1 Lab + Hwk 2: Ant Colony Optimization

This laboratory requires the following equipment:

- Swislab (Linux)
- Matlab (Linux)

The laboratory duration is about 3 hours. Homework will be due on the 7th day after your lab session, at 12 noon. Please format your solution as a PDF file with the name [name]_lab[#].pdf, where [name] is your account user name and [#] is the number of the lab+homework assignment, and upload it as the Report submission onto the student section of the course webpage. All additional material (movies, code, and any other relevant files) should be archived as a zip file with the name [name]_lab[#].tgz (hint: `tar cvzf [name]_lab[#].tgz [directory]`) and uploaded as the Additional Material submission onto the student section of the course webpage. Every misnamed file will result in a 1 point penalty to your assignment grade.

Keep in mind that no late solutions will be accepted unless supported by health reasons (and a note from the doctor). If there are extreme circumstances which will prevent you from attending a particular lab session, it may be possible to make arrangements with the TAs before the lab in question takes place. The more advance notice you can give in these situations, the more likely it is that we will be able to work something out.

1.1 Grading

In the following text you will find several exercises and questions.

- The notation $S_x$ means that the question can be solved using only additional simulation; you are not required to write or produce anything for these questions.
- The notation $Q_x$ means that the question can be answered theoretically, without any simulation; your answers to these questions must be submitted in your report. The length of answers should be approximately two sentences unless otherwise noted.
- The notation $I_x$ means that the problem has to be solved by implementing a piece of code and performing a simulation; the code written for these questions must be submitted in your Additional Material.

Please use this notation in your answer file!

The number of points for each exercise is given between parentheses. The combined total number of points for the laboratory and homework exercises is 100.

1.2 Ant Colony Optimization and the Traveling Salesman Problem

Ant Colony Optimization (ACO) is a meta-heuristic that allows solving a suite of hard optimization problems by using the ant colony/trail laying metaphor [Dorigo2004]. It is inspired by the optimization capabilities of foraging ants as it can be observed in the bridge experiments of J.L. Deneubourg.
The Traveling Salesman Problem (TSP) is a classical optimization problem. It deals with finding the shortest path that connects a number of cities, and passes every city once and only once. Due to its immediate connection to the shortest path problem that ants face during foraging, ACO has been first tested on a TSP problem.

In this lab you will familiarize yourself with different ACO inspired algorithms to solve the TSP, and observe and evaluate their performance experimentally.

## 2 Basic Ant System

We can simulate the ants used in Deneubourg’s experiment in a computer, and use them to solve shortest path problems that are more complex than the double bridge experiment. For that, a shortest path problem can be represented by a graph where edges are suitable paths, and nodes are junctions (Figure 1). Now, we can imagine ants deploying pheromones while traversing the graph by choosing an edge with high pheromone density over an edge with little pheromones.

![Figure 1: A simple shortest path problem represented as a graph. Edges are possible paths, nodes are junctions where decisions need to be made.](image)

**Q1(2):** Consider the graph given in Figure 1. “Simulate” trail laying ants using a pen and paper and assume that 1) an ant always deploys one pheromone on each edge it passes, 2) an ant always chooses the edge that holds the most pheromones, and 3) in the case of no pheromones, the ant chooses a new edge randomly. Let one ant walk at a time. What kind of deadlock might your colony eventually run into?

**Q2(2):** How could such a deadlock be avoided? What additional capabilities would your artificial ants need?

### 2.1 Ant System

Ant System (AS) is the original ant-foraging inspired algorithm presented in Marco Dorigo’s PhD thesis in 1992. In AS, \( m \) ants travel from a random starting point from city to city until all \( n \) cities have been visited. Hereby, paths are chosen randomly with
a probability which is a function of the amount of pheromones already deposited and of the distance to the next city. The probability for an ant \( k \) at city \( i \) to go to city \( j \), is then given by

\[
p^k_{ij}(t) = \frac{Q_{ij}}{\sum_{l \in J_i^k} Q_{il}} \quad \text{with} \quad j \in J_i^k,
\]

which is the feasible neighborhood for ant \( k \) in city \( i \) and \( Q_{ij} \) a combined metric of the quality of the route. In AS, the quality of a route \( ij \) is a function of the pheromones already deposited by other ants given by \( \tau_{ij} \), and a heuristic \( \eta_{ij} = 1/d_{ij} \), with \( d_{ij} \) the distance between city \( i \) and city \( j \). Thus, \( Q_{ij} \) is defined as

\[
Q_{ij} = \left[ \tau_{ij} \right]^\alpha \left[ \eta_{ij} \right]^\beta
\]

with \( \alpha \) and \( \beta \) allowing us to fine tune the impact of pheromones and heuristic information on the metric.

Figure 2: A simple instance of a TSP. Edges are marked with pheromones deposited by ants in previous iterations of the AS algorithm.

After completion of a tour, i.e. arriving at the starting city, ants assess the tour, and deposit an amount of pheromones that is inverse proportional to the tour length on every link \( ij \) they visited, i.e. \( \Delta \tau_{ij}^k = 1/C^k \) with \( C^k \) the total length of the last tour.

