Intro to AI - homework 1

Vehicle Routing Problem (VRP)

Task 2.2 - The Impact of Inertia Weight in PSO

In PSO, inertia weight (w) plays a big role in how the algorithm finds solutions. Think of w like a balance scale between exploring new places (exploration) and making the best of what we already know (exploitation). When w is big, particles in our algorithm go far and wide, looking for new solutions. This is like exploring unknown parts of a map in search of treasure. When w is small, particles stick close to the best solutions they've found, trying to make those solutions even better. This is like digging deeper where we've already found some treasure, hoping to find more.

almost correct.

Task 2.3 - A Second Way to Update Inertia Weight

Another way to adjust the inertia weight (w) could be to change it based on how well the particles are doing. If exploring new areas is working well, we could increase w to encourage even more exploration. If it's not bringing better results, we could decrease w to focus more on improving the solutions we've already found. This method is different from the first because it reacts to how successful the algorithm is at finding better solutions, rather than just following a set plan from the start.

good idea. missing explantion of how to implemet it

Task 3.5 - Answer Questions

[Question 1] Role of Elements in PSO for Solving VRP

In PSO, solving the VRP (Vehicle Routing Problem), particles represent different ways to plan the delivery routes. Each particle has a position that shows one possible set of routes, and its velocity determines how fast the particle's position changes, exploring new routes. The fitness evaluation tells us how good a particular set of routes is, considering factors like distance or time.

Using these elements, PSO searches for the optimal or near-optimal solution by letting particles explore different route plans (positions) and gradually guiding them towards the best solutions found (based on fitness). The particles adjust their routes (positions) based on their own best discoveries (personal best) and the best routes found by any particle (global best), aiming to find the most efficient delivery plan.

[Question 2] Suggesting a Particle Representation for VRP

this is an idea for one route. How would it be represented for multiple.

A particle could be represented as a sequence of numbers, where each number represents a delivery location, and the sequence order dictates the delivery order. The objective function would evaluate how efficient a delivery route is by calculating the total distance travelled and perhaps considering traffic conditions or delivery windows. The goal is to minimize the total distance while adhering to any additional constraints.

[Question 3] Local and Global Best Solutions in PSO

In PSO, the local best (personal best) solution is the best solution found by a single particle so far. The global best solution is the best one found by any particle in the swarm. During optimization, each particle updates its route based on both its best discovery and the overall best route found by the swarm. This helps the particles refine their routes (exploit) while still considering new possibilities (explore), aiming to find the most efficient overall delivery plan.

Task 4.2 - Influence of Different Parameters

In PSO, the parameters we adjust can significantly affect both the quality of the solutions we find and how long it takes the algorithm to run. We experimented with three main parameter sets for each dataset to see how these changes impact performance.

Parameter Set 1:

- **SWARM_SIZE = 50, MAX_ROUTES = 5, C1 = C2 = 2.0, W_MIN = 0.4, W_MAX = 0.9, MAX_ITER = 100**
- This setup aimed for a balanced approach, with a medium swarm size and moderate levels of exploration and exploitation. It provided a good starting point, but there's room for improvement in both solution quality and efficiency.

Parameter Set 2:

- **SWARM_SIZE = 30, MAX_ROUTES = 4, C1 = C2 = 1.5, W_MIN = 0.5, W_MAX = 0.8, MAX_ITER = 150**

- With a smaller swarm and fewer routes, this setup focused more on optimization than exploration, aiming to refine solutions more precisely. It showed that reducing the swarm size didn't necessarily worsen the solution quality, and longer iterations allowed for more detailed exploration within a constrained solution space.

Parameter Set 3:

- **SWARM_SIZE = 70, MAX_ROUTES = 6, C1 = C2 = 2.5, W_MIN = 0.3, W_MAX = 0.7, MAX_ITER = 200**
- This setup was the most aggressive in terms of exploration, with a larger swarm size and higher C1 and C2 values for stronger personal and global best influences. It was designed to explore the solution space more broadly and was particularly useful in more complex scenarios (like Ex2-d22 and Ex3-d33), where the search space was larger.

