Algorithm

In this part, I use the Christofides algorithm [1] to solve the Travelling salesman problem.

Introduction

The Christofides algorithm was first discovered by Nicos Christofides in 1976. This approximation algorithm could guarantee that its solution will be within a factor of 3/2 of optimal solution length and the running time will be polynomial.

How it works

Consider the n-city TSP problem defined in completed graph G = (V, A) where V is the set of vertices and A is the set of edges between vertices. Distances between vertices are stored in the two-dimensional matrix **dists[i][j]**. The vertices should satisfy the triangle inequality that for every three vertices, u, v and x, $w(uv) + w(vx) \geqslant w(ux)$. Function w is the distance between two vertices.

There are five procedures in this algorithm, I will go through them one by one.

1. Find minimum spanning tree T of G. I use prim's algorithm to find the minimum spanning tree in my implementation. The time complexity in my code is O(n^3).

2. Let O be the set of vertices with odd degree in T. Then find the minimum distance perfect matching M from O. Blossom Algorithm [2] can find the minimum-weight perfect matching for this case, but I do not understand how it works after all. So, I decided to use the greedy search as the approximation. The time complexity of the greedy search is O(n^2), but it might not give the best minimum-weight matching.

3. Combine minimum spanning tree T and the perfect matching M to form a graph H in which each vertex has even degree. This procedure only costs linear time.

4. Since each vertex has even degree in graph H, then it can generate the Eulerian circuit in H in which the trail starts and ends on the same vertex and visits every edge in H exactly once. I use Hierholzer's algorithm [3] to form the Eulerian circuit. The time complexity of the algorithm I write is O(n^2)

5. Finally, we generate a Hamiltonian circuit that visited each vertex once. We need to remove the duplication in the Eulerian circuit. Thanks to the triangle inequality, the distance after shortcutting will less than original path. Then we can get the total

distance of the TSP problem. The time complexity of removing duplication is O(n).

Overall, the time complexity of the whole Christofides algorithm is O(n^3) in which n is the number of vertices.

Worst-case Analysis

The original Christofides algorithm could guarantee that the ratio of the answer obtained to the optimum TSP solution is strictly less than 3/2. However, in my implementation, I use approximation to find the minimum-weight perfect matching, so the ratio might higher than 3/2. But we will discuss it later in the experiment.

