|  |  |
| --- | --- |
| Yomna Taher Abdallah | 20190624 |
| Adel Abdelmonem Arfa | **20190280** |

**Task 2**

1. **Initialization:**

**Generate a distance matrix that represents the distances between all pairs of cities.**

**Set the parameters for the ACO algorithm, such as the number of ants, number of iterations, alpha, beta, rho, and q (pheromone deposit factor).**

1. **Ant Construction:**

**Create a few ant agents. Each ant keeps track of its current tour, visited cities, and the total distance traveled.**

**Randomly select a starting city for each ant and mark it as visited.**

**For each ant, construct a tour by choosing the next city to visit based on a combination of pheromone trails and heuristic information (related to the distance between cities). This decision-making process is influenced by the alpha and beta parameters.**

**Update the visited cities and accumulate the total distance traveled by each ant.**

1. **Pheromone Update:**

**Calculate the delta pheromone for each edge in the tours constructed by the ants. The delta pheromone is proportional to the quality of the tour (shorter tours contribute more pheromone).**

**Update the pheromone levels on each edge by evaporating a certain percentage (rho) of the existing pheromone and depositing the delta pheromone.**

**The amount of pheromone deposited is determined by the quality of the tour and the q parameter.**

1. **Iteration:**

**Repeat steps 2 and 3 for a fixed number of iterations.**

1. **Output:**

**Return the best tour found and its corresponding distance.**

**Regarding the chosen distances between the cities in each given configuration, the distances are generated randomly using the “generate\_distance\_matrix” function. The distances represent the distances between pairs of cities in the problem. The distances are symmetric, meaning the distance from city A to city B is the same as the distance from city B to city A. These distances are used by the ACO algorithm to calculate the heuristic information and update the pheromone levels during the execution of the algorithm.**

**Pheromone map of 10 cities:**

A picture containing circle, symmetry, diagram, line

Description automatically generatedA picture containing circle, symmetry, diagram, line

Description automatically generated

A picture containing circle, symmetry, diagram, line

Description automatically generatedA picture containing circle, diagram, line, symmetry

Description automatically generated

A picture containing circle, symmetry, line, diagram

Description automatically generated

A picture containing circle, art, drawing, pattern

Description automatically generated**Pheromone map of 20 cities:**

A picture containing circle, art, drawing, pattern

Description automatically generated

A picture containing circle, art, pattern, drawing

Description automatically generatedA picture containing circle, drawing, screenshot, art

Description automatically generated

A picture containing circle, art, pattern, drawing

Description automatically generated

**The results relating to the set of 10 cities:**

A picture containing text, font, screenshot, design

Description automatically generated

**The results relating to the set of 20 cities:**

A screenshot of a computer

Description automatically generated with medium confidence

**Conclusion:**

**In the 10 cities, the algorithm start with good distance and then get increase the distance until it cover all cities then it use the best one (the most minimum), on the other hand in the 20 cities it start also with good way and keep discover ways but in the final after the roads are increased it didn’t use the best option, so I think it my be not good when I increase the cities, It won’t to use the best one in the final state.**