CSCI203

Assignment 4

Student name: Weicheng Yin

Student number: 5009819 | Email ID: wy157

Algorithm description:

In this assignment, I used two algorithms to solve the TSM problem. The first algorithm is the greedy algorithm, the second is the famous branch and bound algorithm (with breath first traversal).

Greedy algorithm:

In this algorithm, initially, the program will start from node 1. Then it will find the minimum cost travel to the city which hasn't been visited before (e.g. let's assume node 2 is the next city need to be visited). Now from node 2, the program will still find the minimum cost travel to the city which hasn't been visited before. Iterate for the number of city numbers, then the program will get the answer. And this result will be used in the next algorithm as the initial upper bound

Branch and bound algorithm:

In this algorithm, initially the algorithm will use the result calculated by the greedy algorithm as the initial upper bound. Then using calculate the minimum cost for each row of the 2-D array which be discussed later in the data structure. And add them together as the initial lower bound.

Then I will use breath first traversal to expand the path. The node which will be expend is in the top of the priority queue (this could make the algorithm more efficient). The path with the lowest lower bound will always at the top the priority queue.

Data structure:

I will use the 2-D array to store the distance information. And in the branch and bound algorithm, I will have a structure which contains: the current city index of the path, the next city index of the path, the actually cost by now, one array to store the information whether the city is visited or not and the lower bound. And the branch and bound algorithm will use priority queue to store the path information to improve the efficiency of the algorithm.

Result:

This is the result use the Australia_flights.txt as the cost data set.

1. Greedy algorithm:

Number of cities: 11

Tour:

1 2 6 4 5 7 8 10 11 3 9

Wollongong

Sydney

Canberra

Melbourne

Adelaide

Alice Springs

Darwin

Cairns

Gold Coast
Brisbane
Perth
Wollongong
Total cost: 24.6

2. Branch and bound algorithm (Breath-first):
Number of cities: 11
Upper bound: 24.6
Lower bound: 15.04
Optimal tour:
1 3 11 10 8 7 9 5 4 6 2
Wollongong
Brisbane
Gold Coast
Cairns
Darwin
Alice Springs
Perth
Adelaide
Melbourne
Canberra
Sydney
Wollongong
Total cost: 19.75

This is the result using the Australia_roads.txt as the cost data set:
1. Greedy algorithm:
Number of cities: 11
Tour:
1 2 6 4 5 7 8 10 3 11 9
Wollongong
Sydney
Canberra
Melbourne
Adelaide
Alice Springs
Darwin
Cairns
Brisbane
Gold Coast
Perth
Wollongong
Total cost: 17637

2. Branch and bound algorithm (Breath-first):
Number of cities: 11

Upper bound: 17637
Lower bound: 9338
Optimal tour:
1 2 11 3 10 8 7 9 5 4 6
Wollongong
Sydney
Gold Coast
Brisbane
Cairns
Darwin
Alice Springs
Perth
Adelaide
Melbourne
Canberra
Wollongong
Total cost: 14988

Analyse and discussion:

As we could see from the result, by using the greedy algorithm the cost is higher than the cost of the branch and bound algorithm. This because greedy algorithm is actually not calculating the optimal solution of the TSP. But the greedy algorithm will cost less time. Cause the branch and bound algorithm probably will consider more cases than the greedy algorithm in the worst case.

Conclusion:

As the result above we could see that the greedy algorithm will probably not get the optimal solution, as it could be used as the upper bound of the branch and bound algorithm. And the branch and bound algorithm will always get the optimal solution. And to improve the efficiency of the algorithm, we could use the priority queue instead of the queue data structure.