IE 555 PROGRAMMING FOR ANALYTICS

ASSIGNMENT NO 4: SIMULATED ANNEALING

Name: Aditya Patankar UB person #: 50204875

1. Flow chart describing the process:

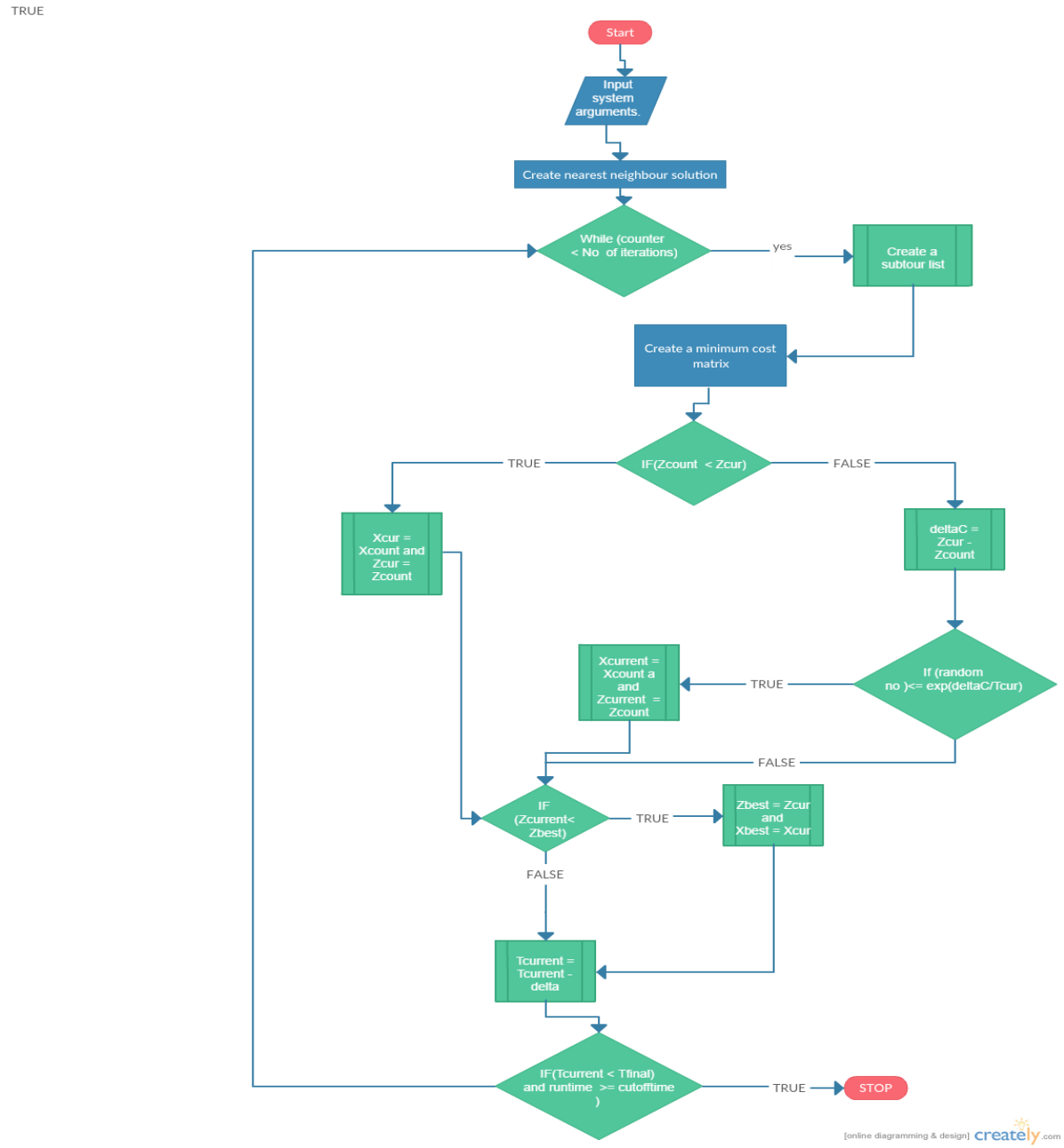

Figure 1 : Flow chart describing the nature of simulated annealing heuristic.

The input data is obtained in terms of a csv file for the location of various nodes in terms of the latitude and the longitude the various locations. Another form of input is obtained in terms of the system arguments to determine if need the distance or time, Gurobi or nearest neighbor or the Simulated annealing heuristic.

- The simulated annealing procedure takes the inputs from the initial values obtained from the nearest neighbor heuristic. The route and the cost are obtained through a function call and are saved as Xini and Zini values. The simulated annealing procedure takes the outputs from the functions named as 'subtourfun' and the 'mincostreturn'.
- 'Subtourfun' Function : This function is used to generate subtours by swapping the locations of the various indices of the tours. The indices are generated randomly and these indices are used ahead to create the reversal of the lists. Five subtours are generated and stored as a list of list.
- Cost calculation : The cost calculation is used to calculate the cost of every subtour out of the five subtours generated. These costs are saved in terms of a dictionary with the key of the dictionary representing the subtour number and the values representing the cost of the subtours.
- Mincostreturn : The minimum cost return function has been used to identify the minimum cost of all the five possible subtours. The indices corresponding to the minimum cost are also saved in order to be used later to identify the best possible subtour. This sub-tour will be called by the simulated annealing as the best possible sub-tour.
- Simulated annealing: The simulated annealing procedure uses the nearest neighbor heuristic as the initial value but later it updates the value of the route and the cost obtained. This update happens after taking into consideration the acceptance probability. If the value of the acceptance probability is greater than the random number generated, then the bad solutions are also accepted. This acceptance of bad solutions, helps eliminate the possibility of getting stuck in a local optimum.

