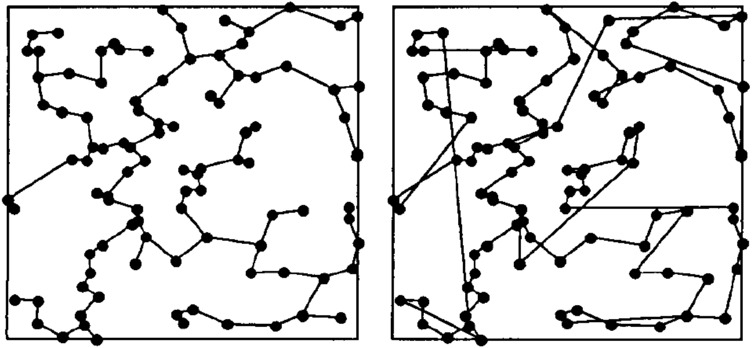


Figure 18. The Minimum Spanning Tree heuristic.

1. Find a minimum spanning tree $O(N^2)$
2. Append the nodes in the tour in a depth-first, inorder traversal

Running time $O(N^2)$ $\quad$ $A = MST(I)/OPT(I) \leq 2$

[Bentley, 1992]

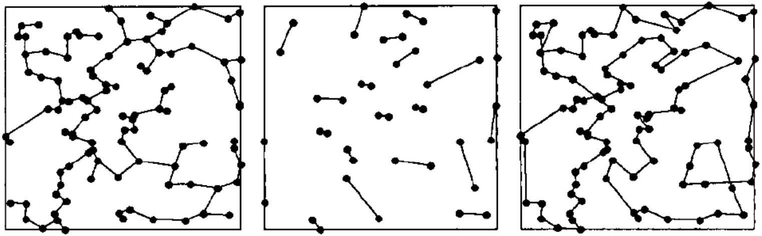
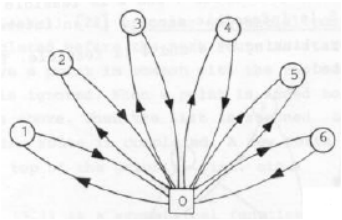


Figure 19. Christofides' heuristic.

1. Find the minimum spanning tree T. $O(N^2)$
2. Find nodes in T with odd degree and find the cheapest perfect matching M in the complete graph consisting of these nodes only. Let G be the multigraph of all nodes and edges in T and M. $O(N^3)$
3. Find an Eulerian walk (each node appears at least once and each edge exactly once) on G and an embedded tour. $O(N)$

Running time $O(N^3)$ $\qquad A = CH(I)/OPT(I) \leq 3/2$

Construction Heuristics Specific for VRP



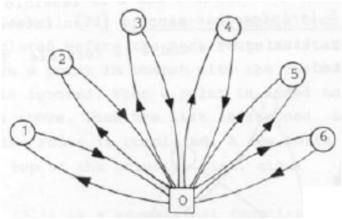
Clarke-Wright Saving Heuristic (1964)

1. Start with an initial allocation of one vehicle to each customer (0 is the depot for VRP or any chosen city for TSP)

Sequential:

2. consider in turn route $(0, i, \ldots, j, 0)$ determine s_{ki} and s_{jl}
3. merge with $(k, 0)$ or $(0, l)$

Construction Heuristics Specific for VRP

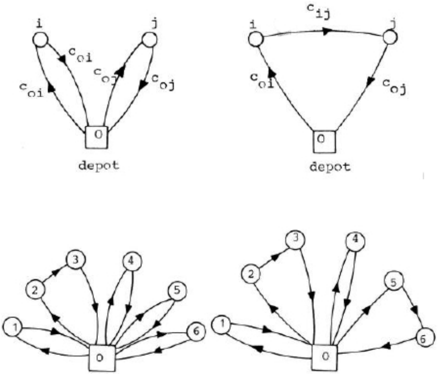


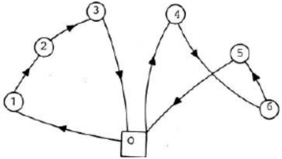
Clarke-Wright Saving Heuristic (1964)

1. Start with an initial allocation of one vehicle to each customer (0 is the depot for VRP or any chosen city for TSP)

Parallel:

2. Calculate saving $s_{ij} = c_{0i} + c_{0j} - c_{ij}$ and order the saving in non-increasing order
3. scan s_{ij}
   merge routes if i) i and j are not in the same tour ii) neither i and j are interior to an existing route iii) vehicle and time capacity are not exceeded


i
j
c_oi
c_oi
c_oj
c_oj
0
depot
c_ij
i
j
c_oi
c_oj
0
depot
1
2
3
4
5
6
0
1
2
3
4
5
6
0


1
2
3
4
5
6
0

(Fiala 1978)

Matching Based Saving Heuristic

1. Start with an initial allocation of one vehicle to each customer (0 is the depot for VRP or any chosen city for TSP)
2. Compute $s_{pq} = t(S_p) + t(S_q) - t(S_p \cup S_q)$ where $t(\cdot)$ is the TSP solution
3. Solve a max weighted matching on the sets S_k with weights s_{pq} on edges. A connection between a route p and q exists only if the merging is feasible.

