CSCI203

Assignment 4

Asjad Athick
4970512

mama158

Methods

2 different methods were used to find solutions for the Travelling Salesman problem.

Data Structure Used: Adjacency Matrix representing the graph

Greedy Algorithm

Starting at the root node, the greedy algorithm chooses the lowest cost path to an unvisited node from the current node. Because the greedy algorithm doesn't backtrack or look into the future, it often does not produce the best solution.

Branch and Bound - Depth First Approach

Starting at the root node, the Branch and Bound algorithm generates a search space of live nodes, each with a lower and upper bound for their solutions. The algorithm then only pursues the branches with promising lower bounds, where the lower bound of the solution is less than the global upper bound. The global upper bound is an acceptable solution that may not necessarily be the best solution.

The approach is better than a brute force search because it does not pursue nodes with lower bounds higher than the global upper bound, reducing the search space drastically.

The depth first approach puts the generated live nodes on a stack, pops a node and generates child live nodes if required.

Results Analysis

The solution produced

The greedy algorithm although quick to run, doesn't produce the best solution.

The branch and bound algorithm takes longer to run, but produces the best possible solution.

Output for Australia_roads:

```
1. Greedy algorithm:

Number of cities: 11

Tour:

       1 2 6 4 5 7 8 10 3 11 9 1

       Wollongong -> Sydney -> Canberra -> Melbourne -> Adelaide -> Alice Springs ->
Darwin -> Cairns -> Brisbane -> Gold Coast -> Perth -> Wollongong

Total cost: 17637


2. Branch-and-bound algorithm (Depth-first):

Number of cities: 11

Upper bound: 17637

Lower bound: 9338

Optimal tour:

       1 2 11 3 10 8 7 9 5 4 6 1

       Wollongong -> Sydney -> Gold Coast -> Brisbane -> Cairns -> Darwin -> Alice
Springs -> Perth -> Adelaide -> Melbourne -> Canberra -> Wollongong

Total cost: 14988
```

The branch and bound costs 14988, compared to the 17637; a 2649 difference in cost.

Output for Australia_flights

```
1. Greedy algorithm:

Number of cities: 11

Tour:

       1 2 6 4 5 7 8 10 11 3 9 1

       Wollongong -> Sydney -> Canberra -> Melbourne -> Adelaide -> Alice Springs ->
Darwin -> Cairns -> Gold Coast -> Brisbane -> Perth -> Wollongong

Total cost: 24.6


2. Branch-and-bound algorithm (Depth-first):

Number of cities: 11

Upper bound: 24.6

Lower bound: 12

Optimal tour:

       1 3 11 10 8 7 9 5 4 6 2 1

       Wollongong -> Brisbane -> Gold Coast -> Cairns -> Darwin -> Alice Springs ->
Perth -> Adelaide -> Melbourne -> Canberra -> Sydney -> Wollongong

Total cost: 19.75
```

Again the branch and bound solution is cheaper, 19.75, compared to the 24.6 produced by the greedy algorithm

Performance

The greedy algorithm compares N nodes, making the best decision at every step. The processing time is considerably lower compared to branch and bound, where the search space is extensive.

For Australia_flights, where the graph size = 11,

The greedy algorithm simply visits 11 nodes.

The branch and bound approach

- Generated 75710 live nodes
- Evaluated 61 full tours, to check if it was the best path

Conclusion

The greedy algorithm is extremely quick to run compared to the branch and bound algorithm. However, the greedy algorithm solutions are often not the best, whereas branch and bound guarantees the best solution for every run.