Homework 15

Brian Knotten, Brett Schreiber, Brian Falkenstein

October 3, 2018

20

A decision tree is as follows: The decision tree can be thought of as follows for each node v at depth i:

- v designates each of $p_1, p_2, ...p_i$ into lists A and B, which represent the routes that either taxi A or taxi B travels.
- v's left child copies the assignments of v to A and B, but appends point p_{i+1} to A.
- v's right child copies the assignments of v to A and B, but appends point p_{i+1} to B.
- At depth i, i points have been visited.

Because each point needs to be visited by at least one taxi, and since there are no benefits for having both taxis visit a point, each point is either visited by taxi A or by taxi B. So there are 2^n possible point designations at the leaves of the decision tree.
The decision tree can be pruned using the following rules:

1. If two nodes u and v have the same total distance, $dist(u_A) + dist(u_B) = dist(v_A) + dist(v_B)$, prune v.

```
taxi(i, a, b):
    if i > n:
        return 0

    return min(
        dist(p[a], p[i]) + taxi(i + 1, i, b)
        dist(p[b], p[i]) + taxi(i + 1, a, i)
    )
```

Let $taxi[i, a, b]$ = the minimum total distance of designating A and B over points $p_1...p_i$, and A's last stop is p_a, and B's last stop is p_b. The dynamic solution algorithm is:

```
for a = 1 to n do:
    for b = 1 to n do:
        for i = n to 1 do:

            taxi[a, b, i] = min(
                                taxi[i, b, i + 1] + dist(p[a], p[i]),
                      taxi[a, i, i + 1] + dist(p[b], p[i])
                    )
```

21

The decision tree can be defined as:

- Every node v contains the cumulative response time up until that point in the path, and the path taken to get to v.
- Every node having at most n children, where each child represents the next point to travel to

- The leaves being paths covering all n points
- The solution being the minimum response time found at the leaves

Without pruning, the size of the tree will be n^n (height of n since every path must cover all points, and a branching factor of n). The tree can then be pruned using the following rules:

1. If a node visits a point in its path that it has already visited, prune it.
2. If 2 nodes at the same depth have visited the same points in its path, prune the one with the longer cumulative response time.