CAP 4630 – Dr. Marques **Summer 2023**

Project #3

Assignment Overview

In this assignment you will implement a nature-inspired approach to solving the classical Traveling Salesman Problem (TSP) using ant colony optimization (ACO) in Python.

Background

This assignment will give you a chance to build upon your understanding of TSP problems (see Project 2) and demonstrate your knowledge of swarm intelligence algorithms. See chapters 6 and 7 of the textbook for additional details.

Solving the TSP using swarm intelligence algorithms: recommendations

This assignment will focus on implementing a Python solution using ACO to solve the TSP. To help you navigate the design process and find good "sources of inspiration" out there, here are a few **recommendations**:

1. Start by revising your understanding of ACO (in general) and planning how to attempt to solve the TSP using ACO.
   - For example: How will your knowledge of the TSP (from Project 2) might help you tackle some aspects of this assignment more quickly than you would otherwise (i.e., if you had to start from a different problem altogether)?

2. Break the problem down into parts and **provide explicit answers to these questions in your report**:
   - How were the **cities and distances represented (as a data structure)?**
   - How did you encode **the solution space**?
   - How did you handle the **creation of the initial ant population**?
   - How did you handle the updating of the **pheromone trails**?
   - Which strategy(ies) did you use to compute the **best solution**?
   - Which **stopping condition** did you use? Why?
   - What **other parameters, design choices, initialization and configuration steps** are relevant to your design and implementation?
   - Which (simple) **experiments** have you run to observe the impact of different design decisions and parameter values? Post their results and your comments.

3. Use *good code* as starting point (and make sure to reference them accordingly in your report!)
   - Examples:
     - Code from the (GAIA) textbook (chapters 6 and 7)
     - https://github.com/yammadev/aco-tsp ("Implementing Ant Colony Optimization (ACO) algorithm for a given Symmetric traveling salesman problem (TSP)" by *Yefferson A. Marín Cantero*)
     - https://github.com/nishnash54/TSP_ACO ("Traveling Salesman Problem using Ant Colony Optimization")
     - https://github.com/rochakgupta/aco-tsp ("Solving Travelling Salesman Problem using Ant Colony Optimization")

4. If you're feeling adventurous, read this 2020 paper proposing a new/improved ACO algorithm for the TSP: https://www.atlantis-press.com/journals/ijcis/125932620

Project Specification

This is a group assignment.
Students are encouraged (but not required) to work in groups of max 3 students.

Ideally, the group should be organized around three main tasks / duties:

- Design of the solution ("architect" role)
- Coding of the solution ("developer" role)
- Documentation of the solution ("reporter" role)

You are required to indicate in your submission "who did what" and document the entire process, from sketching the original plans and dividing up the tasks all the way to polishing the interface, testing the solution, and preparing the report.

Deliverables

You must submit (via Canvas):

- The **link**[1] to a Jupyter notebook on Google Colab containing your entire solution. It must include:
  - Header:
    - Team members' names, date, course name + code, assignment number
  - Your source code
  - Results (of multiple runs) + meaningful comments
  - Plots
  - Figures
  - References (including your "sources of inspiration" for the code)
  - Comments (README-like): installation instructions, dependencies, etc.
  - Project notes (describing what my TA and I cannot see by looking at your source code and/or running your program).
    - Examples: design decisions, documented limitations, future improvements, etc.
  - Your answers to the questions in the previous page.

Notes and Hints:

- **Data for testing (same as Project 2):**
  - You should create a list of N cities (where the default for N = 25) within an (x, y) plane of 200-by-200 (think of them as km).
  - Cities should be placed randomly within this space.
  - Please ensure reproducibility by setting the seed of your random number generator to a fixed value of your choice.
- You are allowed to use numpy and matplotlib.

Bonus opportunities:

- There are no bonus opportunities for this assignment.

[1] When sharing the link to your Google Colab notebook, choose the 'anyone with the link can open it' option, i.e., **don't make it specific to a domain** (such as fau.edu) **or individual** (instructor or TA).