**Assignment 4: Genetic Algorithm and Ant Colony Optimization (ACO)**

Student: Mirza Sijamić

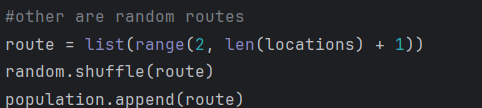
Student ID: msc25001

1. Explain the important operations of the employed algorithm. (0.7)

Four key operations in a genetic algorithm are selection, crossover, mutation, and elitism. In my version. Selection refers to the method of picking parents that will reproduce to produce offspring. I utilize a tournament style technique, for each parent, five individuals are randomly chosen from the population, and the one with the shortest route is selected. This guarantees that superior routes have an increased likelihood of being chosen. Crossover is the action that combines the genes of two parents to generate new offspring. The technique I implement is similar to Order Crossover. Two random crossover points are selected and the segment between these points from the first parent is directly transferred to the child. The positions in the child are then filled with genes from the second parent, maintaining their order from the parent and ensuring that no city is duplicated. Mutation adds variety by modifying an offspring slightly so that it contrasts with its parents. I apply an inversion mutation. With a mutation chance of 2%, two cities in the route are randomly picked and exchanged. This alteration can assist the algorithm in escaping optimal and investigating other solutions. Each candidate solution is evaluated using a fitness function based on the total route distance. The algorithm iterates through many generations, updating the best found solution. It terminates when the best distance falls below a preset threshold (in this case, 9000) or when the maximum number of generations is reached. Algorithm consistently tracks and updates the best route discovered. This method preserves the highest quality solution throughout generations, ensuring that the best outcomes are not lost during crossover or mutation.

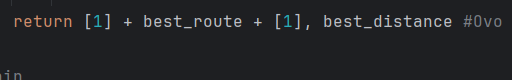
1. Explain the representation of the individual, a solution to the problem, in your algorithm (0.7)

An individual in this algorithm is represented as a list of integers, each corresponding to a city (excluding the starting city, which is fixed as city 1).



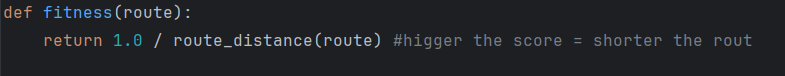
Here, route is a permutation of cities 2 through n, representing the order in which the cities are visited.

Only when reporting the final route is the first city (city 1) added to both the beginning and the end of the list. This ensures that the route starts and ends at the same city.



1. Give the equation of the fitness function used by your algorithm. (0.45)

The fitness function used in the algorithm is defined as:



Fitness function used in the solution is f(x)=1/p(x), where p(x) is sum of distances between cities in a single route. Distance between two cities is calculated using the Cartesian plane distance formula d=√((x1-x2)2+(y1-y2)2).

1. Give the parameters used in your algorithm. Examples: Population size, crossover rate.. (0.45)



 Mutation rate is 2%

ACO

1. Explain the important operations of the Employed algorithm. (0.5)

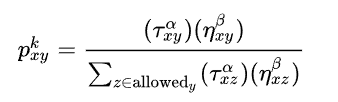
The main operations in the Ant Colony Optimization algorithm are as follows:

Pheromone Initialization is Initially, a small, uniform amount of pheromone is set on all edges between cities. Each ant starts at a designated node (or city) and builds a complete tour by moving from one city to the next. The probability of choosing the next city depends on two factors:

Pheromone Level**:** A higher pheromone value increases the likelihood of that edge being chosen.

Heuristic Information (Visibility): Usually the inverse of the distance between cities, which favors closer cities.

The probability for an ant at city i to move to city j is given by:



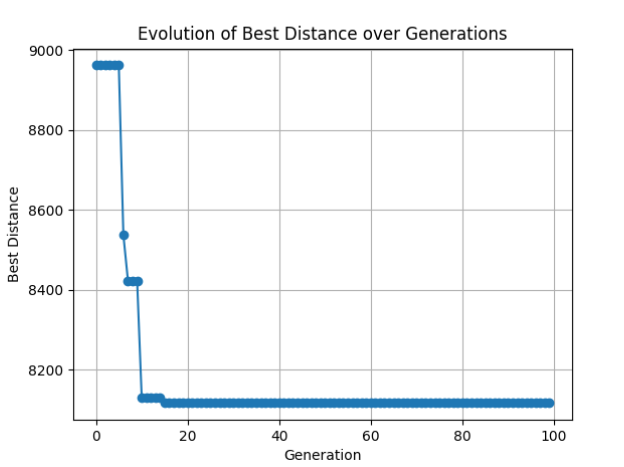
where α and β are parameters controlling the influence of the pheromone and distance, respectively.

Source: https://en.wikipedia.org/wiki/Ant\_colony\_optimization\_algorithms  
Once all ants have constructed their routes, pheromones on all edges are updated:

Evaporation**:**  
A fixed percentage of the pheromone evaporates, which prevents unlimited accumulation and helps the algorithm avoid premature convergence.

Deposit**:**  
Each ant deposits pheromone on the edges it traveled, with the amount often inversely proportional to the tour’s total distance. Better (shorter) routes receive a higher deposit, reinforcing good paths.  
The process of constructing solutions and updating pheromones repeats for a set number of iterations (or until convergence). Over successive iterations, the pheromone trails gradually guide the ants toward increasingly promising regions of the search space.

1. Illustrate how the performance of the population evolves with generations (with a figure.) Write text with the figure as well. (0.5)



In the early generations (the left side of the plot), the ACO algorithm quickly discovers much shorter routes, causing the best distance to drop sharply from around 9000 to roughly 8200. This happens because the ants rapidly reinforce good paths through pheromone updates. After about 10–20 generations, the improvements become very small, and the plot flattens out near 8200, indicating the algorithm has converged to a near optimal solution with only minor refinements thereafter.

1. Compare the results found by this algorithm with the results found with GA. Analyse the differences between the algorithms and the difference in the performance between both. (1)

Search Mechanism:

* ACO**:**  
  Uses indirect communication through pheromone trails and probabilistic path construction. More pheromone on good routes accelerates the search toward promising areas. If the pheromone update is too aggressive, the algorithm might converge prematurely to a suboptimal solution.
* GA**:**  
  Relies on operators such as crossover to combine features from different solutions, and mutation. This mechanism helps maintain diversity and explore the search space more broadly, sometimes at the cost of slower convergence.

Convergence Behavior:

* ACO**:**  
  Often shows rapid early convergence as good paths are quickly reinforced. The performance typically improves sharply in early generations and then stabilises.
* GA**:**  
  Convergence is more gradual because the recombination of individuals allows exploration of a larger variety of solutions. The balance between exploration and exploitation depends on the crossover and mutation rates.

Performance and Robustness:

* ACO**:**  
  Works very well for problems like the Traveling Salesman Problem (TSP), where the reinforcement of good paths can directly guide the search. However, its performance is sensitive to parameter settings such as pheromone evaporation rate.
* GA**:**  
  Can be more robust across various problem types due to its ability to maintain diversity. It might be less sensitive to local optima because the crossover operator can mix information from different solutions. For problems where local search is key, GA might require additional refinement techniques to reach the best possible solutions.