Project Report

Group 16

Members: Junyang Tang, Zeyu Wang, Zoey (Li-Yen) Yang, Xiaodong An, Lu Li

1. Description of the algorithms

Brute Force Algorithm: This algorithm generates all possible routes (using permutation functions) for the Traveling Salesman Problem (TSP) and calculates their costs to find the optimal solution. Suitable for small-scale TSP problems due to its time-inefficiency. For larger problems, it randomly generates and compares full-route solutions within a time limit, ensuring each solution is a valid Hamiltonian path. The accuracy improves with longer cut-off times, but the exponential time cost is a significant drawback.

Approximation Algorithm: Utilizes a heuristic approach to find near-optimal solutions for TSP. It constructs a complete graph with cities as nodes and distances as edges, then calculates the Minimum Spanning Tree (MST) of this graph. A preorder traversal of the MST provides an approximate tour of the cities. This method is more efficient than exact methods, trading some accuracy for efficiency.

Local Search (Hill Climb Algorithm): Starts with a randomized solution and iteratively improves it by making small changes (like swapping two nodes in the route). The solution is updated if it improves (minimizes or maximizes the desired quality). The process continues until a cut-off time is reached or no further improvement is possible. It's efficient but limited to local min/max solutions.

Local Search (Simulated Annealing): This probabilistic optimization algorithm begins with a random solution and iteratively refines it by comparing with slightly modified solutions. It accepts better solutions and, to escape local minimums, sometimes worse solutions based on a controlled probability linked to a decreasing "temperature". It's effective for problems with multiple local minimums, balancing the exploration of the solution space with convergence to an optimal solution.

