Assignment 2: Swarm Intelligence
CS 451: Computational Intelligence

Ali Asghar Yousuf | Muhammad Murtaza

March 18, 2023

1 Ant Colony Optimization

Capacitated Vehicle Routing Problem

1.1 Problem Statement

The Capacitated Vehicle Routing Problem (CVRP) is a variant of the Vehicle Routing Problem (VRP) in which each vehicle has a limited capacity. The problem is to find a set of routes for a fleet of vehicles to visit a set of customers, each with a demand, and return to the depot. The objective is to minimize the total distance traveled by all vehicles. The problem is NP-hard and is known to be computationally intractable for large instances. The problem is a special case of the Traveling Salesman Problem (TSP) in which the salesman has to visit a set of customers and return to the depot.

1.2 Solution Approach

We have used Ant Colony Optimization (ACO) to solve the Capacitated Vehicle Routing Problem. ACO is a metaheuristic optimization algorithm that is inspired by the behavior of real ants. The algorithm is based on the following steps:

1. Initialization
2. Ants' construction
3. Pheromone update
4. Termination

1.2.1 Initialization

The first step is to initialize the parameters of the algorithm. The parameters are:

1. α - Controls the importance of pheromone
2. β - Controls the importance of heuristic information
3. γ - Controls the importance of the capacity constraint
4. `numAnts` - Number of ants
5. ρ - Evaporation rate
6. `iteration` - Number of iterations
7. `capacity` - Capacity of each vehicle
8. τ - Pheromone matrix
9. η - Heuristic matrix
10. `distances` - Distance matrix
11. `demand` - Demand matrix

1.2.2 Ants' construction

In this step, the ants are initialized with random routes in the first iteration and in the next iterations they start constructing their routes. The ants are initialized at the depot and they move to the next node based on the following formula:

$$P(i,j) = \frac{\tau_{i,j}^{\alpha}\eta_{i,j}^{\beta}}{\sum_{k\in\texttt{allowed}} \tau_{i,k}^{\alpha}\eta_{i,k}^{\beta}} \tag{1}$$

where $\tau_{i,j}$ is the pheromone value of the edge between node i and j, $\eta_{i,j}$ is the heuristic value of the edge between node i and j, α and β are the parameters of the algorithm, and `allowed` is the set of nodes that are allowed to be visited by the ant. The ant moves to the next node based on the probability $P(i,j)$. The probability is calculated by taking the pheromone and heuristic values of the edges and then normalizing them. The ants move to the next node until they reach the depot or the capacity of the vehicle is exceeded. The ants then return to the depot and start constructing their routes again. This process is repeated for all the ants.

1.2.3 Pheromone update

In this step, the pheromone values of the edges are updated. The pheromone values are updated based on the following formula:

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \rho\tau_{i,j}^{0} \tag{2}$$

where $\tau_{i,j}^{0}$ is the initial pheromone value of the edge between node i and j, and ρ is the evaporation rate of the algorithm. The pheromone values are updated after each iteration.

1.2.4 Termination

The algorithm terminates when the number of iterations reaches the maximum number of iterations. The best solution is the one with the least distance in all of the iterations.

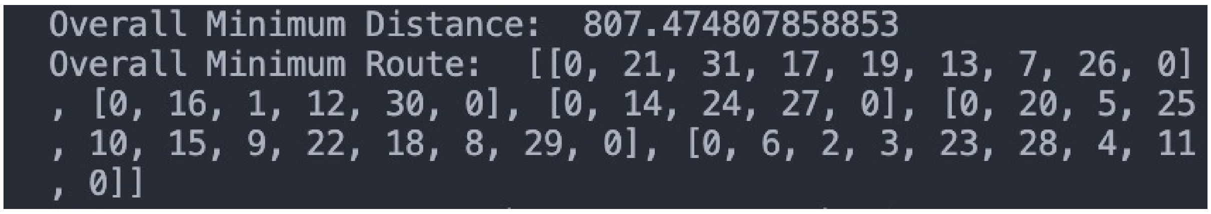
1.3 Optimal Solution

After tweaking the parameters, and trying several different combinations we got the optimal values with the following parameters:

1. $\alpha = 4$
2. $\beta = 4$
3. $\gamma = 0.5$
4. `numAnts` $= 30$

1.3.1 A-n32-k5

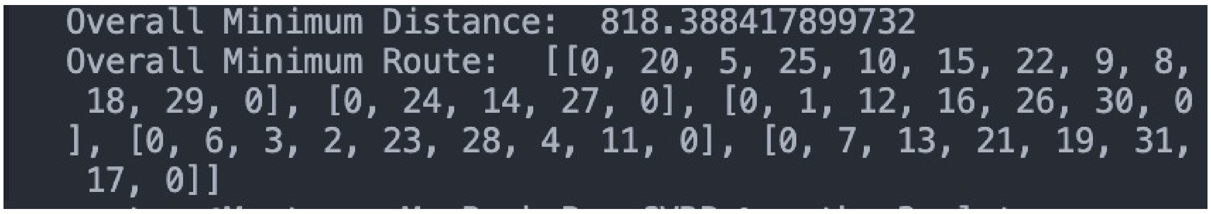
The minimum distance that we could achieve for this dataset was 807.475. The following is the route that we found:



```
Overall Minimum Distance:  807.474807858853
Overall Minimum Route:  [[0, 21, 31, 17, 19, 13, 7, 26, 0]
, [0, 16, 1, 12, 30, 0], [0, 14, 24, 27, 0], [0, 20, 5, 25
, 10, 15, 9, 22, 18, 8, 29, 0], [0, 6, 2, 3, 23, 28, 4, 11
, 0]]
```

Figure 1: Minimum Distance Route

Note: We forgot to plot this result, so the minimum distance for which we have a plot is 818.388



```
Overall Minimum Distance:  818.388417899732
Overall Minimum Route:  [[0, 20, 5, 25, 10, 15, 22, 9, 8,
 18, 29, 0], [0, 24, 14, 27, 0], [0, 1, 12, 16, 26, 30, 0
], [0, 6, 3, 2, 23, 28, 4, 11, 0], [0, 7, 13, 21, 19, 31,
 17, 0]]
```

Figure 2: 2nd Minimum Distance Route

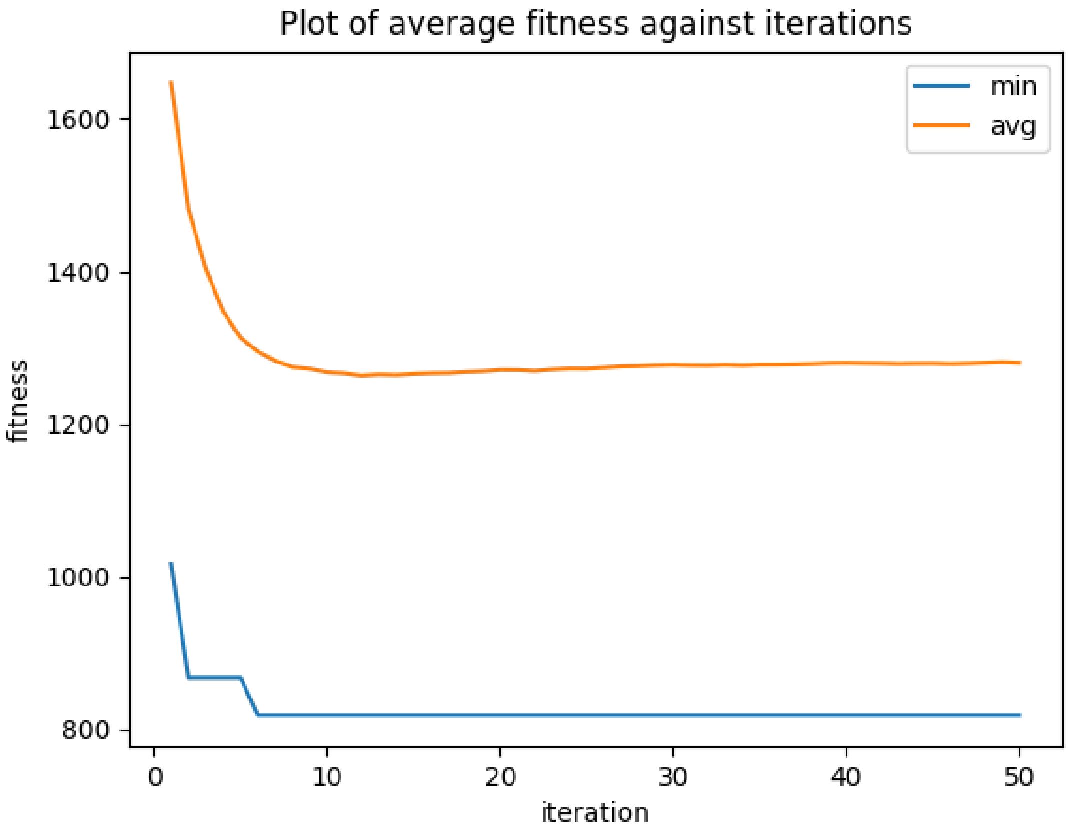
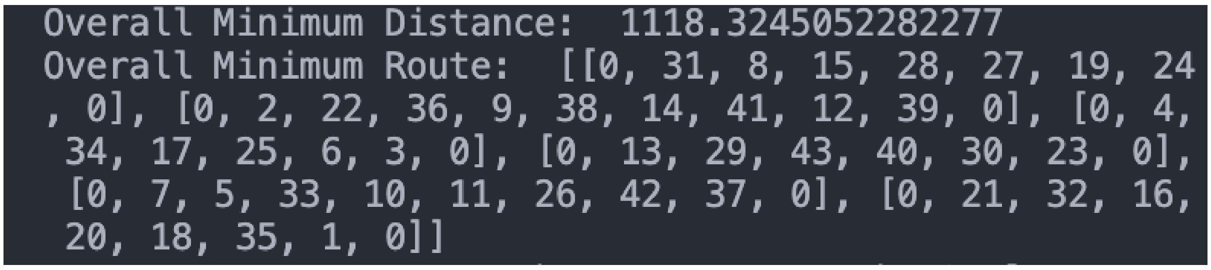


