CSI 5137 A: AI-enabled Software Verification and Testing
Homework Exercise 1

Wei Li 0300113733 report date: Oct/3/2020

1 Homework Description

The assignment requires to implement a solver for the Travelling Salesman Problem (TSP) using one of the metaheuristic search algorithms. Specifically, we consider TSPLIB instances that are symmetric in travel distance edges weights are set as Euclidean distances in 2D.

Deliverables and Environment

1.1 Deliverables

The homework was implemented with python. The files includes:

- TSP_solver.py –a script that solver TSP problem
- Readme.txt –instruction to use the script
- a280.txt –a test instance from TSPLIB
- HC-solution.csv –solution to a280 by hill climbing
- SA-solution.csv –solution to a280 by simulated annealing

1.2 Environment

You need the following environment and package to run the code.

- Python3
- Numpy
- Matplotlib

2 Assignment Report

2.1 A Quick Summary

In the assigment, we take the problem a280.tsp in TSPLIB as the test instance. After a quick evaluation of the total distance of the Sample_Solution we got a total distance of 3548.29.

We try the Steepest Hill Climbing algorithm to approach the problem at first, we test the algorithm for 100 runs and calculate the statistics summary of the results. The average total distance is 5681.9 with minimal 4918.4 and maximum 7537.8. About 70% of the solutions fall into the interval of [5000, 6000], indicating a common local optimum landscape that are not close to the global optimum.

We then implement the Simulated Annealing algorithm and test it on the same dataset for 10 runs. The average distance is 2776.0 and minimum and maximum are 2725.2 and 2826.0 respectively. This implementation of Simulated Annealing yield much better solutions comparing to the local search.

2.2 Metaheuristic Framework

The metaheuristic framework that were used in the implementation will be discussed in this section, while details that associate with each algorithms like hyperparameters setting will be put in their respective section. The 4 component in metaheuristics search framework are: representation, operators, fitness and constraints. When being applied to asymmetric TSP cases, they are:

2.2.1 representation

Assume all the customers in the TSP are assigned a reference number, the representation of a asymmetric TSP solution is a permutation of the reference number such that the customers are visited in the order of the solution.

2.2.2 operators

Operators are used to generate new solutions. We use 2-opt operator to generate neighborhood solution. The idea of 2-opt is simple. Give a solution to TSP problem, ie. a directed cyclic where each node is visited exactly once, we randomly exchange two edges in the solution such that the new solution is still connected. 2-opt operator returns a minimally changed neighborhood to a solution.

2.2.3 fitness

Naturally, we use the overall travel distance as the fitness. In fitness calculation, the cost of returning to the original node is considered.

2.2.4 constraints

Since we consider unconstrained asymmetric TSP, there is no constraint.

2.3 Hill Climbing

We implemented the Steepest Hill Climbing, where in each step we sample a group of 256 neighbors from an solution, and choose the solution with the largest fitness decrease as the new solution. The algorithm terminate when none of the 256 solution generates a better fitness.

We tested 100 runs of the algorithm, and obtained a distribution of the fitness. As shown in Figure 1.
All the result are inferior to the sample solution. Also, it shows that local search returns a large variance in the quality of solution.

2.4 Simulated Annealing

Simulated Annealing is implement in the following manner.

Step1 Generate an initial solution S
Step2 Select a value for the initial temperature $T_1 > 0$, Set the epoch count $k = 1$
Step3 Repeat the following L times (L is called the epoch length).
(1) Generate a neighborhood solution S' of S.
(2) Let $\Delta = Fitness\,(S') - Fitness(S)$
(3) If $\Delta < 0$, let S be S' (downhill move)
(4) If $\Delta \geq 0$, let S be S' with probability $\exp\,(-\Delta/T_k)$ (uphill move)
Step 4. If a given stopping condition is satisfied, stop. Otherwise, let $T_{k+1} = Cool\,(T_k)$ and $k = k + 1$ and go to step3.

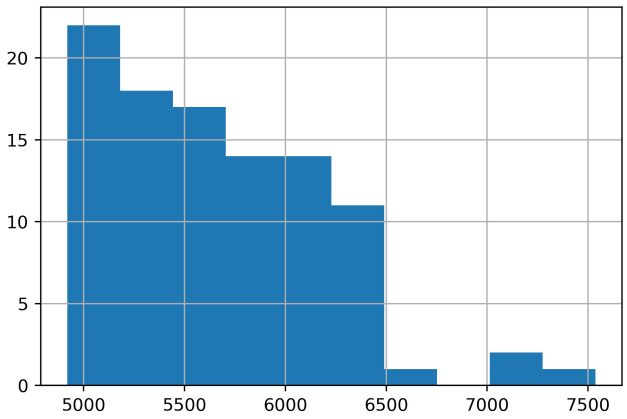

Figure 1: Steepest Hill Climbing Fitness(X axis), number of solution(Y axis)

$Cool\,(T_k)$ stands for a cooling function that slowly decrease the temperature. We use a simple geometric reduction rule: $Cool\,(T_k) = T_{k-1} * \alpha$ where α is a parameter.

We follow [1] to set the parameter α, L and T_1.

- [1] suggest that the initial temperature T_1 should be chosen in such a way that the fraction of accepted uphill transitions in a trial run of annealing process is equal to a given parameter P_1, called the initial acceptance probability. The initial temperature T_1 are then calculated as $T_1 = \bar{\Delta}/\ln P_1$, $\bar{\Delta}$ stands for an average of decrease of fitness function. According to the implementation by [2], we set P_1 as 0.4. The $\bar{\Delta}$ is -81.0 in 10000 trials. So the final T_1 is set to 88.4.
- L is suggested to set as proportional to the size of neighborhood. The size of neighborhood should be $\binom{280}{2} = 39060$ in the case of TSP with 280 nodes. So we set $L = 9765 = 1/4 * \binom{280}{2}$.
- α is set to 0.95 similar to the implementation by [2].

We run the solver on a280.tsp for 10 runs. The average distance is 2776.0 and minimum and maximum are 2725.2 and 2826.0 respectively. This implementation of Simulated Annealing yield much better solutions comparing to the local search.

References

[1] M.-W. Park and Y. Kim, "A systematic procedure for setting parameters in simulated annealing algorithms," *Comput. Oper. Res.*, vol. 25, pp. 207–217, 1998.

[2] D. Johnson, C. R. Aragon, L. A. McGeoch, and C. A. Schevon, "Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning," *Oper. Res.*, vol. 37, pp. 865–892, 1989.