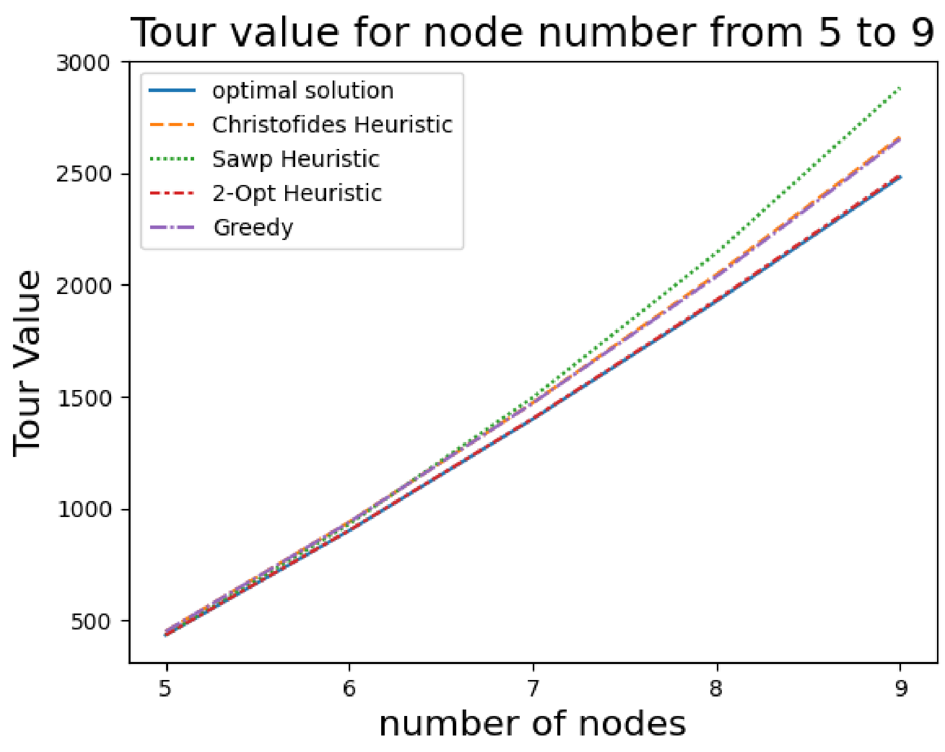
Experiment

Introduction

In tests.py. I generate three tests for the algorithm. The first one is using permutation to compute an exact solution for small number of nodes and compare the result with different heuristics. The second and the third experiments are using the datasets A280 and ATT48 from **TSPLIB** [4] with given best solution.

First Experiment

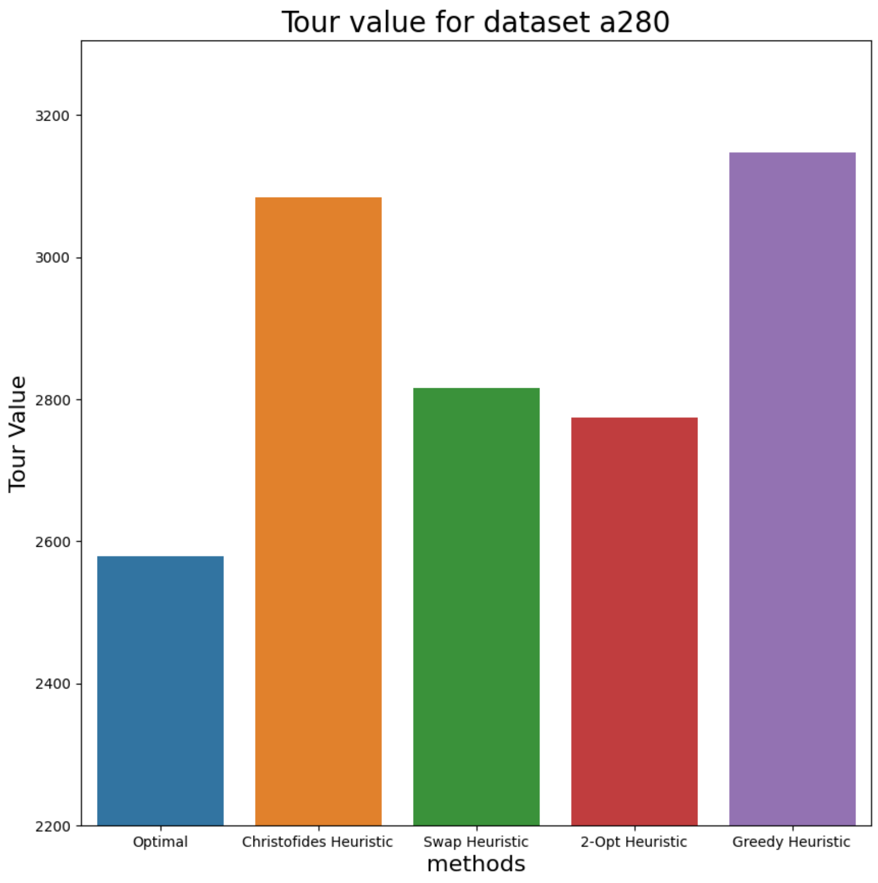
The idea of the first experiment is to test the performance of different heuristics in small number of nodes. The method I use in tests.py is **test_5_to_9_nodes**, this method iterates from 5 to 9 node, generates 100 datasets for each node. I will collect the average tour value using different heuristics. Then I convert the data into csv and print it out using seaborn library. From this line plot, we can see that the differences are not obvious in nodes less than 6. The gap becomes larger as nodes increase. When number of nodes equals to 9, the 2-Opt heuristic still close to the optimal solution, but Christofides heuristic and greedy heuristic is less performative than 2-Opt. The swap Heuristic is the worst-case Here. The differences between optimal were calculated for the most recent generate data.