2. Variation the objective function value with the change in parameters.

Table 1 : Variation of the objective function with the change in parameters.

Parameter				Zvalues
Tzero	Tfinal	delta	I	Distance
50	0.1	0.1	7000	67.261
20	0.1	0.1	700	67.261
200	**1**	**0.8**	**7000**	**60.876**
10	0.005	0.05	7000	63.061
100	0.5	0.1	1000	64.702

- The table 1 shows the variation of the objective function values with the change in the parameters of the values like Tzero, Tfinal, delta and the maximum number of iterations I. The best values are obtained for the parameters Tzero = 200, Tfinal = 1 and delta = 0.8. The values of the distance as the objective function is 60.876. For this objective function value, it is observed that the delta value is 0.8. With the higher temperature set initially, it is observed that there is a higher chance of acceptance at the beginning of the heuristic. Hence there is a chance that even if bad values are accepted, it may

take us to a better valley which may contain the global optimum. This is verified from this observation as well that the objective value obtained is the best compared to others, which shows that accepting bad moves may also lead to better objective function value.

3.

Table 2: Objective function value with constant parameters.

Trial No.	objective value	runtime
1	64.654	12.876
2	67.887	10.647
3	67.621	9.998
4	65.373	10.887
5	63.383	11.887
6	60.876	12.464
7	68.788	12.474
8	70.787	12.074
9	69.677	11.987
10	67.843	11.675

According to the results obtained as shown in the table 2 for the constant parameters with Tzero = 200, Tfinal = 1, delta = 0.8 , I = 7000; it is observed that there is certain variation in the objective function value and the number of iterations. It occurs because the heuristic is based on random numbers, and hence the probability of acceptance varies with each iteration. Hence the number of iterations also varies impacting the run time and the objective function value. This difference is however not a major difference as compared to the difference obtained from changing the parameters.

4.The figure obtained shows the plot of the objective function value vs the no of iterations. The red diamond shows the accepted values, the green hexagon shows the values that are accepted but are actually worse and the yellow hexagon shows the values that are not accepted. There are many bad solutions that are accepted earlier because of the higher acceptance probability at the beginning, which keeps on decreasing as the temperature decreases. Hence we see a lot of green hexagons at the beginning and a lot of yellow hexagons at the end. The heuristic converges to a locally optimum solution as the temperature keeps on decreasing and at the end when the temperature has reduced to Tfinal, it provides with the best of the local optimums.