Figure 3: 2nd Minimum Distance Plot

1.3.2 A-n44-k6

The minimum distance that we could achieve for this dataset was 1118.325. The following is the route that we found:



```
Overall Minimum Distance:  1118.3245052282277
Overall Minimum Route:  [[0, 31, 8, 15, 28, 27, 19, 24
, 0], [0, 2, 22, 36, 9, 38, 14, 41, 12, 39, 0], [0, 4,
 34, 17, 25, 6, 3, 0], [0, 13, 29, 43, 40, 30, 23, 0],
 [0, 7, 5, 33, 10, 11, 26, 42, 37, 0], [0, 21, 32, 16,
 20, 18, 35, 1, 0]]
```

Figure 4: Minimum Distance Route

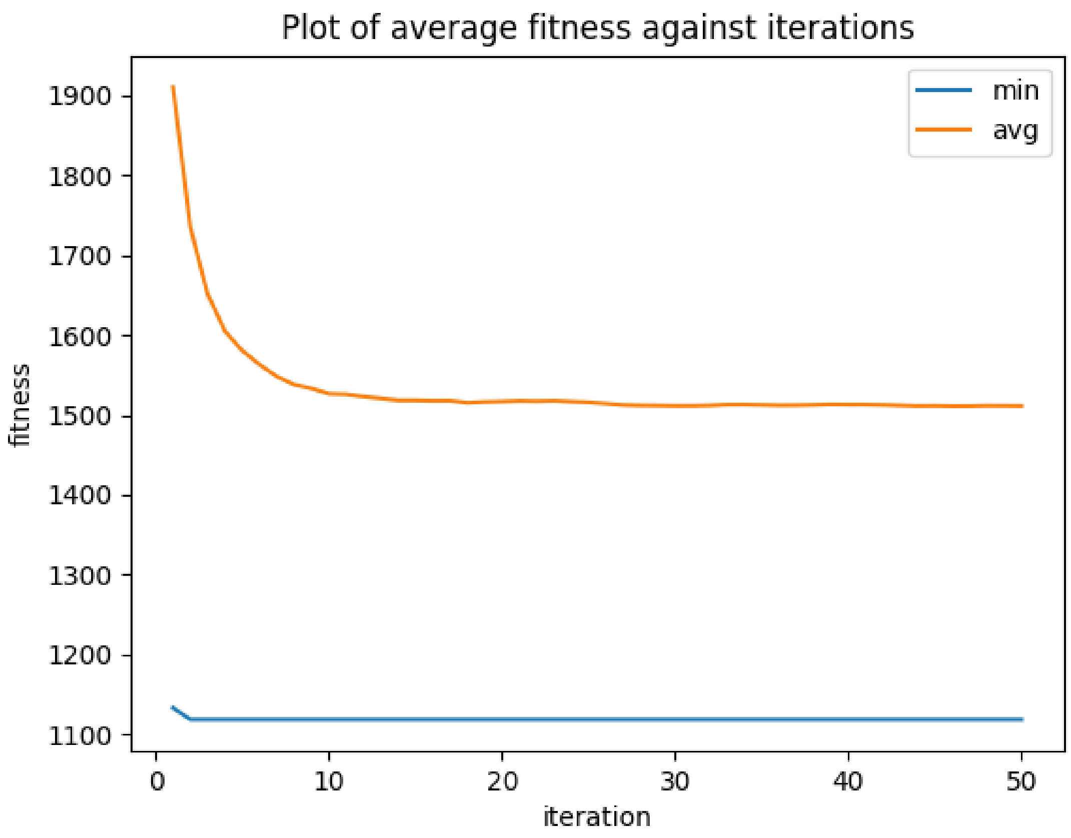
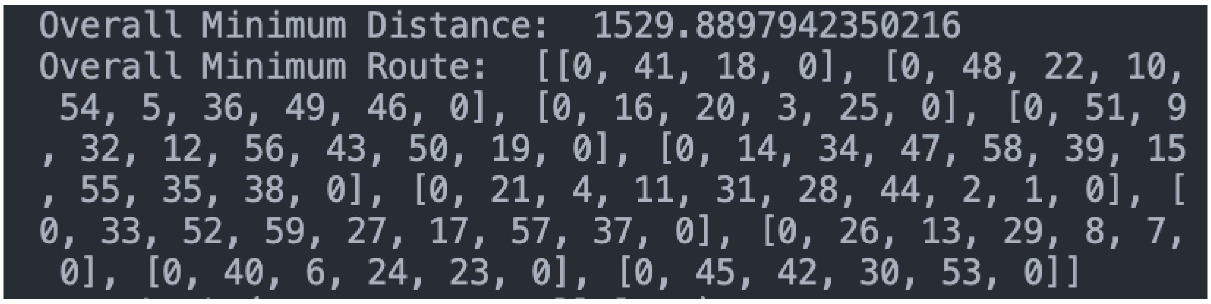


