IE 555 – Programming for Analytics

Homework #9 – OR Applications – Traveling Salesperson Problem

Due Date: Wednesday, May 13, at 5:00pm

Assignment Details

This assignment asks you to solve a traveling salesperson problem (TSP) using simulated annealing.

You are to provide a single Python script (`.py` file) containing a Python function named `solveTSP_SA()`, as described below. It is critical that you follow these instructions **exactly**.

1. The `solveTSP_SA(nodesDF, costDict, timeLimit)` function will accept three (3) inputs, in the following order:

   (a) `nodesDF` – A **VeRoViz "nodes" dataframe** containing n locations, numbered 1 through n. The TSP solution should visit each customer. You may assume that location 1 is the "home" location. Documentation on the dataframe structure may be found at `https://veroviz.org/docs/nodes.html`.

   (b) `costDict` – A **VeRoViz "time" dictionary** describing either the travel time, in [seconds], from each node to every other node. The contents of `costDict` will be generated by the VeRoViz `getTimeDist2D()` function (recall that this is an **input** to your function, so you won't need to calculate this dictionary). The travel time values will be generated using the "ORS-online" dataProvider.
   - Documentation on the "time" dictionary may be found at `https://veroviz.org/docs/veroviz.getTimeDist2D.html`.
   - Documentation on the "ORS-online" dataprovider may be found at `https://veroviz.org/docs/dataproviders.html`.

   (c) `timeLimit` – A **scalar value** describing the maximum runtime of your heuristic, in units of [seconds].

   The `solveTSP_SA()` function should return a **VeRoViz "assignments" dataframe**, as described here: `https://veroviz.org/docs/assignments.html`. You should make the following assumptions when solving your problem (and creating your "assignments" dataframe):

   (a) There is no service time at the nodes (i.e., the salesperson doesn't spend any time at any of the nodes; she simply visits each location instantaneously).

   (b) The salesperson must follow the road network, as determined by the "ORS-online" dataProvider. You'll need to register for an API key from openrouteservice, `https://openrouteservice.org/dev/#/signup`.

2. You should use the latest version of VeRoViz (0.4.0 or greater). To check that you have the latest version, use the following Python commands:

- `import veroviz as vrv`
- `vrv.checkVersion()`

Your function should be saved within a file named `UPPERCASEUBUSERID_tsp.py`, where `UPPERCASEUBUSERID` should be replaced with your UB username in ALL CAPS.

A GitHub "issue" has been created for you to ask questions. I will provide more details about using VeRoViz in our May 7 online class.

Grading

- If you submit properly-working code by the due date, you will have earned 100 points (the maximum score) on this assignment.
- UPDATED MAY 7, 2020: The TA and/or Instructor will evaluate your submission and will assign a grade. Given that this assignment is due during Finals Week, we will NOT be requiring/accepting re-submissions. So, do your best on the first (only) submission.

Submitting Your Assignment

A private GitHub repository will be created for you. Upload your code to the repository and then send an email to the TA when you are finished.