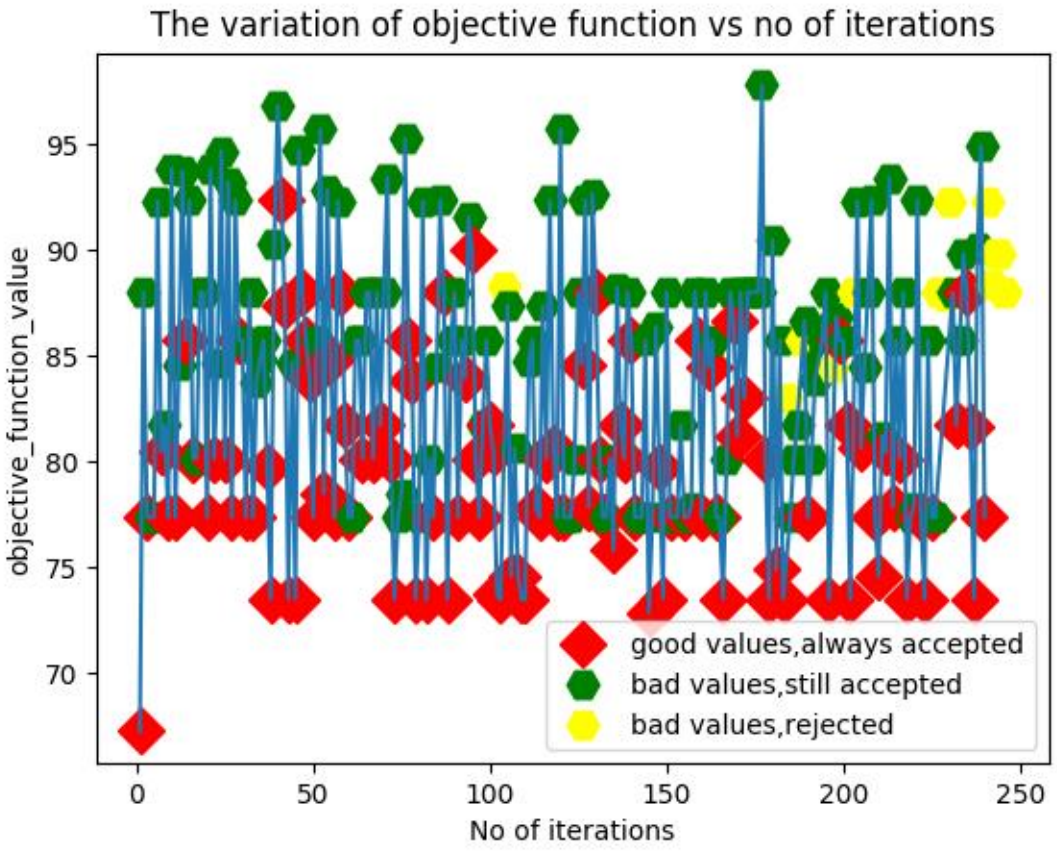

Figure 2: Plot of the objective function vs no of iterations showing the progress of SA heuristic

5. The figure 3 gives a detailed view of the routes obtained from the NN, Gurobi and the SA routes, with the red line indicating the Nearest Neighbor route, the green line showing the IP route, and the blue line showing the route obtained from the SA route.

Table 3: GUROBI vs SA comparison

	GUROBI	SA	DIFFERENCE	%GAP
RUNTIME	1.105	13.157	1205.2	1090.679
OBJECTIVE VALUE	56.4	61.934	553.4	9.812057

The table 3 shows the comparison of the objective function value and the runtime for Gurobi and SA. It is observed that the value obtained from the IP formulation is the global optimum. The runtime for Gurobi is also significantly lesser than the runtime for SA.

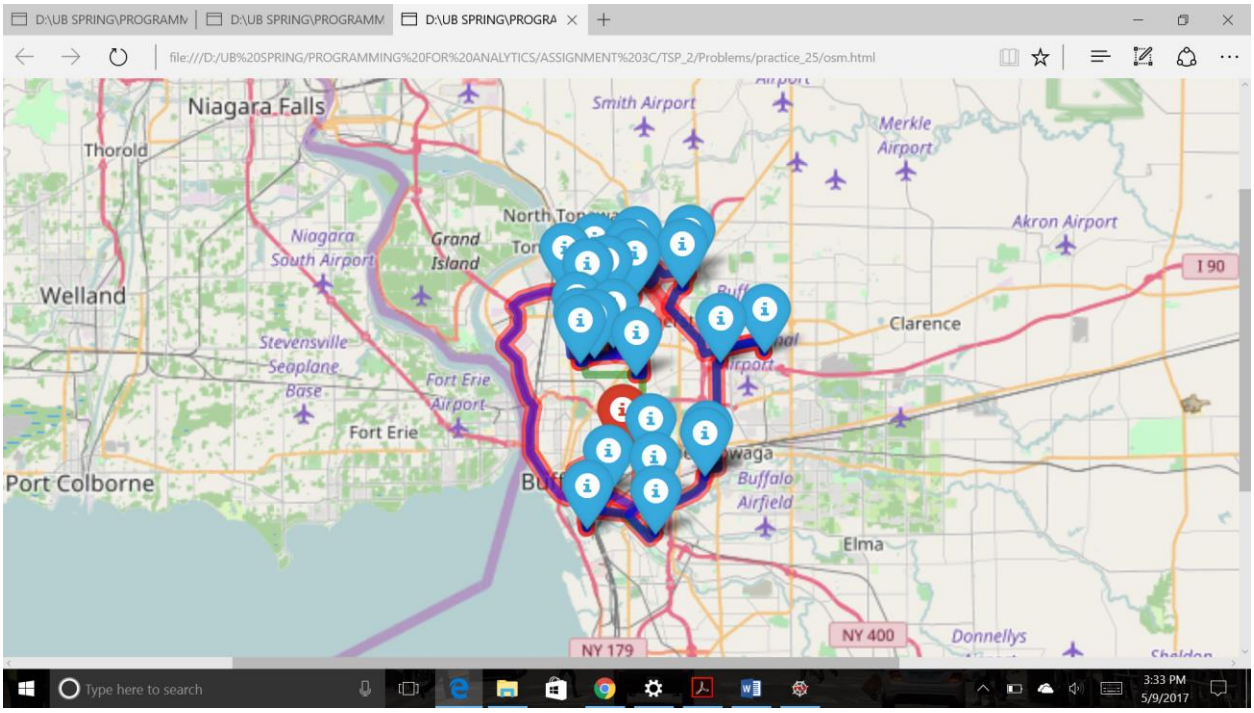

Figure 3: Osm map showing the NN, Gurobi and SA routes.

6. In the command line enter:

python APATANKA_SOLVE_TSP.py practice_25 1 1 1 1 -1 1

where :

APATANKA_SOLVE_TSP.py is the submitted file name.

practice_25 : locations folder.

1 : objective Type

1 : solve using NN

1: solve using IP

1: solve using SA

-1: No time limit, maximum no of seconds for Gurobi to run.

1: detailed routes from MapQuest (TurnbyTurn)