2. Performance

DataSet	BF (cut off:T = 300s)			Approximate		LS (Hill Climb)			LS(Annealing)		
	Time (s)	Quality	Full Tour	Time (s)	Quality	Time (s)	Quality (seed 1-10 averaged)	RelError	Time (s)	Sol.Quality (seed 1-10 averaged)	RelError
Atlanta.tsp				0.002s	2380448	0.01	3187953	30.89%	0.01	2464553	10.61%
Atlanta.tsp						0.1	2488539	11.47%	0.1	2300077	4.21%
Atlanta.tsp	300.000s	2578866	Yes			1	2480534	11.18%	1	2203146	0.00%
Atlanta.tsp						300	2480534	11.18%	300	2295670	4.03%
Berlin.tsp						0.01	25419	60.12%	0.01	16959	40.23%
Berlin.tsp						0.1	15855	36.06%	0.1	11492	11.79%
Berlin.tsp	300.000s	20514	Yes	0.003s	10402	1	11243	9.84%	1	10309	1.67%
Berlin.tsp						300	10505	3.50%	300	10137	0.00%
Boston.tsp						0.01	2249811	51.33%	0.01	1540072	28.89%
Boston.tsp						0.1	1462034	25.10%	0.1	1108242	1.19%
Boston.tsp	300.000s	1848210	Yes	0.004s	1150963	1	1112085	1.53%	1	1129011	3.00%
Boston.tsp						300	1099976	0.44%	300	1095088	0.00%
Champaign.tsp						0.01	175706	62.60%	0.01	122081	46.17%
Champaign.tsp						0.1	117135	43.90%	0.1	81256	19.13%
Champaign.tsp	300.000s	147540	Yes	0.007s	65712	1	80148	18.01%	1	75366	12.81%
Champaign.tsp						300	74577	11.89%	300	69362	5.26%
Cincinnati.tsp						0.01	284191	2.20%	0.01	281041	1.10%
Cincinnati.tsp						0.1	283343	1.90%	0.1	280676	0.97%
Cincinnati.tsp	300.000s	277952	Yes	0.002s	301216	1	283343	1.90%	1	282825	1.72%
Cincinnati.tsp						300	283343	1.90%	300	282278	1.53%
Denver.tsp						0.01	500028	73.05%	0.01	348638	61.35%
Denver.tsp						0.1	358087	62.37%	0.1	217627	38.08%
Denver.tsp	300.000s	408153	Yes	0.014s	134748	1	217784	38.13%	1	160782	16.19%
Denver.tsp						300	157554	14.48%	300	160043	15.81%
NYC.tsp						0.01	6540970	69.01%	0.01	4310077	52.97%
NYC.tsp						0.1	4600574	55.94%	0.1	2725149	25.61%
NYC.tsp	300.000s	5347309	Yes	0.009s	2027107	1	2753457	26.38%	1	2302735	11.97%
NYC.tsp						300	2287107	11.37%	300	2309125	12.21%
Philadelphia.tsp						0.01	2966213	44.50%	0.01	1924767	14.47%
Philadelphia.tsp						0.1	1910066	13.81%	0.1	1761882	6.56%
Philadelphia.tsp	300.000s	2271573	Yes	0.003s	1646249	1	1746182	5.72%	1	1719899	4.28%
Philadelphia.tsp						300	1746182	5.72%	300	1838309	10.45%
Roanoke.tsp						0.01	7050109	88.11%	0.01	6278414	86.65%
Roanoke.tsp						0.1	6270761	86.63%	0.1	4169776	79.90%
Roanoke.tsp	300.000s	6013414	Yes	0.077s	838282	1	4384446	80.88%	1	2217859	62.20%
Roanoke.tsp						300	1488289	43.67%	300	1482709	43.46%
SanFrancisco.tsp						0.01	5421481	79.06%	0.01	3946717	71.24%
SanFrancisco.tsp						0.1	3937235	71.17%	0.1	2400645	52.72%
SanFrancisco.tsp	300.000s	4465539	Yes	0.019s	1134989	1	2436426	53.42%	1	1581011	28.21%
SanFrancisco.tsp						300	1521652	25.41%	300	1567765	27.60%
Toronto.tsp						0.01	9604665	82.56%	0.01	6616722	74.68%
Toronto.tsp						0.1	6830478	75.48%	0.1	4000581	58.13%
Toronto.tsp	300.000s	7914377	Yes	0.022s	1675105	1	3978072	57.89%	1	2656610	36.95%
Toronto.tsp						300	2385423	29.78%	300	2506264	33.16%
UKansasState.tsp						0.01	66955	5.96%	0.01	62962	0.00%
UKansasState.tsp						0.1	66699	5.60%	0.1	62962	0.00%
UKansasState.tsp	300.000s	62962	Yes	0.002s	68090	1	66699	5.60%	1	62962	0.00%
UKansasState.tsp						300	66699	5.60%	300	62962	0.00%
UMissouri.tsp						0.01	711831	74.96%	0.01	531837	66.48%
UMissouri.tsp						0.1	533344	66.58%	0.1	325570	45.25%
UMissouri.tsp	300.000s	589018	Yes	0.021s	178249	1	338192	47.29%	1	214576	16.93%
UMissouri.tsp						300	215698	17.36%	300	207108	13.93%

3. The Effect of Cutoff Time

We conducted experiments with cutoff times ranging from 0.01 to 300 seconds on our algorithms.

Brute Force: While increasing the cutoff time does improve the performance of the brute force method, its effectiveness is still not ideal compared to other algorithms, even at the maximum cutoff time of 300 seconds. This is because it only explores a small fraction of all possible permutations within the given time, making it challenging to guarantee an optimal solution. Although an optimal solution can be obtained using this method, the computation time on a standard computer could span from days to years.

Approximation:

The results clearly indicate that the actual runtime of the Approximation algorithm is typically less than or around 0.01 seconds in most cases. Therefore, investigating the impact of cutoff time on the Approximation Method is not particularly meaningful.

Local Search (Metropolis Algorithm):

For most cases, a larger cutoff time tends to yield better solution quality. However, there are exceptions, such as when the algorithm has already reached a local minimum early on, and extending the time does not improve the quality further.

Local Search (Simulated Annealing):

Generally, a longer cutoff time leads to better solutions (lower cost). However, on occasion, especially with a large cutoff time (300 seconds), the algorithm may return a solution of worse quality.