Figure 5: Minimum Distance Plot

1.3.3 A-n60-k9

The minimum distance that we could achieve for this dataset was 1529.890. The following is the route that we found:



```
Overall Minimum Distance:  1529.8897942350216
Overall Minimum Route:  [[0, 41, 18, 0], [0, 48, 22, 10,
 54, 5, 36, 49, 46, 0], [0, 16, 20, 3, 25, 0], [0, 51, 9
, 32, 12, 56, 43, 50, 19, 0], [0, 14, 34, 47, 58, 39, 15
, 55, 35, 38, 0], [0, 21, 4, 11, 31, 28, 44, 2, 1, 0], [
0, 33, 52, 59, 27, 17, 57, 37, 0], [0, 26, 13, 29, 8, 7,
 0], [0, 40, 6, 24, 23, 0], [0, 45, 42, 30, 53, 0]]
```

Figure 6: Minimum Distance Route

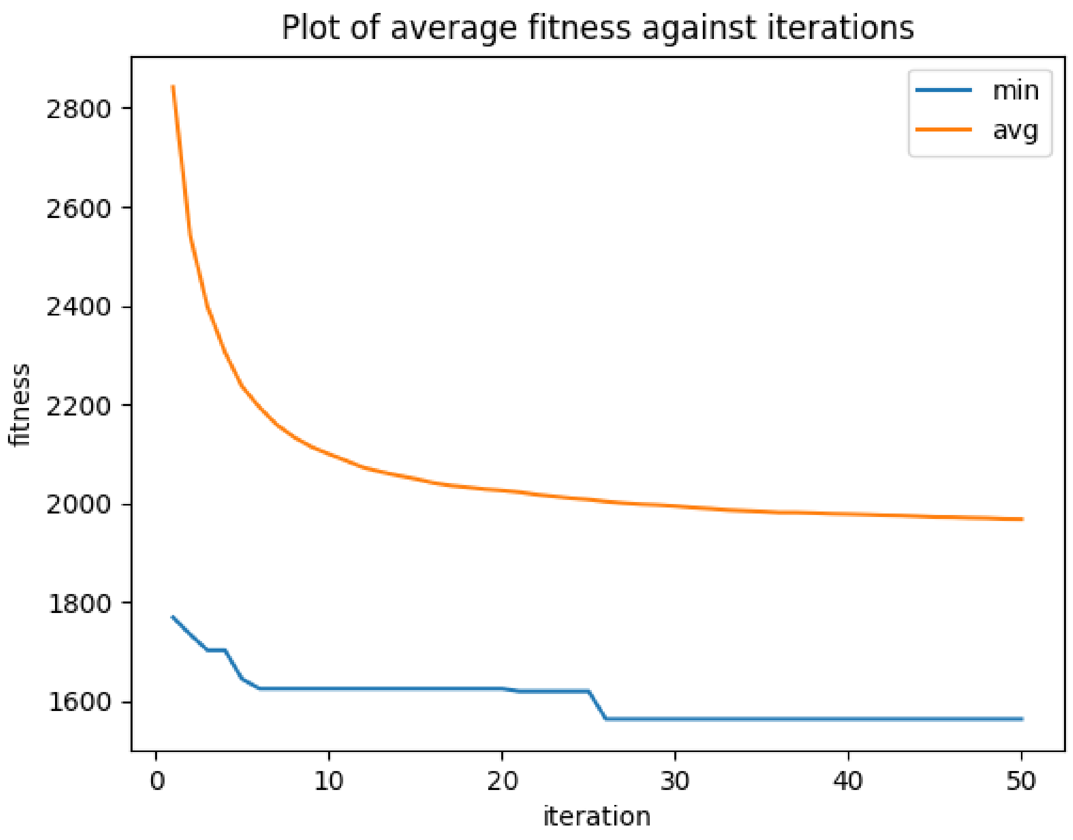
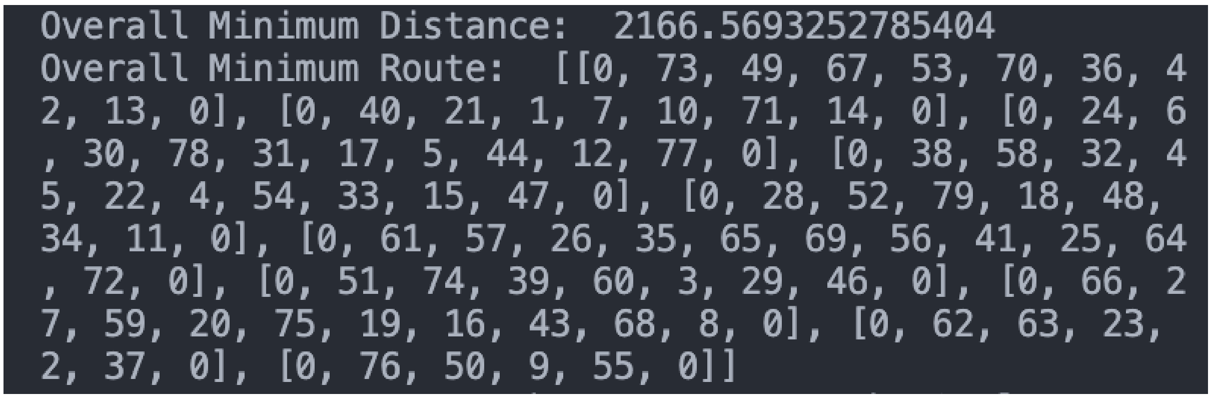


Figure 7: Minimum Distance Plot

1.3.4 A-n80-k10

The minimum distance that we could achieve for this dataset was 2166.569. The following is the route that we found:



```
Overall Minimum Distance:  2166.5693252785404
Overall Minimum Route:  [[0, 73, 49, 67, 53, 70, 36, 4
2, 13, 0], [0, 40, 21, 1, 7, 10, 71, 14, 0], [0, 24, 6
, 30, 78, 31, 17, 5, 44, 12, 77, 0], [0, 38, 58, 32, 4
5, 22, 4, 54, 33, 15, 47, 0], [0, 28, 52, 79, 18, 48,
34, 11, 0], [0, 61, 57, 26, 35, 65, 69, 56, 41, 25, 64
, 72, 0], [0, 51, 74, 39, 60, 3, 29, 46, 0], [0, 66, 2
7, 59, 20, 75, 19, 16, 43, 68, 8, 0], [0, 62, 63, 23,
2, 37, 0], [0, 76, 50, 9, 55, 0]]
```

