Probabilistic Algorithms Project
Comparing heuristics for TSP

Consider a TSP instance with $N = 411$ 2-dimensional points, where the distance between two points with coordinates (x_1, y_1) and (x_2, y_2) is given by the Euclidian norm:

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Denote $L(\sigma) = \sum_{i=1}^{N} d(\sigma(i), \sigma(i+1))$ the length of the Hamiltonian cycle represented by the permutation $\sigma = (\sigma(1), \ldots, \sigma(N))$ of the N index[1]. The coordinates are stored in the file "TSP_411.txt", where each line has the form "index x-coordinate y-coordinate".

A. Implement the following optimisation heuristics.

- *Construction heuristics*:
  a) Best insertion;
  b) Best-Best insertion;
  c) Shortest edge;
- *Improvement heuristics*:
  1. Greedy local search using as small move: a) Swap; b) Translation; c) Inversion; d) Mixed move: all the previous moves, each one selected with a probability of 1/3.
     The maximum number of moves is set to $10 * N^2$ in each case;
  2. Simulated Annealing algorithm using the criterion : a) Metropolis; b) Heat bath condition.
     For each criterion, the new configuration is generated using one of the four variants of small moves. The initial T_{max} temperature is set to 100, $T_{min} = 1$, the cooling schedule is exponential with $T_{new} = 0.99 T_{old}$ and the number of moves at a fixed temperature is set to 1000.[2]

[1] You are free to use any type of representation for TSP (permutations, adjacent matrices, incidence matrices) if you consider appropriate for your implementation. In each case, $L(\cdot)$ returns the length of a Hamiltonian cycle.

[2] For better final solutions, you may increase the cooling factor (up to 0.999) or the number of moves at fixed temperature (up to 10000), but with the price of a longer runtime.

B. Perform a pairwise comparisons of algorithms using two-sample tests, for the following pairs:

a) [Best insertion, Best-Best insertion]; [Best insertion, Shortest edge];

b) Greedy [Swap, Translation], [Swap, Inversion], [Swap, Mixed], [Translation, Inversion], [Translation, Mixed], [Inversion, Mixed];

c) Simulated Annealing [Metropolis, Heat bath] (only with swap moves)

Denote $\mathcal{L}$ the sequence $L(\sigma_1^*), ..., L(\sigma_m^*)$ of the best solutions generated after calling each implemented algorithm $m = 30$ times[3]. The written report of the project must include:

- A summary table showing, for each implemented algorithm, the minimum of $\mathcal{L}$, the maximum of $\mathcal{L}$, the average of $\mathcal{L}$ and the bounds of the 95% confidence interval for this average.

- For each implemented algorithm, the plot of the best solution after $m = 30$ calls (i.e, the solution minimizing $\mathcal{L}$).

- Performance plots (based on a single call of the corresponding algorithm):

  – In the implementation of greedy local search algorithm, construct the vector `LS` of the values $L(\sigma_k)$, where σ_k is the solution obtained after k moves. Plot the vector `LS` vs. number of moves k, with $k = 1 \dots (10 * N^2)$, for each of the four variants of the algorithm.

  – In the implementation of simulated annealing algorithm, calculate, at each temperature T, the minimum, the maximum and the average of the values $\{L(\sigma_i)\}$, where $\sigma_i, i = 1 \dots 1000$, are all solutions processed at T. Plot these three values vs. current temperature T (where T belongs to $[1 \dots 100]$) for each of the eight variants of the algorithm.

Note 1. For each algorithm, the template of the corresponding MATLAB function must be

function [Tour] = *heuristic_name*(Nodes, Dist, ...)

where the two (compulsory) input parameters are `Nodes` (a matrix those rows contain the point coordinates) and `Dist` (the symmetric distances matrix)[4]. You are free to use any additional input parameter you consider necessary for implementation. The vector `Tour` represents the solution σ returned by the algorithm.

Note 2. A project team must have minimum two and maximum three members. For the teams with three members, the list of implemented algorithms also include the Saving heuristic.

[3] If your implementation of a particular algorithm takes too much time to execute, you may reduce m, but justify, in the written report, your choice.

[4] Of course, the Python' users may adapt the data structures of input parameters.