Observations:

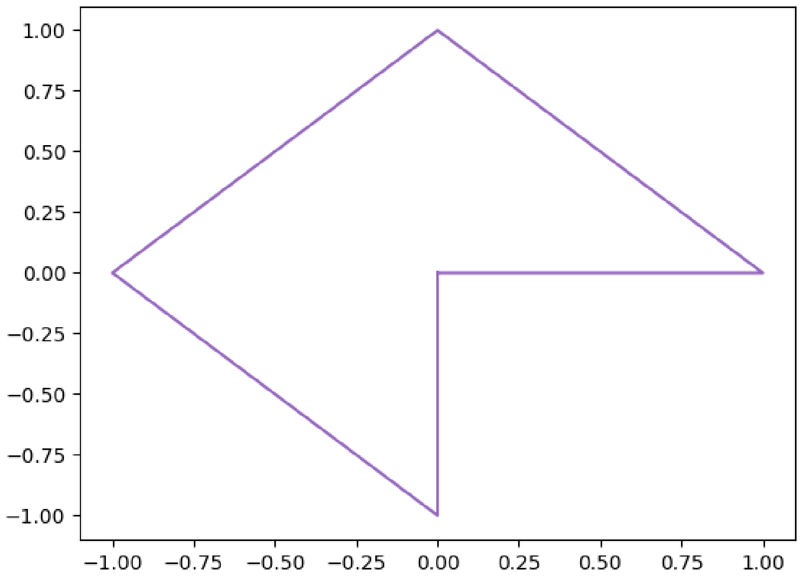
- **Swarm Size:** Larger swarms tended to explore the solution space more thoroughly but increased computational load and run-time. Smaller swarms were quicker but risked missing out on potentially better solutions.
- **MAX_ROUTES:** Increasing the maximum number of routes allowed the algorithm to consider more complex solutions but also increased the complexity of the optimization problem.
- **C1 and C2 Values:** Higher values encouraged particles to follow their own best and the global best more aggressively, potentially leading to faster convergence but risking premature convergence to suboptimal solutions.
- **W_MIN and W_MAX:** A wider range allowed for greater variation in exploration and exploitation behaviors over the course of the algorithm's run. Starting with a higher W_MAX promoted initial exploration, while decreasing towards W_MIN focused on exploitation.
- **MAX_ITER:** More iterations gave the swarm more opportunities to improve upon their solutions, but also increased run-time. The trade-off between

solution quality and computational time became evident, especially in complex scenarios.

Each parameter set influenced the balance between exploration and exploitation, affecting both the quality of the solution and the time required to reach it. Experimentation showed that there's no one-size-fits-all configuration; the best parameter set depends on the specific characteristics and requirements of the problem being solved.

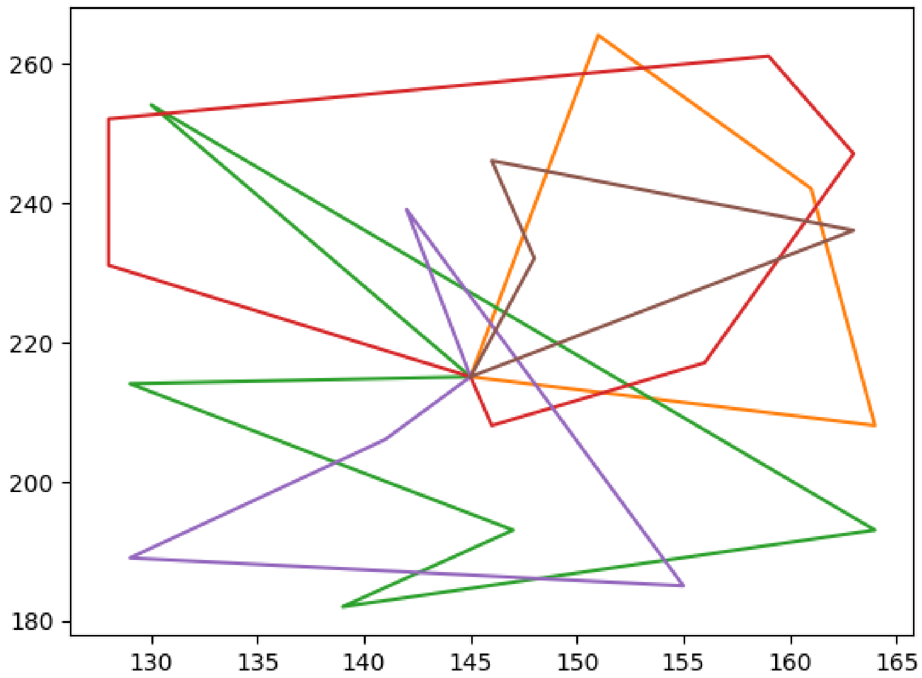
Task 4.3 - Best Results for Each Dataset

For **Ex1-d5**, the best results came from using a swarm size of 50, a maximum of 5 routes, C1 and C2 set to 2.0, and inertia weight ranging from 0.4 to 0.9 over 100 iterations. The total distance was 6.24 with a total time of 36.23.