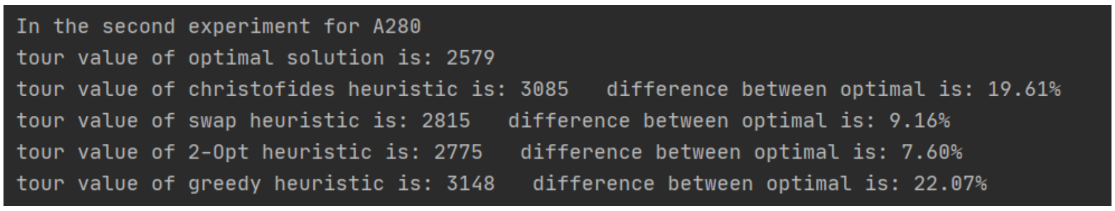


```
When nodes = 9,
tour value of optimal solution is: 2483
tour value of christofides heuristic is: 2662   difference between optimal is: 7.22%
tour value of swap heuristic is: 2884   difference between optimal is: 16.14%
tour value of 2-Opt heuristic is: 2492   difference between optimal is: 0.36%
tour value of greedy heuristic is: 2653   difference between optimal is: 6.84%
```

Second Experiment

The dataset of the second experiment is a Euclidean setting with 280 nodes, I compare the tour value and generate the graph. From the bar chart, we can see that Christofides heuristic and Greedy heuristic have large differences between optimal, the calculation shows that they are both about 20% difference between optimal. The Swap heuristic and 2-Opt heuristic has better performance for less than 10% differences.

