



## DEPARTMENT of COMPUTING

### College of Business & Technology

EAST TENNESSEE STATE UNIVERSITY

## CSCI 5260 – ARTIFICIAL INTELLIGENCE

### PROJECT 2 – GENETIC SEARCH

#### BUCKY'S QUEST

##### BACKGROUND

Bucky is almost 110 years old, but is having a terrible time convincing local public health officials that he is part of the current phase of the COVID vaccine rollout. He heard that physicians offices in Johnson City will administer the vaccine at some point, so Bucky has made it a daily task to visit each of the physicians offices in town to check on vaccine availability. Because he has so many responsibilities on campus, he needs to make his route as optimal as possible—resulting in the shortest distance. He will begin and end at ETSU's campus so he can attend to other things both before he leaves and after he returns.

##### PREPARATION

Download the **project2\_tsp.py** file from D2L.

##### NECESSARY LIBRARIES

1. Probably needed from pip install—
  - a. `osmnx` – A library that allows you to pull data from the Open Street GPS and store it as an undirected graph of GPS coordinates.
    - i. <https://osmnx.readthedocs.io/en/stable/>
  - b. `plotly.graph_objects` – `plotly` is a data visualization library, and the `graph_objects` class represents parts of a figure.
    - i. <https://plotly.com/python/>
  - c. `networkx`
  - d. `pandas`
2. Built-in Python libraries—
  - a. `csv` – used for CSV file reading/writing
  - b. `collections` – allows you to use python's double-ended and other queue structures.
  - c. `random` – the Python random number generator
  - d. `numpy` – the Python library for matrix manipulation and other scientific operations
  - e. `operator` – allows for targeted list sorting
  - f. `math` – for floor, ceiling, square root, etc.
  - g. `matplotlib.pyplot` (optional) – for providing additional data plot functionality.

##### REQUIREMENTS

You must complete the code for the following:

1. `run_ga()`
  - a. This should initialize the population and then repopulate until the max number of generations is reached.
2. `calculate_fitness()`
  - a. This code should determine the appropriate fitness for a particular chromosome.

- b. The chromosome should represent the ordering of the points list.
- 3. `initialize_population()`
  - a. This code will initialize an entire population (a single generation) of chromosomes.
  - b. The number of chromosomes to generate is controlled by the `POPULATION_SIZE` variable.
  - c. Each chromosome will be of size `len(points)`. Note that the chromosome will not contain the start/end locations.
  - d. You should calculate the fitness for each chromosome in the population. Add the chromosomes to the population in the format `[chromosome, fitness]`.
  - e. Sort the population based on the fitness and add it to the generations list.
    - i. `generations[x][0]` will be the best route for a particular generation x.
- 4. `repopulate(gen)`
  - a. Creates a new population for generation gen, by doing the following
    - i. `Selection(gen-1)`
    - ii. `Crossover(parent1, parent2)`
    - iii. `Mutate(child)` [at an acceptable mutation rate]
    - iv. Appends the child to the new population until full
  - b. Sorts the new population by fitness
  - c. Appends the population to the generations list.
- 5. `selection(gen)`
  - a. Selects and returns two parent chromosomes from generation gen (the generation number).
- 6. `crossover(p1, p2)`
  - a. Employs a crossover strategy on parents p1 and p2 (passed in the form of chromosomes).
  - b. Returns a child chromosome.
- 7. `mutate(chromosome)`
  - a. Accepts a chromosome that represents the ordering of the points list.
  - b. Returns the mutated version of the chromosome.

## PROVIDED FUNCTIONS

---

1. `plot_path(lat, long, origin_point, destination_point, fitness)`

Given (1) a path-ordered list of latitudes, (2) a path-ordered list of longitudes, (3) an origin point as a tuple in the format `(osmid, {"y": latitude, "x": longitude, "osmid": long_integer})`, (4) a destination point as a tuple in the format `(osmid, {"y": latitude, "x": longitude, "osmid": long_integer})`, and (5) the fitness score for the route, open a browser and display a map of the route.

2. `plot_ga()`

Plots the results of the genetic algorithm run, including the best and worst for each generation.

3. `haversine(point1, point2)`

Returns the Great Circle Distance in miles between point1 and point2. Both points should be an osmid.

4. `show_route(generation_number)`

Displays the route for generation\_number as a map. Note that this only depicts the haversine distances rather than navigation routing.

5. `main()`

Reads the file `addresses_geocoded.csv`, which corresponds to the addresses in `addresses.csv` and initializes the points array as nodes.

## SUBMISSION AND DUE DATE

Submit all code, zipped into a folder: project2\_tsp. The folder should be self-contained in a way that allows the code to run. You can assume that I have all necessary libraries installed.

Submit a written report to the dropbox separately that explains each of the following:

1. Your initialization and repopulation strategy
2. Your selection strategy
3. Your crossover strategy
4. Your mutation strategy

In the report, be sure to fully describe how your strategy work, and be sure to include why you believe it is better than other approaches.

**Project 2 is due to the D2L dropbox at or before Monday, February 22, 2020 at 11:59 PM**

## RUBRIC

A letter grade will be assigned to each of the following, and will translate to a numeric grade based on the scale in the syllabus, and averaged into an overall percentage. As a reminder, anything below a C translates to an F by University Graduate School policy. It is provided here to appropriately reflect each level.

For source code, please add comments so I can understand what is going on. Believe it or not, some student code is difficult to read. :D

	A		B			C			D		F	Zero
	A	A-	B+	B	B-	C+	C	C-	D+	D	F	0
<b>Bucky's Quest</b>												
run_ga												
calculate_fitness												
initialize_population												
repopulate												
selection												
crossover												
mutate												
Writeup/Discussion												