CSE 6140 Project Report

Chao-Wen Chang, Bowen Mu, Mo Sun, Mengjiao Wu, Xiaohan Yu

December 4, 2023

As we been told in the assignment, we used 3 algorithms to solve the TSP problem.

For the **brute force** algorithm, we enumerated all the possible routes by conducting permutations of the location list and updated the minimum total route cost until the time limit reaches. This algorithm runtime is $O(n!)$ where n is the number of locations. However, this method guarantees a global optimal solution if we have infinite time.

The **approximation** algorithm is implemented by applying Prim's algorithm to find the minimum spanning tree (MST). In words, it started from the first location and repeatedly grows the spanning tree by adding the shortest edge that connects a visited location to an unvisited location. This process continued until all locations are included in the spanning tree. All the selected edges were stored. Next, the depth first search (DFS) algorithm was implemented to find the order to visit all locations. This method has polynomial runtime $O(ElogV)$, where $E = V(V-1)$ is the number of edges and V is the number of locations. This method has the smallest computational cost among the three and it guarantees a total cost no more than 2 times of the optimal solution.

We used simulated annealing method for the **local search**. A random route is initialized by shuffling the location list and we calculate the cost. We then generate a trial route by randomly swap two locations' order and calculate the new cost. We always accept the new trial with lower energy, and we accept the new trial with higher energy with an acceptance rate of $exp(\frac{E_{new}-E_{old}}{kT})$, where E is the cost, k is a constant, and T is temperature. By this mean, we are able to include configurations with higher energy with certain probability that escape from local minima and explore for more possible solution. Higher temperatures result in more capability to jump across energy barriers. Temperature gradually went down by a fixed ratio every M steps. We stop the iteration after the cutoff time, a certain temperature or a total number of steps is reached. The runtime and the quality of the solution is in general between the first and second algorithm. We also tuned the hyper-parameters (initial T, k, $nSteps$, M, $coolingFraction$) for the best performance before pushing them to GitHub.

The performance of each algorithm is tabulated in the table below. With the cut off time 300s, MST and the Local Search method outperform the BF method. The increase of cutoff time enhances the BF and the LS performances since they did not compute the full tour (FT in the table) for most of the graphs, while the MST method always results in full tours since it is a deterministic algorithm.

We calculated the relative error (RelError) of the solutions relative to the best solution found (for LS, we chose the best run of the 10 as the best solution when applicable). In LS-Sol.Quality, we list the average of the ten runs using different random seeds, and we calculate the relative error between the average solution of LS and the best solution of LS when LS is the best algorithm. When FT could be achieved, BF and LS both provided optimal solution, though LS took longer. For other cases, Approximation algorithm generally provided good solution in a short time, while the performance of LS varied and BF tended to provide the solution far from the optimal.

Performances of the three Algorithms

	Brute Force (BF)				Approximation				Local Search (LS)					
Dataset	Time(s)	Sol.Quality	FT	RelError	Time(s)	Sol.Quality	FT	RelError	Time(s)	Sol.Quality	FT	RelError	Best Sol.	Best Algo
Atlanta	300	3630534	No	0.812	0.0004	2314410	Yes	0.155	224.12	2018144	Yes	0.007	2003763	LS
Berlin	300	19438	No	0.901	0.0038	10227	Yes	0	300	27163	No	1.656	10227	Approx.
Boston	300	2227323	No	1.463	0.002	1142864	Yes	0.264	300	924316	No	0.022	904212	LS
Champaign	300	215798	No	2.344	0.004	64529	Yes	0	300	111641	No	0.730	64529	Approx.
Cincinnati	26.4	277952	Yes	0	0.0001	308737	Yes	0.111	129.16	278231	Yes	0.001	277952	BF/LS
Denver	300	547974	No	2.986	0.012	137474	Yes	0	300	344123	No	1.503	137474	Approx.
NYC	300	7210976	No	3.479	0.0074	2011514	Yes	0.249	300	1649493	No	0.024	1610047	LS
Philadelphia	300	3624919	No	1.597	0.0011	1648282	Yes	0.181	300	1428536	No	0.023	1395981	LS
Roanoke	300	6849948	No	7.097	0.199	845944	Yes	0	300	3099697	No	2.664	845944	Approx.
SanFrancisca	300	5600573	No	4.280	0.019	1153336	Yes	0.087	300	1234379	No	0.164	1060717	LS
Toronto	300	9179267	No	4.512	0.025	1754373	Yes	0.054	300	1724000	No	0.035	1665254	LS
UKansasState	25.8	62962	Yes	0	0.0001	68006	Yes	0.080	127.89	63015	Yes	0.001	62962	BF/LS
UMissouri	300	662935	No	2.988	0.023	166220	Yes	0	300	555361	No	2.341	166220	Approx.