

TravelingSalespersonReport

Emily Elzinga

November 28 2022

1 Complexity

1.1 Time

Inner reduction for loops contribute $2n^2$, where n is the number of cities. When the for loop inside the *branch_and_bound* function is added in the complexity is $2n^3$. Finally, the outer while loop runs as long as there are subproblems in the queue or until time runs out. This loop adds b^n complexity, where b is the average number of nodes put on the queue with the expansion of each branch and n is the number of cities. This number will vary with each problem since the cost matrices of each branch are different. In the worst case when each city has a path to every other city, the while loop actually runs $(n - 1)!$ times.

So, the worst case time complexity is $O(n^3 * (n - 1)!)$, and the average case time complexity is $O(n^3 * b^n)$.

1.2 Space

The space complexity is $O(n^2 * b^n)$. The inner matrix reduction for loops again add n^2 , and the for loop inside the *branch_and_bound* function doesn't add any other complexity like it does time. So the only thing left is the priority queue. I used the built-in Priority Queue in the *queue* class, which uses the binary heap implementation. Worst case, the while loop adds $n!$ complexity. Average case, the complexity is b^n , where b is once again the branching factor.

2 Data Structures

[5] Describe the data structures you use to represent the states. I used lists appended to each other to create the cost matrices. **[5] Describe the priority queue data structure you use and how it works.** I used the *queue* Python class, which uses a binary heap to store data and works exactly the same as the one we designed in the Dijkstra project. **[5] Describe your approach for the initial BSSF.** I used a greedy algorithm. If the greedy approach yields no solution, the random search is used as a second resort.

3 Results Table

[25] Include a table containing the following columns.

Table 1 – Results from the TSP B&B algorithm

# Cities	Seed	Runtime (sec.)	Cost of best found	Max # of stored states	# of BSSF updates	Total # of states created	Total # of states pruned
15	20	5.88	10534	3107	1	24735	21628
16	902	3.17	7892	316	0	7041	6725
20	209	60	12858	66754	1	139065	72311
18	301	60	11714	51505	1	188001	136496
19	39	60	16519	140635	0	140642	7
17	6	33.9	9097	14303	1	123675	109372
16	782	24.8	8987	13149	1	97353	84204
15	691	1.29	7889	225	0	5298	5073
16	189	2.92	8339	1203	1	10616	9413
16	94	24.5	9799	10132	1	93330	83198