ARTIFICIAL INTELLIGENCE

ASSIGNMENT 3

TSP-Simulated Annealing

CHARU CHAUHAN(CS14M058) & MEENAKSHI RAY(CS14M031)

11/28/2014

ASSIGNMENT STATEMENT: Experiments with cooling rate and various neighborhood functions. A set of city exchange and edge exchange neighborhood operators must be available to the user. The starting tour maybe randomly generated, or imported from the greedy approach. Plot the cost of the tour as the program runs.

INTRODUCTION:Simulated Annealing is a generic probabilistic algorithm used to find an approximate solution to global optimization problems. It is inspired by annealing in metallurgy which is a technique of controlled cooling of material to reduce defects. The simulated annealing algorithm starts with a random solution. Each iteration finds a random nearby solution by using a perturbation method. If this solution is a better solution, it will replace the current solution. If it is a worse solution, it may be chosen to replace the current solution with a probability that depends on the temperature parameter. As the algorithm progresses, the temperature parameter decreases, giving worse solutions a lesser chance of replacing the current solution. The algorithm strategy shifts from explorative tendency to exploitative strategy with time. Fig 1 illustrates the overall flow of simulated annealing algorithm.The initial explorative tendency helps us to avoid the problem of getting stuck in local optima(Refer Fig 2). At later stages when the temperature is less then the explorative tendency becomes insignificant and the movement is largely with the gradient.

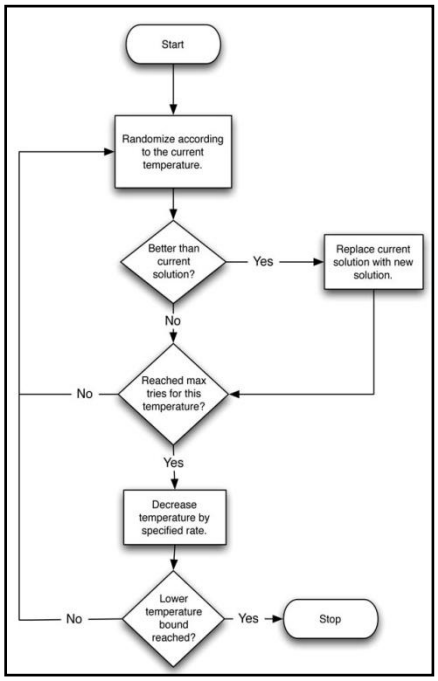

Fig - 1

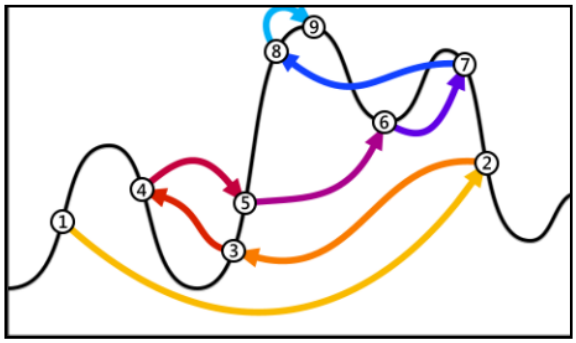

Fig - 2

ASSIGNMENT DETAILS: In this assignment we have implemented simulated annealing algorithm for TSP. We have provided the user flexibility to choose the initial temperature and the cooling rate so that **cooling schedule** can be set as required. We have also given the option for choosing the perturbation method. We can either use **the 2 city exchange** perturbation or **edge exchange(2 or 3)** perturbation. In addition to this the user has the freedom to choose how often the visualization should be updated. This gives the user an option to speedup the simulation so that more time is spent in computation and less time in updating UI. We have used the concept of simulated annealing restart. After **annealing process stops** we feed the solution of this process to annealing **restart** process. The cooling schedule for annealing restart process is kept fixed , starting temperature 100 and cooling rate 0.999.This is because the performance doesn't significantly improve even if starting temperature is kept too high in restart process. We have generated the temperature vs tour cost graph for both annealing and annealing restart process. The restart process provides **significant improvement** in the tour cost. Please refer Fig 3 and Fig 4 respectively to see how the final tour and tour cost is updated with temperature in both simulated annealing and annealing restart process.

