Traveling Salesperson Project

Math 242

due Friday, May 14

The traveling salesperson problem (TSP) is a classic combinatorial optimization problem. Given a set of N points, the problem is to find the shortest *tour* that visits each point once and returns to the starting point. It's very difficult to find the exact solution when N is big, but we can find approximate solutions using simulated annealing.

For this project, we will use N points sampled uniformly at random from the unit square $[0, 1] \times [0, 1]$.

Your Tasks

1. Implement a simulated annealing algorithm to find approximate solutions to the traveling salesperson problem. Set things up so you can easily vary the number of points N. You should be able to find good tours of several hundred points in reasonable amounts of time.

   Make plots showing the tours that you find for several different values of N. It should be clear from your plots that you are able to find near-optimal tours.

2. Investigate the following question: How does the approximate length of the near-optimal tour depend on N? Specifically, estimate the average lengths of near-optimal tours for different values of N. Make a conjecture, supported by computational evidence, of what function best describes how the near-optimal tour length depends on N.

Your Report

Turn in your investigation as a Python Colab notebook. As usual, submit code that runs and explain what your code does. Your goal should be to communicate your work to another person (e.g., another student at your level who is not in this course).

Grading Rubric

Your notebook will be graded on a scale of 0 to 16 points. The following rubric gives characteristics of notebooks that will merit sample point totals. (Interpolate the following for point totals that are not divisible by 4.)

16 points. Problems and goals are clearly stated, including relevant definitions or parameters. Computations are complete; code runs and is clearly explained. Conclusions are clearly stated and backed up by sufficient computational evidence. Limitations of the methodology, extensions for future work, and conjectures are discussed. Notebook is well-formatted and easy to read.

12 points. Problems and goals are stated well, though relevant definitions or parameters may be missing. Computations are mostly complete; code runs, but explanation is weak. Conclusions are unclear or not well justified. Insufficient discussion of limitations, extensions, and conjectures.

8 points. Statement of problem or goal is unclear. Computations are incomplete; explanation is ambiguous. Code may produce errors when run. Conclusions are possibly correct, but not justified. Little or no discussion of limitations, extensions, or conjectures. Notebook is difficult to read.

4 points. Serious misunderstanding of the problem or goal. Computation is inadequate for the task at hand. Work is not clearly explained. No discussion of limitations, extensions, or conjectures. Notebook is difficult to read.

0 points. Notebook is not turned in.