CSE 6140 Final project

Seungjun Cha, Jimin Park, Soomin Lim

(a) A brief description of each of the algorithms

Brute force:

This algorithm is a **brute force approach** that performs a depth-first search over all possible tours while pruning any partial paths whose accumulated cost exceeds the current best solution. A nearest-neighbor tour provides the initial upper bound, and a time cutoff ensures the search terminates while returning the best tour found within the allowed time.

Minimum spanning tree approximation:

This algorithm is an **approximation method** that first computes a minimum spanning tree of the graph using Prim's algorithm, then performs a preorder depth-first traversal of the MST to obtain a Hamiltonian tour.

2-opt local search:

This algorithm is a **first-improvement 2-opt local search method** that begins with a nearest-neighbor tour (with a random start) and repeatedly scans edge pairs to find the first 2-opt move that reduces the tour cost. Upon finding any improving swap, it immediately applies and restarts the search in the remaining edge pairs, continuing this hill-climbing process until no further improvements occur or the time cutoff is reached.

This usually converges very fast because it is confined locally to a single initialized tour. For part (c), we have implemented the multi-start version, which repeatedly generates new tours and applies first-improvement 2-opt hill climbing to each one, strictly until the time cutoff is reached. This aims to explore the other feasible space beyond the local optima and return the best 'global' solution.

(b) Results table

Red denotes cases where the algorithm achieved the best solution, and **purple** denotes those where it achieved the worst. The reference value for the relative error (RelError) was defined as the lowest tour cost obtained across all local search (LS) runs, which served as the global best known solution for each city.

Overall, the LS method substantially outperformed both the brute-force and approximation approaches, with a significant advantage in execution time. Across all cities, the 10-run average of LS was consistently better than the solution quality of the approximation method. LS was slightly worse than brute force in only two cases (Atlanta and Cincinnati), where the number of nodes was small enough for brute force to find the optimum within the 600-second cutoff. For the case of UKansasState with only 10 nodes, both brute force and LS were able to reach the global minimum. Regardless, the best solution found across all methods and runs was always produced by LS.

	Brute force (cutoff = 600 s)			
	time	Sol.Quality	full tour	**RelError**
Atlanta	600	**2025073**	TRUE	21310
Berlin	600	8324	TRUE	782
Boston	600	1028480	TRUE	118092
Champaign	600	58797	TRUE	5806
Cincinnati	0.286	**277952**	TRUE	0
Denver	0.286	109681	TRUE	6695
NYC	600.01	1816015	TRUE	240305
Philadelphia	600	1464819	TRUE	64773
Roanoke	599.99	779832	TRUE	102783
SanFrancisco	599.99	896718	TRUE	59065
Toronto	600.01	1358974	TRUE	166909
UKansasState	0.0408	**62962**	TRUE	0
UMissouri	600	157181	TRUE	18747

Table 1. Brute force result

	Approx			
	time	Sol.Quality	full tour	**RelError**
Atlanta	8.45E-05	**2380448**	TRUE	376685
Berlin	0.0003418	**10402**	TRUE	2860
Boston	0.0002337	**1150963**	TRUE	240575
Champaign	0.000548	**65712**	TRUE	12721
Cincinnati	4.04E-05	**301216**	TRUE	23264
Denver	0.0010061	**134748**	TRUE	31762
NYC	0.0005256	**2027107**	TRUE	451397
Philadelphia	0.000282	**1646249**	TRUE	246203
Roanoke	0.0047774	**838282**	TRUE	161233
SanFrancisco	0.000825	**1134989**	TRUE	297336
Toronto	0.0013222	**1675105**	TRUE	483040
UKansasState	4.33E-05	**68090**	TRUE	5128
UMissouri	0.0011735	**178249**	TRUE	39815