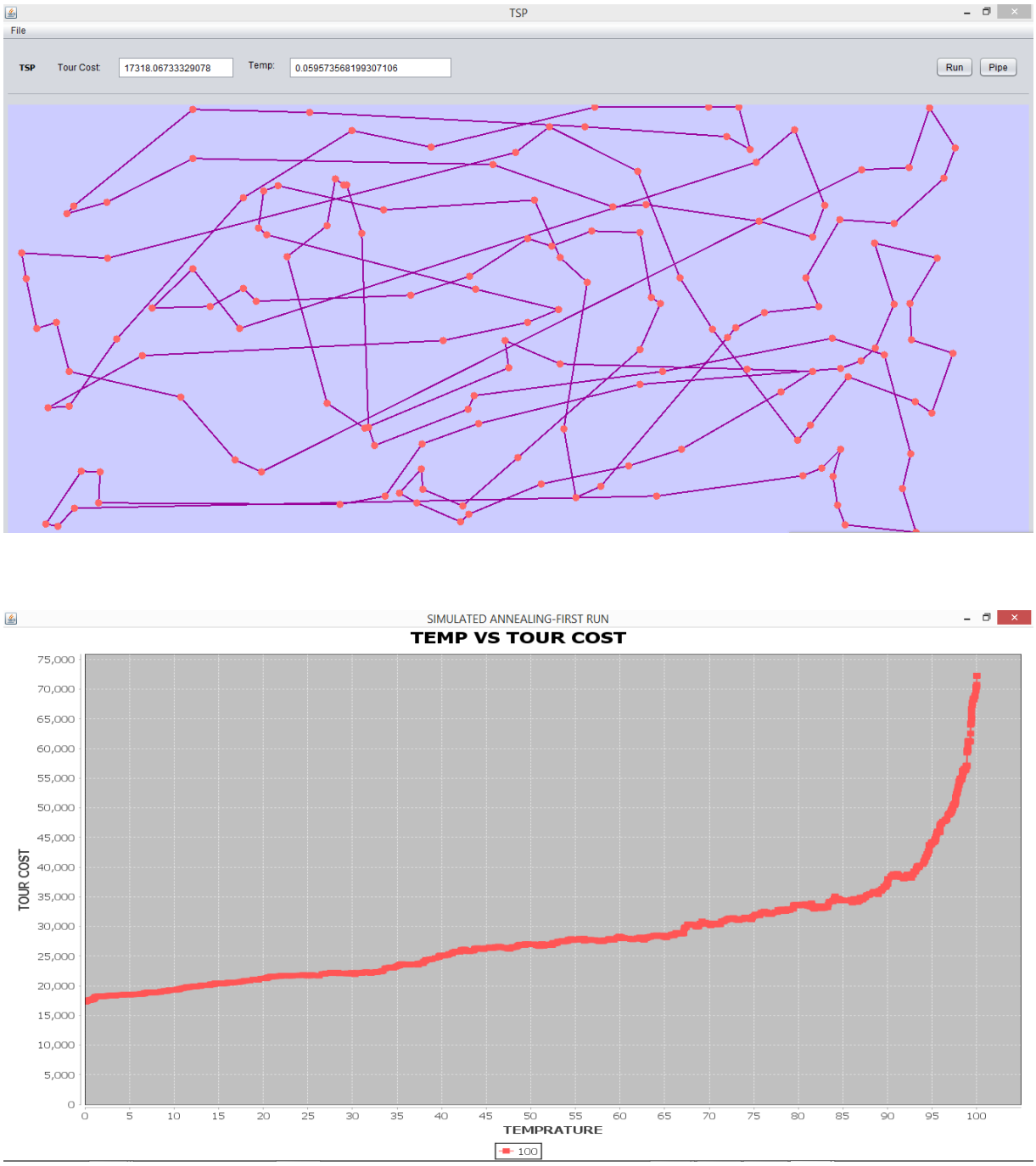

Fig 3. (i)Final tour (ii)Variation of tour cost with temperature in annealing process.

