# MSc AI Lab: Second Assignement

Prof. Dr. Luca Maria Gambardella
Teaching assistants: Umberto Junior Mele

19 October 2023

## Homework 3 Ant Colony System for Travelling Salesman Problem

### Grade

The project grade, corresponding to 30% of your final grade, is given by a weighted average of your performance in the homeworks (10%, 25%, **30%** and 35%).

### Deadline

This assignment must be submitted by **23:59** (Lugano's time) on **6th December 2023**.

### Instructions

Your task is to solve the Travelling Salesman Problem (TSP) using an Ant Colony System (ACS). Reference papers for implementation details include "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem" and "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows". The implementation will be based on a provided notebook and should be tested following the rules below. Implement the following ACS *variants*, enhancing each step:

1. **Starting Point**: Use the notebook available at `ACS_for_the_TSP.ipynb` as your base.
2. **Target Problems**: Implement the ACS to search for the global minimum of the TSP instances: eil76, ch130, d198 (small size problems).
3. **ACS variants**: Test the ACS using the following three variants:
   - Basic Ant Colony System (ACS)
   - ACS enhanced with 2-opt
   - ACS with 2-opt and candidate list
4. **ACS Settings**: Configure your ACS using the following settings:
   - Number of ants ($m$): 10
   - Importance of trail ($\beta$): 2
   - Trail persistence ($\alpha$) and pheromone evaporation rate ($\rho$): 0.1
   - Initial pheromone level ($\tau_0$): $(n \cdot L_{nn})^{-1}$
   - Probability ($q_0$) values: 0.5, 0.98, $(1 - \frac{13}{n})$
   - "Candidate List (cl): You have the option to use either a fixed candidate list size of 15, or opt not to use a candidate list at all. Additionally, an alternative candidate list strategy based on the POPMUSIC heuristic will be provided in the next lesson
   - Global updating rule: Global best approach
   - Use 2-opt only for the best ant in each iteration

5. **Output Requirements**: Each implementation should not exceed a runtime of 3 minutes per combination of ACS variant, instance, and seed. For each combination, present the following:
   - For each run: The best gap and cost achieved, and the number of tours generated to achieve the best tour.
   - For each variant: A table mirroring the results of TABLE IV in the "ACS" paper.
   - For each problem instance: A plot showing the average of the three runs and the best gap achieved for each run, with the gap on the y-axis and iteration number on the x-axis.

### Submission Guidelines

1. **File Type**: Submit as a Python Jupyter notebook.

2. **Content**: Discuss your findings briefly and provide evidence for each tasks completed.

3. **Exclusions**: Do not include plot images or files from the `AI2022MA` directory.

4. **File Naming**:
   - For a single file: <Name Surname>_MScAI23_hw3.ipynb
   - For multiple files: Compress them into a folder named <Name Surname>_MScAI23_hw3.zip

### Evaluation Criteria

Your submission will be assessed based on the following:

1. **Correctness**: Implementation should correctly apply the Ant Colony System algorithm and its variants to the Travelling Salesman Problem.

2. **Analysis**: Ability to analyze and interpret the results, including the efficiency and effectiveness of different ACS variants.

3. **Clarity**: Code and accompanying documentation should be clear and well-organized, facilitating easy understanding and evaluation.

4. **Presentation**: Quality of the graphical representations and tables, and how well they communicate the results.

### Additional Notes

- Ensure that your code is well-commented and follows good programming practices.

- Extra credit may be awarded for innovative enhancements or particularly insightful analysis.

- Remember, this homework is designed not only to test your technical skills but also your ability to conduct and present research in a clear and concise manner.