Figure 8: Minimum Distance Route

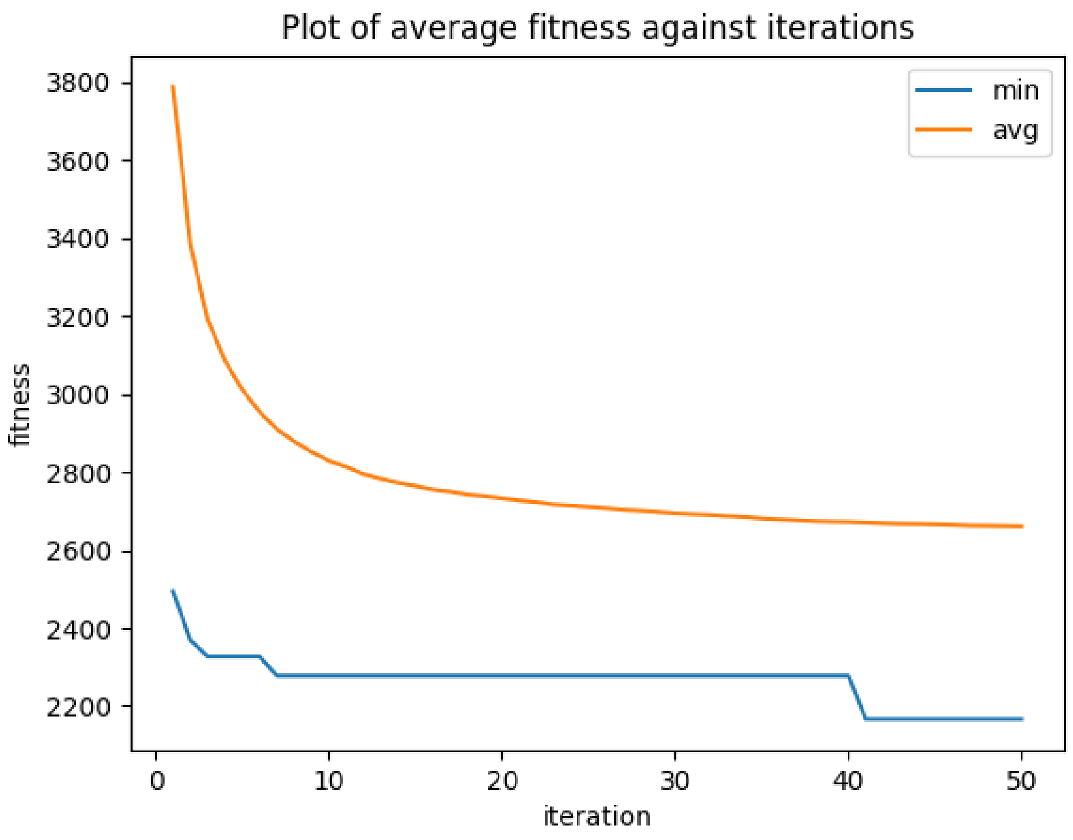
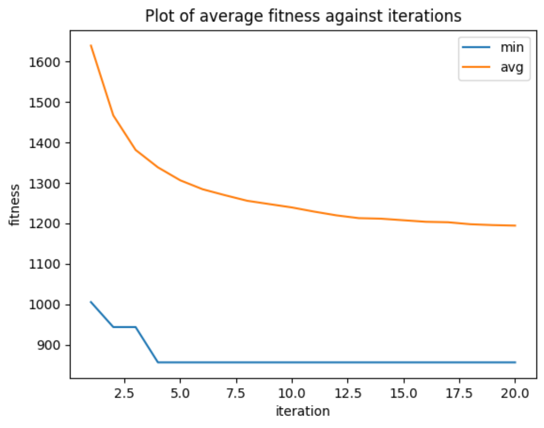


Figure 9: Minimum Distance Plot

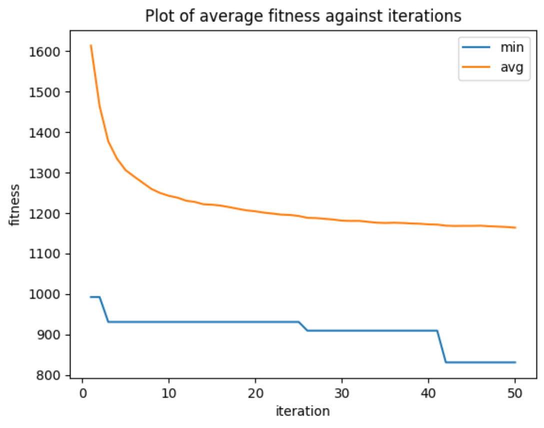
1.4 Plot

Plots of the best and average solution of Capacitated Vehicle Routing Problem using Ant Colony Optimization. The best solution is the one with the least distance. The average is calculated of the distance of all the solutions in the population in each iteration.

