The Travelling Salesman

Marietjie Venter

The Setting

- A salesman wants to sell his product in a number of cities.
- He wants to visit each city exactly once and then return to the city where he started off.
- The cities can be visited in any order.

The Problem

- Find the order in which the salesman has to visit the cities so that the total travelling distance is a minimum, called the optimal tour.
- Very simple ☺

The Solution

- Not so simple ☹
- Optimal solution is of O(n^x)
- Optimization problem:
  - Find the best solution you can in the given time, even though it might not be the optimal solution.

Always Remember

- Keep track of the best solution so far.
- Keep track runtime so far (if there is a time limit)
- Give the best solution so far when time runs out.

Possible Approaches

- Good old random!
  - Generate random solutions.
  - Keep track of the best solution so far.
  - Give the best solution so far when time runs out.
- Techniques can be used to improve the solution (discussed later).

Nearest Neighbour

- Construct the tour by going from each city to the closest unvisited city until all the cities have been visited.
- Some cities can be “forgotten” only to have to be inserted later at high cost to the solution.
- (Greedy algorithm)

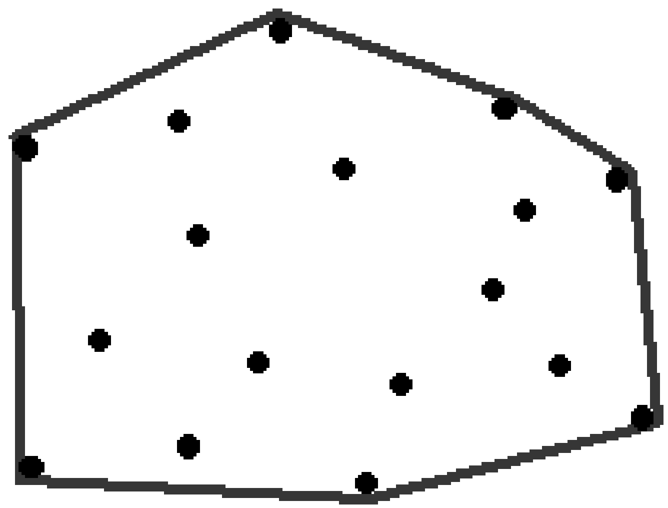
Insertion Heuristics

- Start with a subtour.
- Keep adding cities until all the cities are included.
- Things to consider:
  - Choice of starting subtour.
  - How to choose the next node (city) to insert in the tour.
  - Where to insert it.

Choice of Subtour

- Typically 3 cities, e.g. the 3 cities that form the largest triangle.
- Very good option: the tour that forms the *convex hull* of all the nodes (cities).

Convex Hull



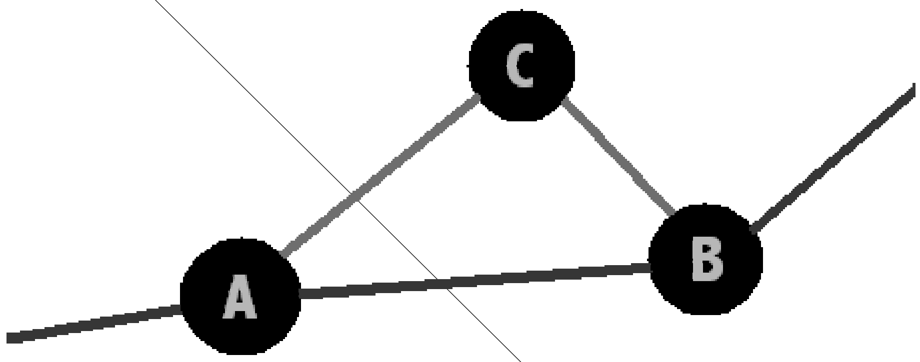
– Each dot represents a city.

– Red “ring” illustrates the convex hull.

- Tour is convex.
- All the cities fall inside the ring.
- As if you wrap an elastic band around all the cities.

Cheapest Insertion

- Each time insert the city which causes the lowest increase in total distance.



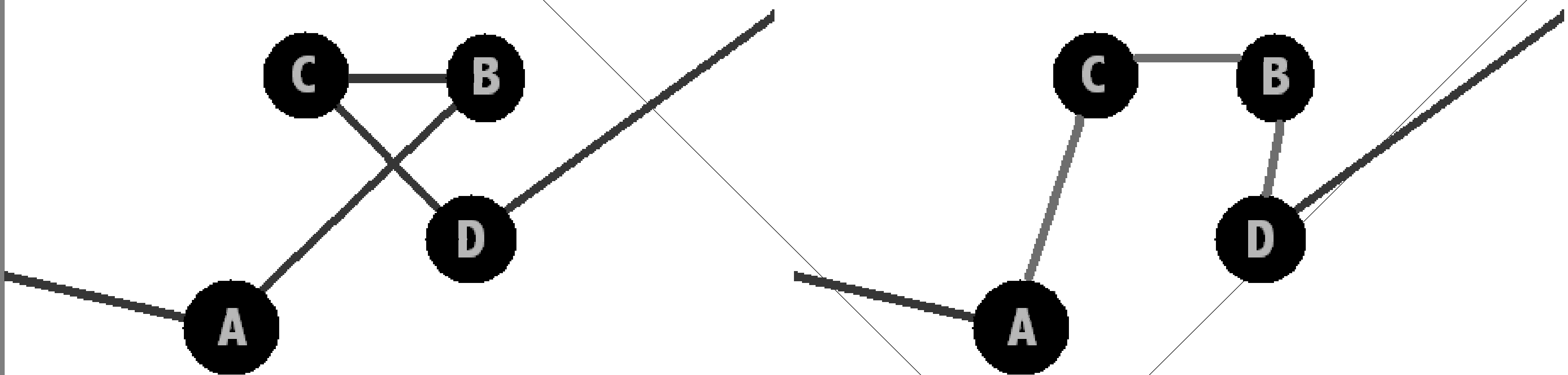
- ((dist AC + dist CB) – dist AB) is a minimum.
- (Greedy algorithm)

Farthest Insertion

- Insert the city of which its closest distance to the existing tour is a maximum.
- The idea is to fix the overall layout of the tour as soon as possible.

Improving Solutions

- Exchange: change the order in which 2 cities occur in the tour and check if this decreases the total distance.



Improving Solutions

- Genetic Programming:
  - Mutation
    - Randomly alter the tour to see if a better one can be found.
  - Selective "breeding"
    - Take two good solutions and combine them to see if a better one can be constructed.

The Travelling Salesman

- Nearest Neighbour
- Insertion Heuristics
  - Convex hull
  - Cheapest insertion
  - Farthest insertion
- Improving solutions