Pawan Acharya
CS4412
4/14/2022
Project 5

1. **Include your well-commented code**
   The source code for the project is included in the zip file.

2. **Time and Space complexity for the algorithm**
   For creating the initial bssf from a random tour, I think that the space complexity is O(n) and time complexity is also O(n). Similarly, to create the reduced cost matrix the space and time complexity is O(n^2). For SearchStates, the time complexity is O(n^3) and space complexity is also O(n^3). And for bssf initialization, the time complexity could take up to O(nlogn). So, the priority queue has both time and space complexity of O(n). Thus the overall branch and bound algorithm for this TSP problem will take O(n^3+logn) and space complexity of O(n^3).

3. **Describe the data structures you use to represent the states.**
   Results are stored in an empty dictionary. The other data structure used is 2D array; as a cost matrix, which consists of the index for representing the cities, distance between them and which path to go. This gave a better way to store the initial cost matrix. Similarly, priority queue data structure used is described below.
4. **Describe the priority queue data structure you use and how it works.**
   I think that the main idea was which state to go next, so using the built in data structure from the python “heapq” as priority queue, importing it, allowed an easier way. This way we can find the next state with the highest priority, and use it to navigate further down the tree.
5. **Describe your approach for the initial BSSF.**
   The initial bssf was taken from the random tour using the defaultRandomTour(). No other implementations were used for initial bssf this time. I will try to use a greedy approach to find the initial bssf in my group project.
6. **Include a table containing the following columns.**

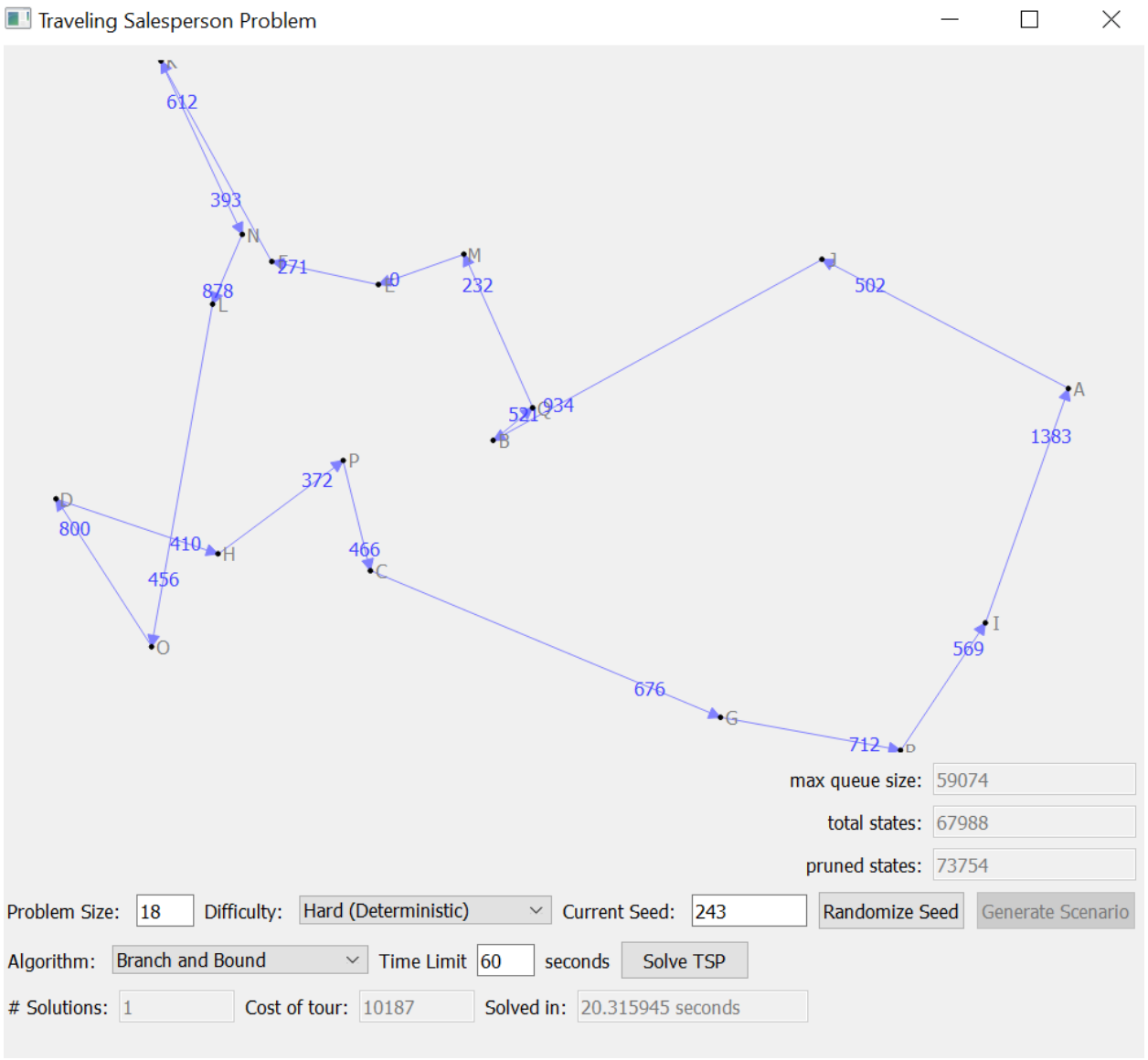
# Cities	Seed	Running time(sec)	Cost of best tour found (*=optimal)	Max # of stored states at a given time	# of BSSF updates	Total # of states created	Total # of states pruned
15	20	1.0810	10534	4083	1	4752	5030
16	902	2.3143	7954	8480	1	9752	10799
19	937	3.7544	9754	11648	1	12982	14420
17	674	19.7591	9592	60633	1	70684	80253