Table 2. Minimum spanning tree approximation result

	LS (10 runs each)				
	time (Average)	Sol.Quality (Average)	Sol.Quality (Min)	full tour	**RelError (Average)**
Atlanta	0.00031309	2051419.2	2003763	TRUE	47656.2
Berlin	0.00321633	**7902.9**	7542	TRUE	360.9
Boston	0.00172117	**933085**	910388	TRUE	22697
Champaign	0.00497032	**53733.5**	52991	TRUE	742.5
Cincinnati	9.70E-05	279297.5	277952	TRUE	1345.5
Denver	0.01579591	**104891.3**	102986	TRUE	1905.3
NYC	0.00909575	**1624081.7**	1575710	TRUE	48371.7
Philadelphia	0.00110749	**1426365**	1400046	TRUE	26319
Roanoke	0.3174452	**686904.9**	677049	TRUE	9855.9
SanFrancisco	0.02164162	**852824.3**	837653	TRUE	15171.3
Toronto	0.03259639	**1236076.3**	1192065	TRUE	44011.3
UKansasState	7.93E-05	**62962**	62962	TRUE	0
UMissouri	0.02849571	**141642.3**	138434	TRUE	3208.3

Table 3. 2-opt local search result

(c) Cutoff time vs. solution quality

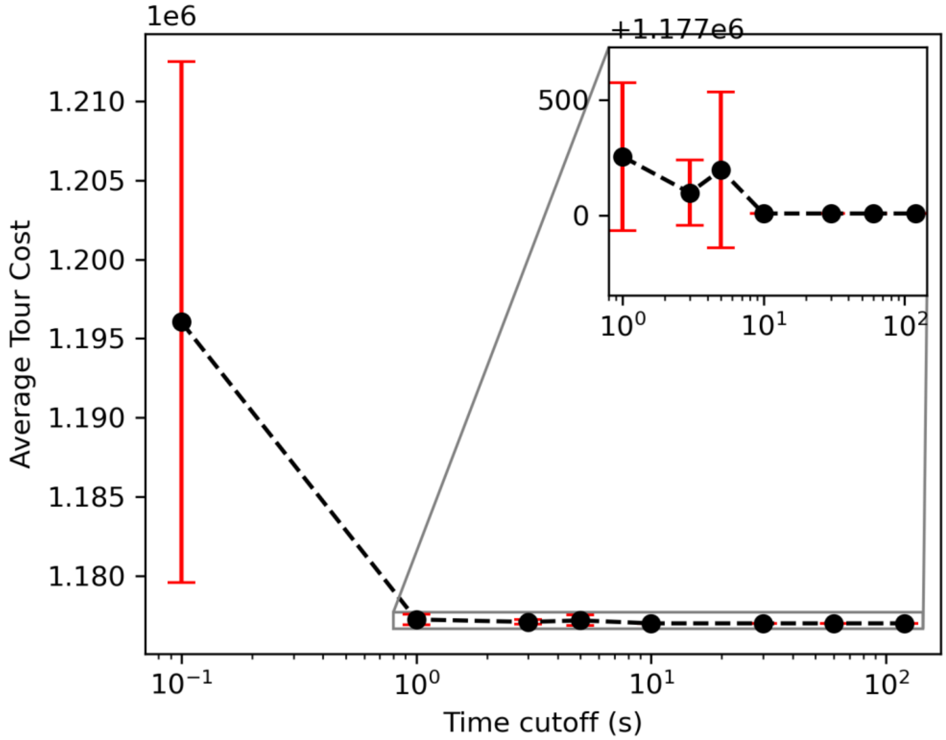

Figure 1. Cutoff time vs. 10-runs average tour cost using the 2-opt multi-start LS method. The red error bar indicates the standard deviation across 10 runs for each time cutoff.

Here, 10 runs for each of the 8 different cutoff times (0.1, 1.0, 3.0, 5.0, 10.0, 30.0, 60.0, 120.0 seconds) were tested in order to study the effect of cutoff time on the average tour cost. As expected, the average tour cost decreases rapidly as the time cutoff increases. By the 10-second cutoff, all of the runs were able to reach the same optima, resulting in the standard deviation of zero. No further improvement was observed beyond the 10-second cutoff.