Dataset A-n32-k5



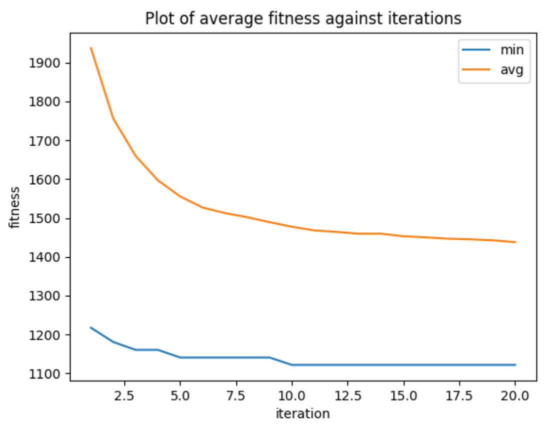
(a) ACO with 20 iterations



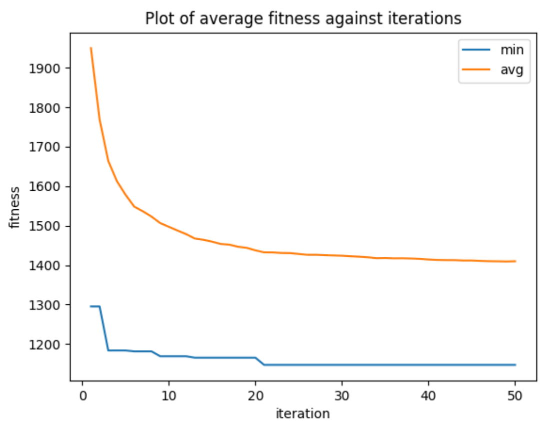
(b) ACO with 50 iterations

Figure 10: Note: Fitness is the distance

Dataset A-n44-k6



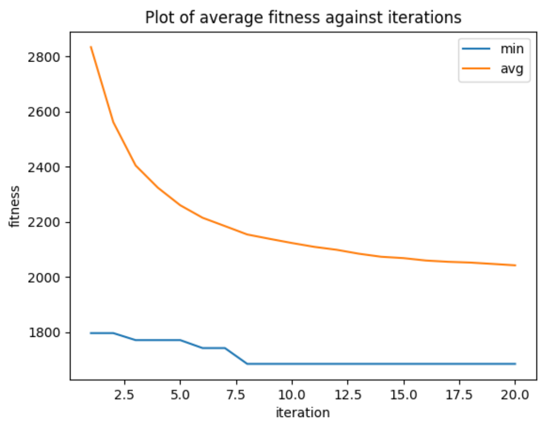
(a) ACO with 20 iterations



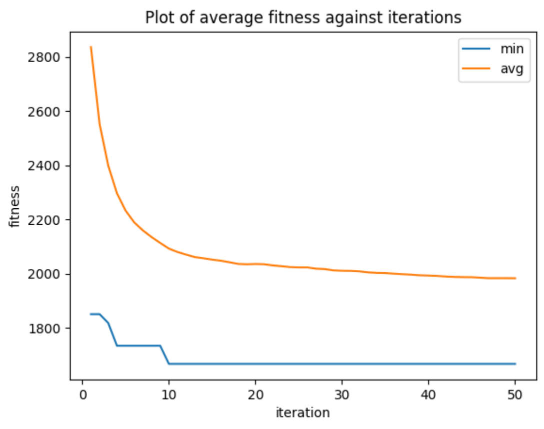
(b) ACO with 50 iterations

Figure 11: Note: Fitness is the distance

Dataset A-n60-k9



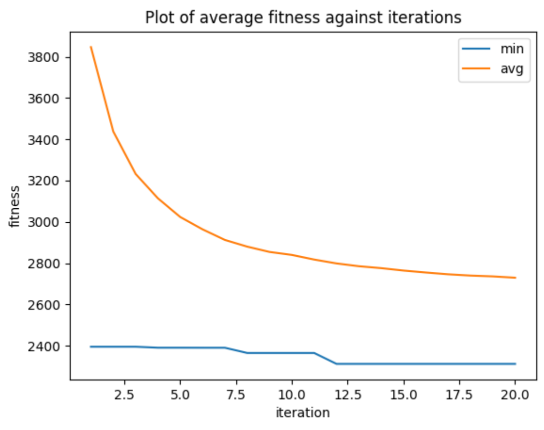
(a) ACO with 20 iterations



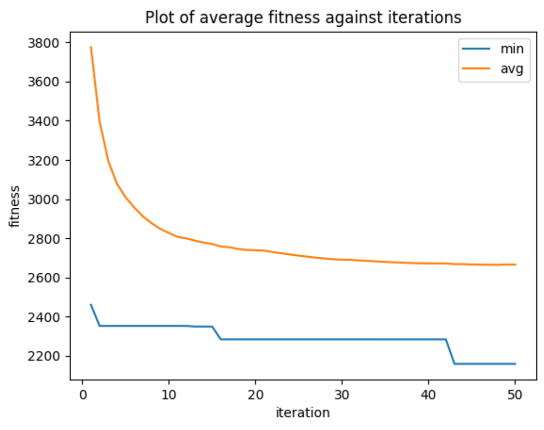
(b) ACO with 50 iterations

Figure 12: Note: Fitness is the distance

Dataset A-n80-k10



(a) ACO with 20 iterations



(b) ACO with 50 iterations

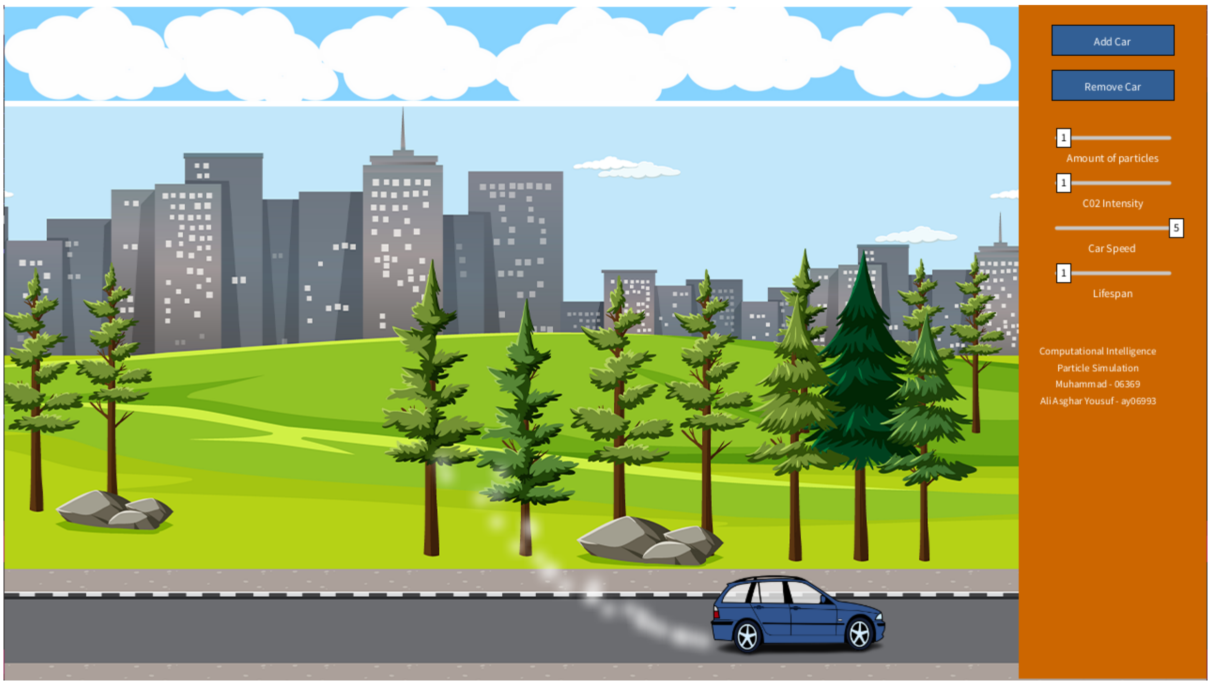
Figure 13: Note: Fitness is the distance

2 Visualizing Swarms

Visualizing Smoke Particle from Car Exhaust System

Given the current climate change events, it is important to create awareness of our daily carbon footprint, so we have made this smoke simulator that will aware people of how a car can damage our atmosphere, while at the same time we can represent particle system with it.

Particle System is a collection of many minute particles that together represent a fuzzy object. Here we have a smoke particles that together will represent smoke from car exhaust system.



1
Amount of particles
1
C02 Intensity
5
Car Speed
1
Lifespan

Computational Intelligence
Particle Simulation
Muhammad - 06369
Ali Asghar Yousuf - ay06993

Figure 14: Car Exhaust Smoke Simulator

First we will classify all the attributes of a particle system

- Position: Each particle has a given position on the screen with 'x' and 'y' coordinate.
- Velocity: Each particle has a randomly generated velocity which is constant.
- Lifetime: Life of a particle can be changed in this system. When a life of particle ends, it dies from the system.
- Color: Initially it is white, if the CO2 intensity increases, it becomes black.
- Shape: The particle has round shape.
- Size: 16 pixels.

Each of the following phases of particles has been implemented in this system

- Particle Lifespan
  - Each particle is generated randomly in each iteration
  - Particles moves upwards towards the sky.
  - Particle is rendered on the screen on each given position

  - Particle then reaches extinction phase when it dies from the system

- Particle Generation
  - Particle is generated in each iteration of the system randomly. We can control particle generation by increasing or decreasing its chances of generation in the given system.

- Particle Dynamics
  - The particle has a position vector and it is moved by adding the corresponding velocity vector to it.
  - The particle has a zero acceleration and its velocity is kept constant. However acceleration can be added to velocity.

- Extinction

The video of the simulation can be seen here: https://www.youtube.com/watch?v=BU9WsN1110Y