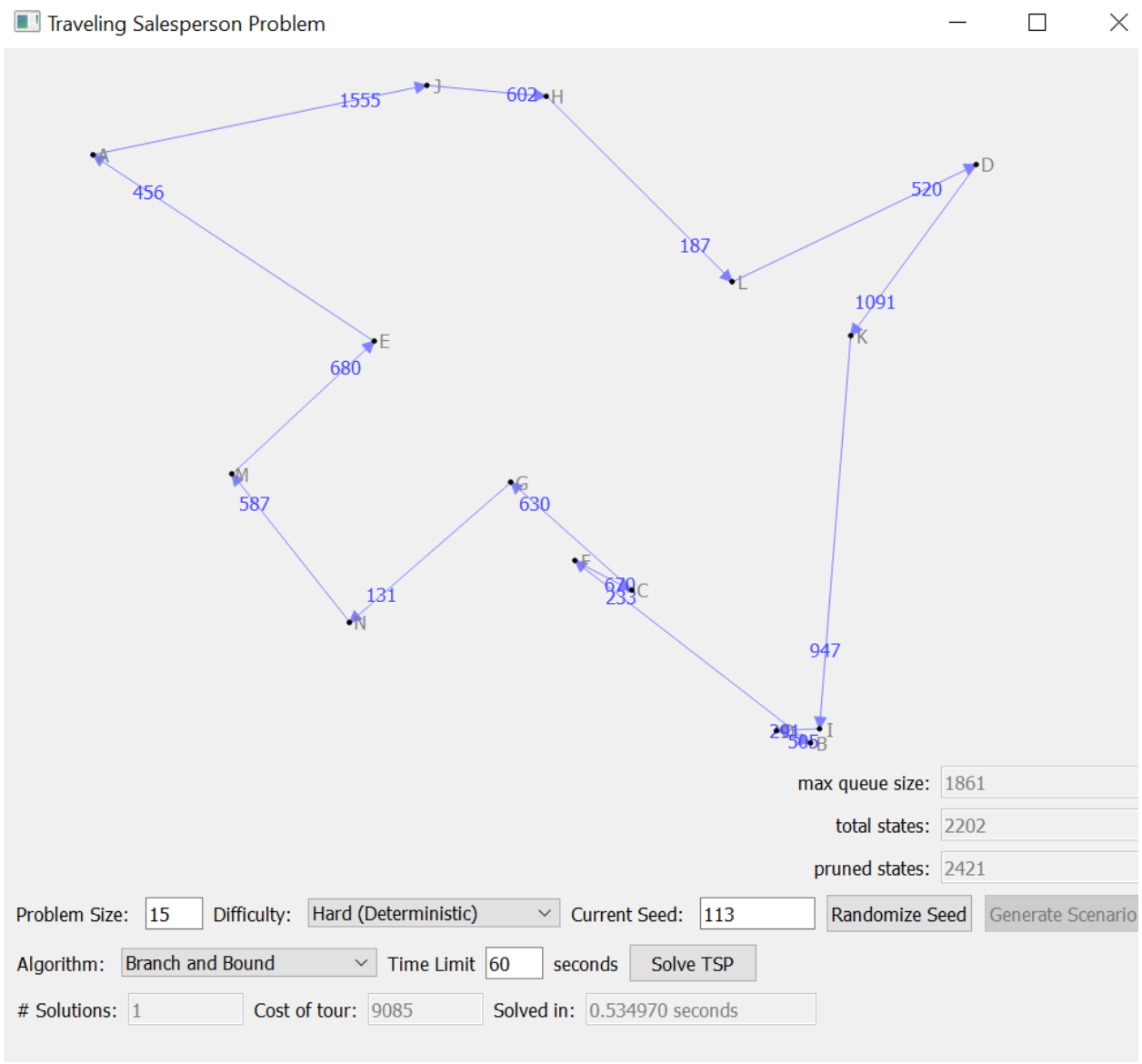
18	**505**	**29.2207**	**10156**	**94255**	**1**	**106735**	**116712**
20	**231**	**60.00035**	**32416**	**173822**	**0**	**192795**	**38638**
14	**101**	**5.4508**	**9073**	**20056**	**1**	**25264**	**28250**
12	**438**	**0.8547**	**8793**	**3228**	**1**	**4375**	**4544**
21	**200**	**60.0004**	**25295**	**153081**	**0**	**171122**	**50658**
23	**152**	**60.00337**	**31662**	**151281**	**0**	**164693**	**36447**

