Traveling Salesman Problem

Vladimir Marin - 322CA

No Institute Given

1 Problem description

1.1 Question asked and abstraction

"Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?"

The easiest way for TSP to be modeled is as an undirected weighted graph. Each city equates to a node, and each road/path represents an edge with its cost analogue to its real-life counterpart distance. The problem is ultimately a minimisation problem, starting and finishing at a specified node, having visited each other nodes only once. Often times, the ideal model consists of a complete graph.

1.2 Importance

TSP is representative of an entire class of problems known as "combinatorial optimisation problems". Specifically, should an efficient algorithm for TSP be found, then the same algorithm could be used to solved a large range of NP-complete problems

1.3 Real life applications

The Traveling Salesman Problem and its abstractions are relevant in a large number of fields ranging from logistics, microchip manufacturing and planning, to something like DNA sampling in genetic biology, neural networks and astronomy.

2 Algorithms

2.1 Heuristic Method - Nearest Neighbour

This approach is a greedy algorithm. It states that we begin with the desired node and then visit the nearest unvisited node (the least cost between source and destination). It's time complexity is quadratic: $O(n^2)$.

The algorythm goes as follows:
1) Select a random city n and set is as the starting city n0 a random city n and set is as the starting city n0
2) Find the nearest unvisited city and go there
3) Mark the current city as visited
4) Are there any unvisited cities? If yes, go to (2)
5) Return to the starting city

It is a sub optimal method, but perhaps the first one that comes to mind when trying to solve TSP.

2.2 Exact method - Held–Karp algorithm

This algorithm is a dynamic programming algorithm which takes as input a cost matrix between a set of nodes, and the output is the minimum-length path that visits each node exactly once, and returns to the starting node. It's time complexity is exponential: $O(2^n)$.

Notation: 1: starting city, S: list of nodes visited, $e! = 1$: city not contained in S, $g(S, e)$: distance. Suppose $S = s_1, ..., s_k$ is a set of k nodes. For $i = 1; i <= k; i++$, $S_i = S \div s_i = s_1, ..., s_i - 1, s_i + 1, ..., s_k$. The shortest path from 1 through S has s_i as its penultimate node, then removing the final edge from this path must result in the shortest path from 1 to s_i through S_i, meaning there are only k candidate shortest paths from 1 to e through S.
The Held-Karp bound is the relaxed solution to the linear programming of the TSP. Usually, HK lower bound is about 0.8% below the optimal tour length.

3 Evaluation criteria

Each algorithm will be simulated using MATLAB and there will be 10 size scenarios, from a 10 node graph to an 100 node graph. Every graph will span over an area of 1000 square units. The criteria used to compare the two methods will be the optimal route length and the elapsed time.

4 Conclusion

Algorithm	**Optimal Route Length (units)**	**Elapsed Time (sec.)**
Nearest Neighbour	653.57	1.5
Held - Karp	643.85	30

Table 1. Average performance

Although the Nearest Neighbour approach returns a sub optimal result, every path being longer than its Held-Karp counterpart, it is, of course, a lot faster, with a time complexity of O(n). Held-Karp's biggest disadvantage stands in its inefficiency when it comes to larger inputs, its shortcomings becoming apparent as soon as a 20 node graph.

References

1. Arora, Sanjeev (1998), bluehttp://graphics.stanford.edu/courses/cs468-06-winter/Papers/arora-tsp.pdf/.
2. Stem Lounge, bluehttps://stemlounge.com/animated-algorithms-for-the-traveling-salesman-problem/.
3. MathWorks, bluehttps://www.mathworks.com/matlabcentral/fileexchange/25542-nearest-neighbor-algorithm-for-the-travelling-salesman-problem
4. MathWorks, bluehttps://www.mathworks.com/matlabcentral/fileexchange/31454-dynamic-programming-solution-to-the-tsp
5. International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-4 Issue-6, August 2015, bluehttps://www.ijeat.org/wp-content/uploads/papers/v4i6/F4173084615.pdf.