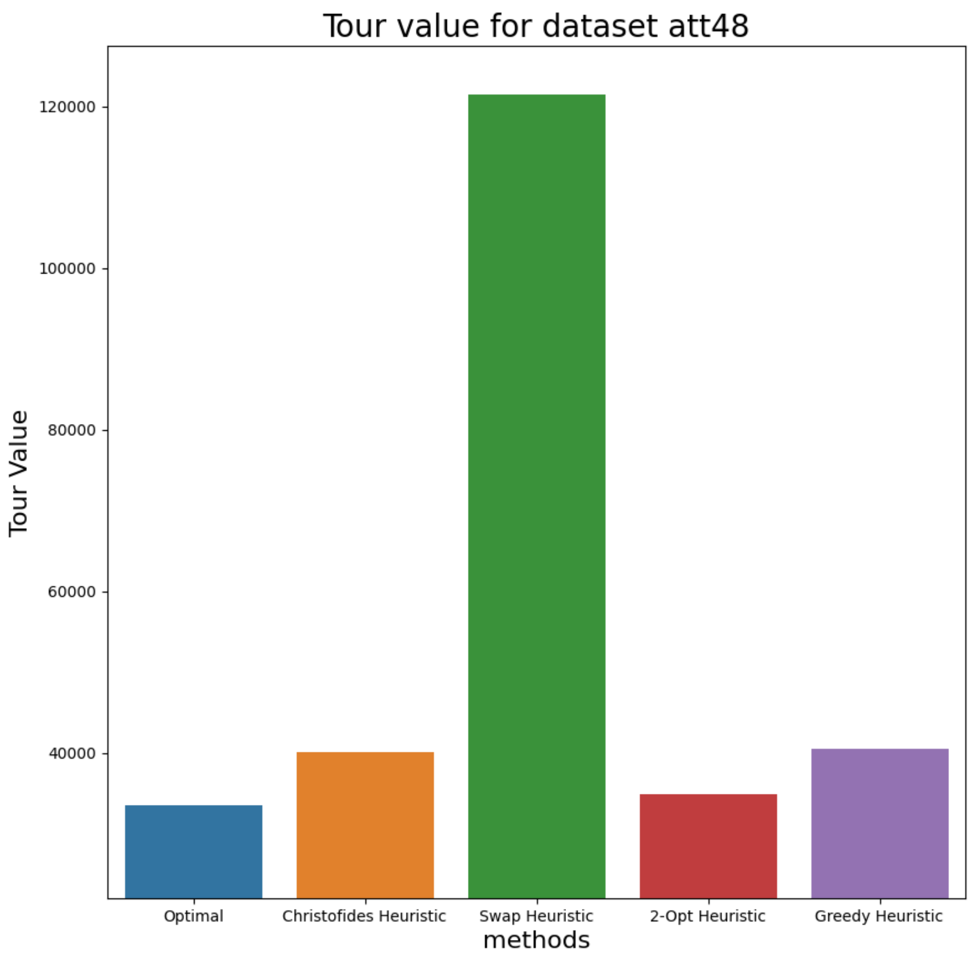


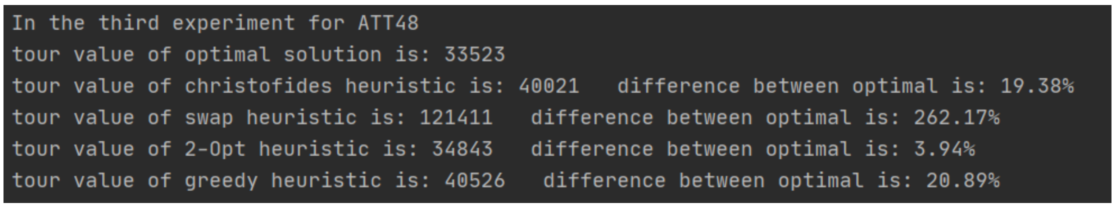


```
In the second experiment for A280
tour value of optimal solution is: 2579
tour value of christofides heuristic is: 3085   difference between optimal is: 19.61%
tour value of swap heuristic is: 2815   difference between optimal is: 9.16%
tour value of 2-Opt heuristic is: 2775   difference between optimal is: 7.60%
tour value of greedy heuristic is: 3148   difference between optimal is: 22.07%
```

Third Experiment

The dataset of the third experiment is a set of 48 cities (US state capitals) in Euclidean setting. I also generate the bar chart and calculate the differences. From the graph, we can see that the tour value of Swap heuristic is much higher than any other which is 262.17% difference between the optimal solution. Due to the outlier, we can not see clearly of the difference between Christofides, 2-Opt and Greedy. We can obtain the differences from the calculation. Once again, the performance of Christofides and Greedy are both around 20% difference between optimal. Surprisingly, 2-Opt heuristic only have 3.94% difference compares to the optimal which is great.





```
In the third experiment for ATT48
tour value of optimal solution is: 33523
tour value of christofides heuristic is: 40021   difference between optimal is: 19.38%
tour value of swap heuristic is: 121411   difference between optimal is: 262.17%
tour value of 2-Opt heuristic is: 34843   difference between optimal is: 3.94%
tour value of greedy heuristic is: 40526   difference between optimal is: 20.89%
```

Conclusion

From these three experiments, we can conclude that the 2-opt heuristic has the best performance within these four heuristics. Theoretically, the Christofides heuristic should have the lower bound of 3/2 Opt which means has better performance than the 2-Opt heuristic. The reason for unsatisfying in my experiment is I use the approximation to find the minimum-cost perfect matching.

Reference

[1] CHRISTOFIDES, N. WORST-CASE ANALYSIS OF A NEW HEURISTIC FOR THE TRAVELLING SALESMAN PROBLEM, 1976

[2] Blossom Algorithm https://en.wikipedia.org/wiki/Blossom_algorithm

[3] Hierholzer's algorithm https://slaystudy.com/hierholzers-algorithm/

[4] TSPLIB http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html