Discussion of the Results:
The direct jump in the time came as a result of my implementation. This is because I implemented in a way that, if it cannot find the optimum solution within a lower bound then it just returns the initial bssf. So, it stuck finding the leaf node with better cost then the bssf. So, the results from my algorithm are in the table above.

Mechanism: The main mechanism comprises: Priority Queue SearchStates Reduced Cost Matrix, and updating it BSSF Initialization Expanding one SearchState into others. While going down the tree, it could continue a lot, but cannot find an optimal solution within that given period of time. With these exceptions, the algorithm had to delve deeper down the tree, so new states were not discovered frequently before time ran out. More states would have been founded if the time limit had been increased. The number of times BSSF is updated is decreasing as more cities are added to the overall feasible trip.

Some Outputs:


Traveling Salesperson Problem
612
393
N
271
M
232
878
L
502
A
934
1383
B
372
P
D
800
410
H
466
C
456
O
I
569
676
G
712
max queue size: 59074
total states: 67988
pruned states: 73754
Problem Size: 18
Difficulty: Hard (Deterministic)
Current Seed: 243
Randomize Seed
Generate Scenario
Algorithm: Branch and Bound
Time Limit 60 seconds
Solve TSP
Solutions: 1
Cost of tour: 10187
Solved in: 20.315945 seconds


Traveling Salesperson Problem
J
H
A
D
L
K
E
M
G
F
C
N
B
I
1555
602
456
520
187
1091
680
587
630
131
233
947
max queue size: 1861
total states: 2202
pruned states: 2421
Problem Size: 15
Difficulty: Hard (Deterministic)
Current Seed: 113
Randomize Seed
Generate Scenario
Algorithm: Branch and Bound
Time Limit 60 seconds
Solve TSP
Solutions: 1
Cost of tour: 9085
Solved in: 0.534970 seconds