Insertion Heuristic

$$\alpha(i, k, j) = c_{ik} + c_{kj} - \lambda c_{ij}$$

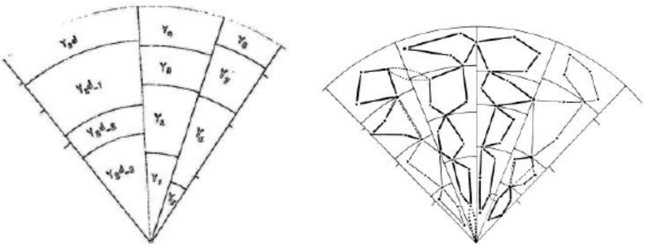
$$\beta(i, k, j) = \mu c_{0k} - \alpha(i, k, j)$$

1. construct emerging route $(0, k, 0)$
2. compute for all k unrouted the feasible insertion cost:

$$\alpha^*(i_k, k, j_k) = \min_p \{\alpha(i_p, k, i_{p+1})\}$$

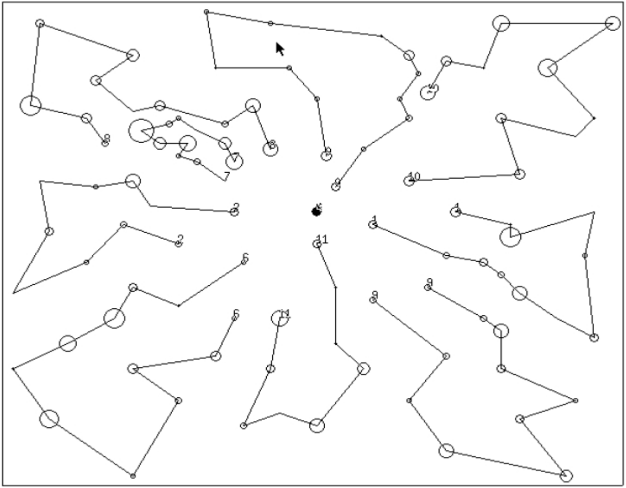
if no feasible insertion go to 1 otherwise choose k^* such that

$$\beta^*(i^*_k, k^*, j^*_k) = \max_k \{\beta(i_k, k, j_k)\}$$



Cluster-first route-second: **Sweep algorithm** [Wren & Holliday (1971)]

1. Choose i^* and set $\theta(i^*) = 0$ for the rotating ray
2. Compute and rank the polar coordinates (θ, ρ) of each point
3. Assign customers to vehicles until capacity not exceeded. If needed start a new route. Repeat until all customers scheduled.



Cluster-first route-second: **Generalized-assignment-based algorithm**
[Fisher & Jaikumur (1981)]

1. Choose a j_k at random for each route k
2. For each point compute

$$d_{ik} = \min\{c_{0,i} + c_{i,j_k} + c_{j_k,0}, c_{0j_k} + c_{j_k,i} + c_{i,0}\} - (c_{0,j_k} + c_{j_k,0})$$

3. Solve GAP with d_{ik}, Q and q_i

$$\begin{aligned} \min \quad & \textstyle\sum_i \sum_j d_{ij_k} x_{ij_k} \\ & \textstyle\sum_i q_i x_{ij_k} \leq Q \\ & \textstyle\sum_{j_k} x_{ij_k} \geq 1 \\ & x_{ij_k} \in \{0, 1\} \end{aligned}$$

Cluster-first route-second: **Location based heuristic**
[Bramel & Simchi-Levi (1995)]

1. Determine seeds by solving a capacitated location problem (k-median)

2. Assign customers to closest seed

(better performance than insertion and saving heuristics)

Cluster-first route-second: **Petal Algorithm**

1. Construct a subset of feasible routes

2. Solve a set partitioning problem

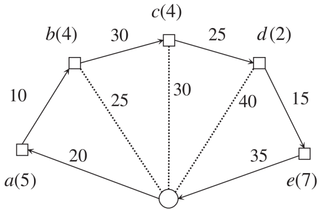
Route-first cluster-second [Beasley, 1983]

1. Construct a TSP tour (giant tour) over all customers
2. Split the giant tour. Idea:
   - Choose an arbitrary orientation of the TSP;
   - Partition the tour according to capacity constraint;
   - Repeat for several orientations and select the best

   Alternatively: use the optimal split algorithm of next slide.

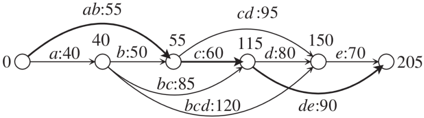
(not very competitive if alone but competitive if inside an Evolutionary Algorithm [Prins, 2004])

- From the TSP tour,

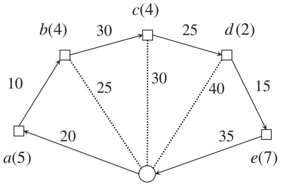


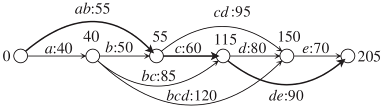
Assume $Q = 10$

- construct an auxiliary graph $H = (X, A, Z)$.
  $X = \{0, \ldots, n\}$,
  $A = \{(i, j) \mid i < j$ trip visiting customers $i + 1$ to j is feasible in terms of capacity$\}$,
  $z_{ij} = c_{0,i+1} + l_{i+1,j} + c_{0j}$, where $l_{i+1,j}$ is the cost of traveling from $i + 1$ to j in the TSP tour.



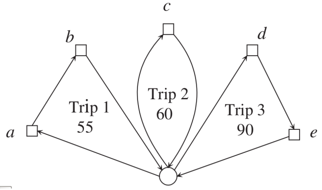
One arc ab with weight 55 for the trip $(0, a, b, 0)$





- An optimal CVRP solution given the tour corresponds to a min-cost path from 0 to n in H. Computed in $O(n)$ since H is circuitless.

- The resulting CVRP solution with three routes



Exercise

Which heuristics can be used to minimize K
and which ones need to have K fixed a priori?