# Georgia Institute of Technology - 2023 Fall CSE 6140 Final Project

Group 22: Cai, Meng-Ru | Kao, Ting-Yang | Lin, Ping-Chuan | Lu, Tien | Pan, Yu-Chen (alphabet order)

### *Algorithms for Traveling Salesman Problem*

- Brute Force: Create an initial route according to the random seed input and then keep shuffling the tour to generate a new tour. To increase efficiency, fix the start location and only shuffle the rest of the tour since it is cyclical. In addition, when calculating the quality of a new tour, directly return non-full tour distance if the quality value is already higher than the current minimum quality. If random seed is not input, then default random seed 84 is used.
- Approximate: First, use Prim's algorithm to find the minimum spanning tree. Then use random seed to determine the starting location to find the tour. If random seed is not input, then default random seed 20 is used.
- Local search (simulated annealing): First, construct an initial tour and set up model parameters. Before reaching cutoff time or maximum iteration steps, randomly swap two locations in each iteration: if the new tour has a shorter distance, update the distance and tour; if not, then update the distance and path with a small probability so that it can escape from local minimum. To increase efficiency, apply the approximate approach to get the tour as our initial tour, and consider 3 patterns when randomly swapping two locations. Also, keep the best solution found during the process.

### *Test Results*

The following tests are done to get the optimal results:

- Use the 10 randomly selected seeds (9, 16, 18, 33, 58, 61, 64, 73, 84, 98) to test both BF and LS approaches.
- Set max cutoff time of both BF and LS approach as 100 seconds for all cities. But we find the optimal solution for Cincinnati and UKansasState in 10 seconds, and the optimal solution for Atlanta in 30 seconds.
- Use 100 random seeds (0 to 99) to find optimal solution for Approx approach.

| | BruteForce | | | | | Approx | | | | LS (avg of 10 random seed runs) | | | | LS (best quality) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset(.tsp) | Times | Sol.Quality | Full.Tour | seed | RelError | Sol.Quality | Full.Tour | seed | RelError | Times | Sol.Quality | Full.Tour | RelError | Times | Sol.Quality | Full.Tour | seed |
| Atlanta | 100 | 2,559,425 | Y | 98 | 28% | 2,206,618 | Y | 1 | 10% | 30 | 2,003,763 | Y | 0% | 30 | 2,003,763 | Y | 18 |
| Berlin | 100 | 19,947 | Y | 73 | 164% | 9,285 | Y | 52 | 23% | 100 | 7,840 | Y | 4% | 100 | 7,542 | Y | 9 |
| Boston | 100 | 1,789,478 | Y | 18 | 100% | 1,028,108 | Y | 32 | 15% | 100 | 898,727 | Y | 1% | 100 | 893,536 | Y | 18 |
| Champaign | 100 | 141,506 | Y | 9 | 169% | 59,686 | Y | 5 | 13% | 100 | 53,335 | Y | 1% | 100 | 52,657 | Y | 61 |
| Cincinnati | 10 | 277,952 | Y | 18 | 0% | 297,490 | Y | 26 | 7% | 10 | 277,952 | Y | 0% | 10 | 277,952 | Y | 73 |
| Denver | 100 | 396,683 | Y | 84 | 288% | 125,541 | Y | 20 | 23% | 100 | 104,634 | Y | 2% | 100 | 102,277 | Y | 9 |
| NYC | 100 | 4,969,866 | Y | 33 | 216% | 1,857,564 | Y | 10 | 18% | 100 | 1,592,985 | Y | 1% | 100 | 1,574,005 | Y | 58 |
| Philadelphia | 100 | 2,105,593 | Y | 64 | 51% | 1,685,283 | Y | 16 | 21% | 100 | 1,410,444 | Y | 1% | 100 | 1,395,981 | Y | 64 |
| Roanoke | 100 | 5,889,479 | Y | 18 | 728% | 786,888 | Y | 20 | 11% | 100 | 721,111 | Y | 1% | 100 | 711,530 | Y | 84 |
| SanFrancisco | 100 | 4,358,719 | Y | 84 | 412% | 1,032,609 | Y | 74 | 21% | 100 | 874,423 | Y | 3% | 100 | 852,127 | Y | 73 |
| Toronto | 100 | 7,596,730 | Y | 61 | 502% | 1,580,203 | Y | 39 | 25% | 100 | 1,302,123 | Y | 3% | 100 | 1,261,494 | Y | 9 |
| UKansasState | 10 | 62,962 | Y | 18 | 0% | 67,636 | Y | 0 | 7% | 10 | 62,962 | Y | 0% | 10 | 62,962 | Y | 58 |
| Umissouri | 100 | 575,236 | Y | 84 | 330% | 164,665 | Y | 43 | 23% | 100 | 137,533 | Y | 3% | 100 | 133,825 | Y | 73 |

### *Analysis of Local search algorithm*

The following plots are generated from the best quality result of each city by setting run time as 100 secs.

- For Cincinnati and UKansasState, we set the final cutoff time as 10 secs because even though the model still has not converged yet, we already found the best solution.
- For Atlanta, we set the final cutoff time as 30 secs because all 10 test cases already found the same answer, which we believe is the best solution.
- For all other cities, the trend is obvious. The model would escape from local minimum and gradually converge toward global minimum. The usage of MST as an initial tour is to eliminate solutions that are far away from ground truth, which cannot become ground truth by swapping two nodes.
- We only use two parameters to fine tune the model and apply to all cities, making it difficult to find global minimums for all cities since the characteristics of each city are different, for example some with fewer locations but higher

average distance between two locations, and vice versa. If fine tune the model for each city with different parameters, then the result could be better.