546U4030 SOFT COMPUTING METHODS AND APPLICATIONS

Coding Assignment 9 (2021)

ACO Optimization System for theTSPs

Refer to the course materials of Ant Colony Optimization techniques, develop an ACO optimization system for solving the traveling salesman problem. In particular, you are asked to implement the ACS.

Conduct necessary system requirement analysis before implementing your system: to design your data structured and user interfaces for solving the TSP. Follow the computation procedures of a meta heuristic optimization algorithm (referring to the GA that you implemented), design an ACS solver. You may follow the assignment structure of the GA solver to implement your ACO system. For example, you can design a GenericACO solver and an ACS system to derive it. Check the demo app to plan your interfacing functions, properties, and event to design your ACS class as well as any enums that facilitate the implementation.

You are given several text files consisting of the required data of a TSP. In your main form, you need to read in a TSP file and keep the coordinates of the cities and the known best routing sequence as well. In contrast, you can design a TSP class to deal with a TSP completely. Your system should have basic yet friendly user interfaces for problem loading, ACO parameter settings, and stopping condition settings, etc.

In addition to your source code, you need to submit a short video showing the computation capability of your ACS. In the video demo, please run the benchmark Taiwan50TSP and att48TSP.

Travelling Salesman Problems:

A travelling salesman need to visit n cities and exactly once for each. The planar Cartesian coordinates of the cities are given and the Euclidean distances between cities can be computed accordingly. A TSP is therefore, to find a sequence of city visiting that has the shortest length (including the distance from the lastly routed city back to the first one).

n : number of cities,

(x_i, y_i): Cartesian coordinates of city i ; $i = 0,1,\cdots,n-1$

$d_{i,j}$: distance from city i to city j ;

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$$\min f(\mathbf{z}) = d_{z_{n-1},z_0} + \sum_{i=0}^{n-2} d_{z_i,z_{i+1}}$$, where

$$\mathbf{z} = [z_0, z_1, \cdots, z_{n-1}], z_k \in \{0,1,\cdots,n-1\}, z_k \neq z_{k'}$$

In academics, the TSP and related problems (such as VRPs) had been studied several decades ago. There is a standard file format for specifying a TSP and related problem. Text file format is used with an extension name “tsp”. In addition, the known best routing sequence for the problem is also provided in text

format with extension opt.tour. See TSPLIB: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/. The file format is is complicated yet comprehensive to cover a variety of TSP-related problems.

We instead will use a concise text file for a TSP whose city coordinates are known and Euclidean distances are used for distance measurement. A sample file:

48	Number of cities = 48 // First row: one integer
1 6734 1453	City =1 Xcoordiante=6734 Ycoordiante=1453
2 2233 10	City =2 Xcoordiante=2233 Ycoordiante=10
3 5530 1424	City =3 Xcoordiante=5530 Ycoordiante=1424
…	…
47 5185 3258	// City47
48 3023 1942	// City48
1 8 38 31 44 … 12 15 40 9	1 8 38 31 44 … 12 15 40 9 // The best route

You need to read in the file and display the cities in a Chart with a point series. If the best route is given, is should be displayed using a line series. You should subscribe the SoFarTheBestSolutionIsUpdated event of your ACS solver to display your currently computed so-far the best route and its length as well.

You are encouraged to reuse your permutation GA solver for solving the same TSP

Class Design of A Sample Class: