James Keane S1973227

IADS Coursework 3

Part C – Your Own Algorithm

The algorithm that I have implemented is the cheapest insertion algorithm, its known as a construction heuristic. Essentially the idea of a construction heuristic is to progressively construct a sub tour of the network. This is done by repetitively selecting a node and inserting it into the most optimal position in the sub tour.

The cheapest insertion does this by being as 'greedy' as possible. At each iteration the algorithm selects the closest node to any of nodes in the subtour and then places it at the position that would give the lowest cost.

The algorithm has 3 key stages:

- Initialisation
    - The algorithm randomly selects a node and inserts it into an empty sub tour
    - It then selects the node that is closest to the first node in the sub tour
- Selection
    - The algorithm selects nodes by choosing an unused node that is the closest to any of the nodes in the subtour
- Insertion
    - The algorithm then chooses inserts the selected node at a position in the sub tour that yields the lowest cost journey

The algorithm will repeat selection and insertion until all nodes in the network have been selected.

The algorithm has a runtime of O(n^2 log(n)) as for each node selection, there is a sort. For each insertion, there is an array insertion and deletion.

I found this algorithm from a paper that analyses several heuristic methods for the travelling salesman problem. The reference and a link to the paper can be found below.

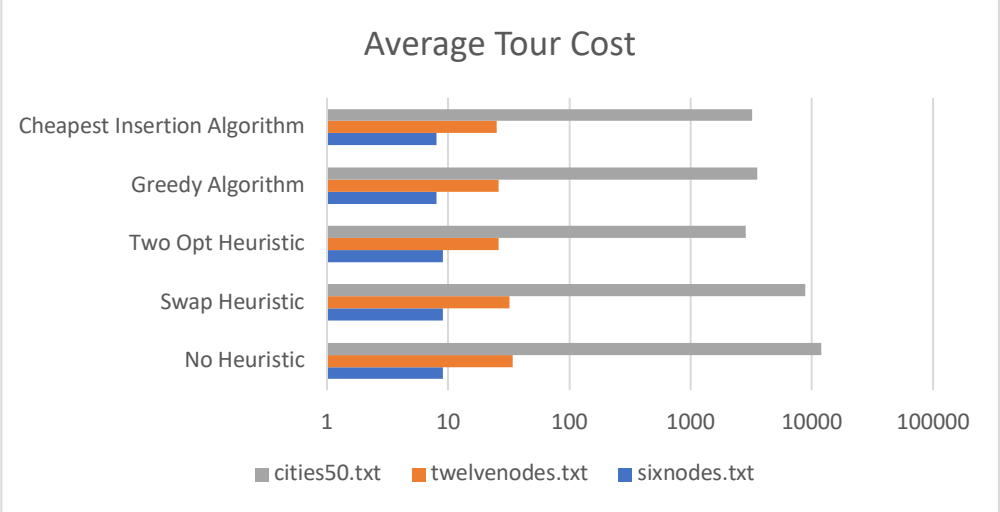
Rosenkrantz, Daniel & Stearns, Richard & II, Philip. (1977). An Analysis of Several Heuristics for the Traveling Salesman Problem. SIAM J. Comput.. 6. 563-581. 10.1137/0206041.

https://www.researchgate.net/publication/220616869_An_Analysis_of_Several_Heuristics_for_the_Traveling_Salesman_Problem

James Keane S1973227

Part D – Experiments

The first experiment was to find which algorithm was the best at finding the lowest tour costs across the data provided, 'sixnodes.txt', 'twelvenodes.txt' and 'cities50.txt'. For each heuristic used in Part B and my own implementation of the Cheapest Insertion Algorithm in Part C, I ran 100 iterations of each heuristic using each data set provided. As a control I also ran each dataset without a heuristic.
As both Swap and 2-Opt heuristics take a value k which limits the number of sweeps, I decided that I would set k to be -1 which means that both algorithms continue to sweep until no improvement is made. Below is a table and a graph showing the results from this experiment.

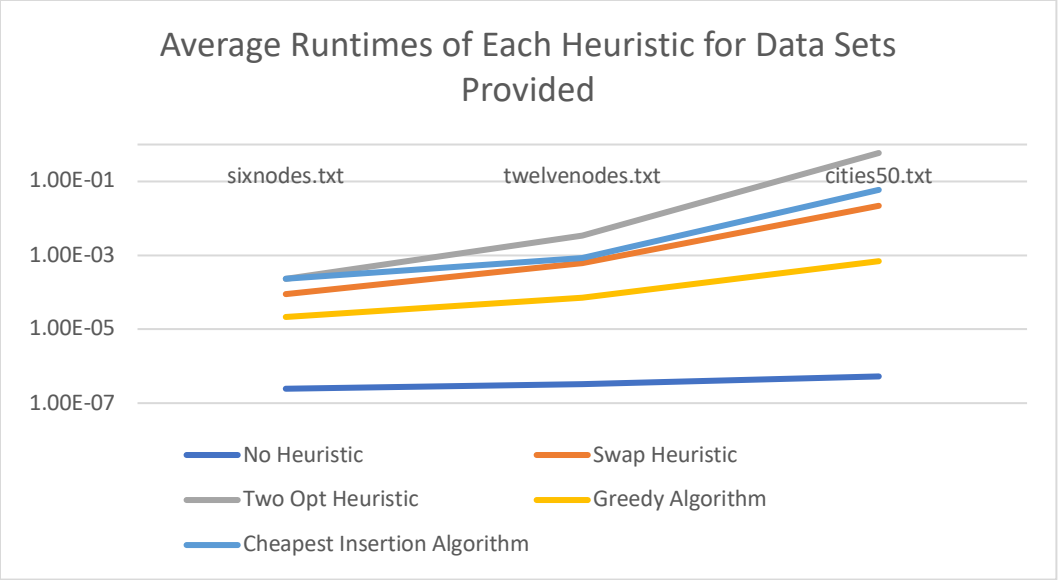


Heuristics	sixnodes.txt	twelvenodes.txt	cities50.txt
No Heuristic	9	34	11907.243
Swap Heuristic	9	32	8805.297
Two Opt Heuristic	9	26	2856.639
Greedy Algorithm	8	26	3523.348
Cheapest Insertion Algorithm	8	25.18	3203.669

From this we can conclude that the cheapest insertion algorithm is the most optimal for smaller networks of nodes, however when the number of nodes becomes approx. 50, the Two Opt Heuristic becomes the better performing algorithm for calculating the most optimal tour.

The second experiment I conducted was to find the runtimes of each heuristic for each data set. This is to compare how each heuristic performs as the data set becomes larger. For this experiment I ran 100 iterations of each heuristic on each data set, collected the time taken for each iteration and calculated an average. I included the runtimes for each data set without a heuristic as a control. Below is a table and a graph illustrating the results.

James Keane S1973227



Algorithms	sixnodes.txt	twelvenodes.txt	cities50.txt
No Heuristic	2.46E-07s	3.24E-07s	5.29E-07s
Swap Heuristic	8.93E-05s	0.00062504s	0.02196057s
Two Opt Heuristic	0.00023129s	0.00343651s	0.58997968s
Greedy Algorithm	2.15E-05s	7.25E-05s	0.00069597s
Cheapest Insertion Algorithm	2.33E-04s	0.00083136s	0.0592544s

From our results we see that although 2 opt was gave the best tour cost for the cities50 dataset it also had the worst runtime by a large amount, the axis is logarithmic. The best overall performing algorithm is really between Greedy and Cheapest Insertion as both have similar performance. Where Greedy might shortfall in having a less optimal tour cost it makes up for it in better runtime.