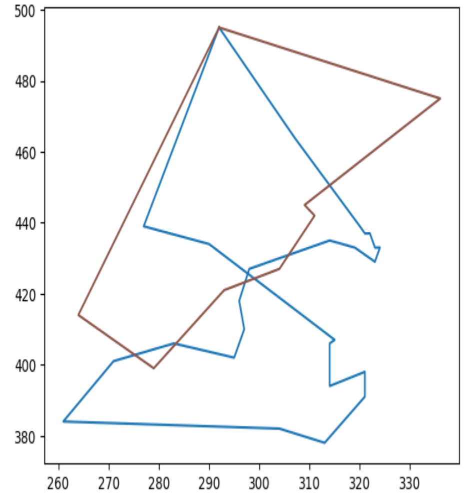

```
vehicle 1 route: depot ->0 -> 0 -> depot
Distance for vehicle 1 r_distance=0.0, Time traveled = 0.0
vehicle 2 route: depot ->0 -> 0 -> depot
Distance for vehicle 2 r_distance=0.0, Time traveled = 0.0
vehicle 3 route: depot ->0 -> 0 -> depot
Distance for vehicle 3 r_distance=0.0, Time traveled = 0.0
vehicle 4 route: depot ->0 -> 0 -> depot
Distance for vehicle 4 r_distance=0.0, Time traveled = 0.0
vehicle 5 route: depot ->0 -> 1 -> 2 -> 3 -> 4 -> 0 -> depot
Distance for vehicle 5 r_distance=6.242640687119285, Time traveled = 36.23
Total Distance is: 6.242640687119285, Total Time = 36.23
```

For **Ex2-d22**, using a swarm size of 70, a maximum of 6 routes, C1 and C2 at 2.5, with inertia weight from 0.3 to 0.7 over 200 iterations, resulted in a total distance of 682.13 and a total time of 134.61.




```
vehicle 1 route: depot ->0 -> 0 -> depot
Distance for vehicle 1 r_distance=0.0, Time traveled = 0.0
vehicle 2 route: depot ->0 -> 15 -> 7 -> 1 -> 0 -> depot
Distance for vehicle 2 r_distance=127.91262519904768, Time traveled = 21.46
vehicle 3 route: depot ->0 -> 13 -> 17 -> 21 -> 18 -> 3 -> 0 -> depot
Distance for vehicle 3 r_distance=196.22500986921634, Time traveled = 28.140000000000004
vehicle 4 route: depot ->0 -> 14 -> 12 -> 5 -> 2 -> 4 -> 11 -> 0 -> depot
Distance for vehicle 4 r_distance=142.5160150819944, Time traveled = 36.93
vehicle 5 route: depot ->0 -> 16 -> 19 -> 20 -> 8 -> 0 -> depot
Distance for vehicle 5 r_distance=136.69295227730365, Time traveled = 19.28
vehicle 6 route: depot ->0 -> 10 -> 6 -> 9 -> 0 -> depot
Distance for vehicle 6 r_distance=78.7865284205577, Time traveled = 28.8
Total Distance is: 682.1331308481198, Total Time = 134.61
```

For **Ex3-d33**, the parameters of a swarm size of 70, 6 maximum routes, C1 and C2 at 2.5, and inertia weight from 0.3 to 0.7 over 200 iterations achieved a total distance of 634.64 and a total time of 175.97.




```
vehicle 1 route: depot ->0 -> 3 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10 -> 32 -> 11 -> 1 -> 15 -> 17 -> 25 -> 26 -> 28 -> 16 -> 24 -> 23 -> 20 -> 21 -> 22 -> 19 -> 18 -> 31 -> 30 -> 0 -> depot
Distance for vehicle 1 r_distance=380.264818009511, Time traveled = 149.16000000000003
vehicle 2 route: depot ->0 -> 0 -> depot
Distance for vehicle 2 r_distance=0.0, Time traveled = 0.0
vehicle 3 route: depot ->0 -> 0 -> depot
Distance for vehicle 3 r_distance=0.0, Time traveled = 0.0
vehicle 4 route: depot ->0 -> 0 -> depot
Distance for vehicle 4 r_distance=0.0, Time traveled = 0.0
vehicle 5 route: depot ->0 -> 0 -> depot
Distance for vehicle 5 r_distance=0.0, Time traveled = 0.0
vehicle 6 route: depot ->0 -> 4 -> 2 -> 12 -> 13 -> 14 -> 27 -> 29 -> 0 -> depot
Distance for vehicle 6 r_distance=254.37450525512224, Time traveled = 26.810000000000002
Total Distance is: 634.6393232646333, Total Time = 175.97000000000003
```