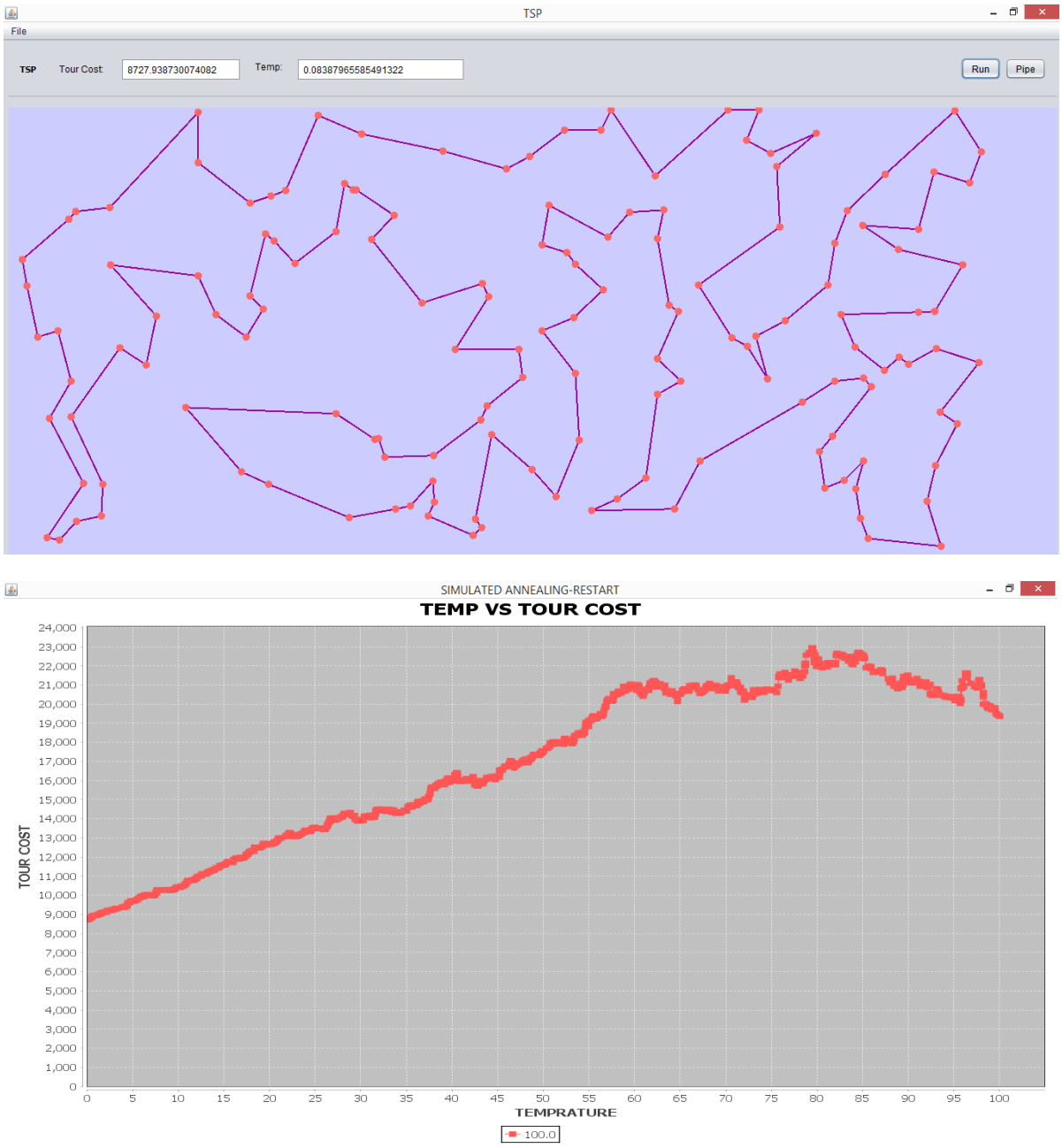

Fig 4. (i)Final tour (ii)Variation of tour cost with temperature in restart annealing process.

On observing the results that were achieved by using the city exchange and edge exchange perturbation we could see **that edge exchange performs better than the city exchange**. The application now successfully helps user in experimenting with the cooling schedule, perturbation methods on varied number of cities and visualize the tour as well as the change of tour cost with temperature for both annealing and annealing restart process.The temperature vs tour cost plot clearly shows that in starting the process is explorative and it is frequent to make moves which lead to worse tours,but near to freezing temperature only those tours are made that are better than the current tour.