**Q3(3):** Consider the TSP in Figure 2. Edges are marked with pheromones (the numbers next to the edges) that have been deposited during previous tours of one ant, which is now sitting at the black vertex. What will be the route that is most likely taken by the ant (assume heuristic information not to be available/constant)? Your solution should be a sequence of integers.
Q₄(3): Again consider the TSP in Figure 2. Ignore now the pheromone values, and let the ant decide solely based on heuristic information. What will be the route that is most likely taken by the ant in this case? Keep in mind that the heuristic information associated with an edge is anti-proportional to the Euclidean distance in the Figure. Your solution should be a sequence of integers.

S₅: Open Matlab and type in the commands

```
cities=randommap(6)
tsp('random',cities)
```

which create a matrix with 6 cities at random locations (2D coordinates), and solves the TSP by creating a random tour. You can also type `load eil51`, which will load a `cities` structure with 51 cities whose optimal value is known to be 246 [TSPLIB].

I₆(12): Implement the missing code in the function `solvetsp_greedy` (within `tsp.m`). Instead of choosing a city randomly, your algorithm should choose the closest city from the list of unvisited cities. Test your solution by running

```
tsp('greedy',cities)
```

for different scenarios created using `randommap()` for a small number of cities, and qualitatively verify the result using the `eil51` map. `tsp.m` should be the only file you submit along with your solution!

I₇(12): The function `solvetsp_ant` provides a skeleton of an Elitist Ant System (EAS). In EAS only the best ant of a tour is allowed to deploy pheromones. Implement the missing two lines that allow an ant to calculate the probability for choosing the next city according to the equation above. Test your solution by running

```
tsp('ant',cities)
```

S₈: You can now compare the different approaches (random, greedy, and EAS) using the command `testall(cities)`

Q₉(4): Revisit Figure 2 and imagine an ant constructing a solution for the TSP using EAS given the current pheromone table. What happens when a) an entry in the pheromone table reaches zero, and b) an entry in the pheromone table gets much higher than others?

Q₁₀(4): How could you prevent the problems observed above? At which step of the algorithm would you implement this feature in the function `solvetsp_ant`? (Qualitative answer, no line numbers).
Q₁₁(3): Observe the number of tours that EAS needs for finding the best solution (you can increase the number of tours using the parameter \texttt{ntours}). What happens? What could you do to improve the result?

Q₁₂(2): The TSPs considered so far were fully connected, i.e. every city could be reached from every other city, leading to a quadratic distance matrix. How would you need to modify this matrix in order for taking into account cities that are not directly connected?

2.2 Local Search

Whereas constructive algorithms like the EAS and its derivatives are constructing solutions in an incremental way using some heuristic rules, Local Search algorithms (LS) optimize a given solution by iteratively exploring neighborhoods of the solutions that are given by local changes.

Q₁₃(4): Encode both tours of Figure 3 as a character string and compare them. What happens to the order of a substring when you swap two edges?

I₁₄(15): A skeleton implementing local search is provided in the function \texttt{local_search_2opt}. Implement a 2-opt local search, which systematically evaluates all permutations of a solution that can be achieved by exchanging two edges. Test your solution with \texttt{testall(cities,1)}, the second parameter enables local search in all of the above algorithms.

S₁₅: Start the application \texttt{swislab} and open some of the larger instances in the \texttt{instances} folder (e.g., att532 or pcb1173). The application allows you to choose various implementations of AS and ACS and test their performance.

Q₁₆(4): Open the instance eil51.tsp. What parameters do you have to choose for ACS in order to get to the optimum of 425?

Q₁₇(3): How is the use of 2-opt local search different in ACS from the simple approach you used in the Matlab code above?
3 Theory and Homework

Q_{18}(4): The pheromones of the ants in Deneubourg’s experiment evaporate very slowly, and thus ants cannot respond to changes in the environment at all. Can you think of another benefit that evaporating pheromones might yield? What is the drawback if the evaporation rate is too high?

Q_{19}(2): When or for which kind of problems do you need a high evaporation rate?

I_{20}(15): Think about ways of improving the EAS skeleton in TSP.M and implement them. Use `load eil51` for loading a map of 51 cities. You should get down systematically to fitness below 440 by implementing the improvements discussed in the questions above.

Q_{21}(2): Consider a routing problem in a large IP network. What are the main difficulties/differences when applying an ACO algorithm for the TSP for finding shortest routes in the network?

Q_{22}(3): The network was established/re-configured only recently, and routing tables are incomplete. Imagine router 4 sending off a forward ant towards node 1. Assuming the queue length at the routers being not available/constant, which is the route the forward ant will most likely take, and what will the entry in the routing table of router 4 be in this case?

Figure 4. A simple network with 5 routers. Links are marked with the current latency that packets will face on a link. The routing tables contain the expected latency (L) to a certain goal via a certain link from previous runs of forward ants.
Q23(4): Imagine again router 4 sending off a forward ant towards router 1. Describe two possible scenarios that would lead to better routing information based on a) an action of another router in the past, and b) a decision of the forward ant.

4 References

[TSPLIB] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.