Solving the Traveling Salesman Problem Using Genetic Algorithm and Particle Swarm Optimization

Erla Hoxha
Business Informatics, Epoka University
erlahoxha04@gmail.com

February 12, 2025

1. INTRODUCTION

The traveling salesman problem[1,7] or TSP is one of the most difficult problems in computer science. The TSP can be imagined as a salesman that is trying to visit a set of cities on a massive road trip and its goal is to find the shortest route possible starting from one city and then visiting all cities only once and returning back to the original destination.
The TSP can be modeled as a graph theory problem where every city is a vertex or node of a graph and the graph is assumed to be complete meaning there exists a direct edge between every pair of vertices given any two cities so we assume that the distance from city a to city b is the same as the distance from b to a.

One of the first steps in any complex problem is to try the most direct method to finding a solution for the TSP. So we are looking for a tour of the graph which is defined as a path that starts in a node, visits all the nodes in the graph and then returns back to the starting node . The brute force[2] solution involves finding every single possible tour of the cities and then picking the tour that has the minimum total distance traveled. This works for small instances of the problem but as soon as we increase the number of cities to just 20 cities it already becomes completely unreasonable for a computer to even try. One of the reasons the TSP is such a challenging problem is that it belongs to a class of problems called NP hard problems. All NP Hard problems are at least NP Complete problems. NP Complete problems are problems for which the solution can not be solved in a reasonable amount of time (Polynomial time) and they also can not be verified in polynomial time[6].Due to its NP-hard nature, the TSP has direct implications in several areas such as logistics, route planning, and bioinformatics [3]. Because there is no scalable algorithm that will allow us to solve this in a reasonable time for a large number of points, the only thing we can do is come up with an estimate for the solution. There are various techniques we can use for this, for example, simulated annealing, ant colony optimization, but in this paper I will analyze the Genetic Algorithm and the Particle Swarm Optimisation.

2. GENETIC ALGORITHM

Genetic Algorithms involve a heuristic process based on the concept of natural selection[3].The genetic algorithm can give us a good solution that is near to optimal, but it doesn't guarantee us that it will give the optimal solution. A genetic algorithm works in two steps. First, we create an initial population that will contain an initial number of possible solutions. Then, we evolve this initial population through a number of generations, and each time, we will get closer to the final solution.

So in theory a genetic algorithm is trying to emulate the genetic evaluation process to find the solution. It uses chromosomes, crossover , mutation and survival of the fittest to try to find the good solution. The biggest advantage is that we don't really need to know how to solve for the optimal solution, we just need to know how to evaluate the quality of the solution and through an iterative process we will get a good solution.

HOW GENETIC ALGORITHM WORKS

So we start with randomly generated k chromosomes and this is called population (Fig. 1). Each of these chromosomes actually presents an individual solution. With this (k) initial chromosome through crossover and mutation we create another k-offspring[7].

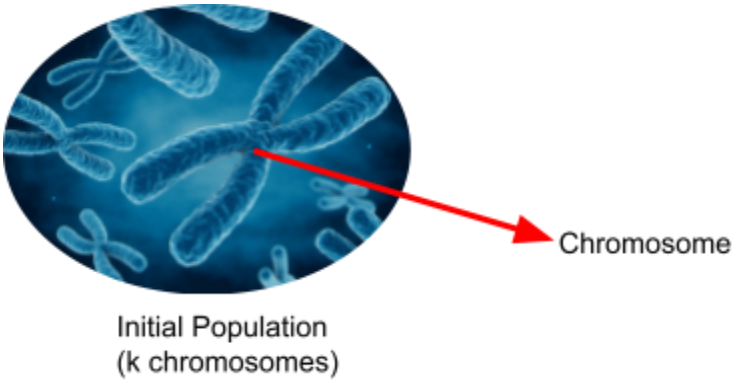
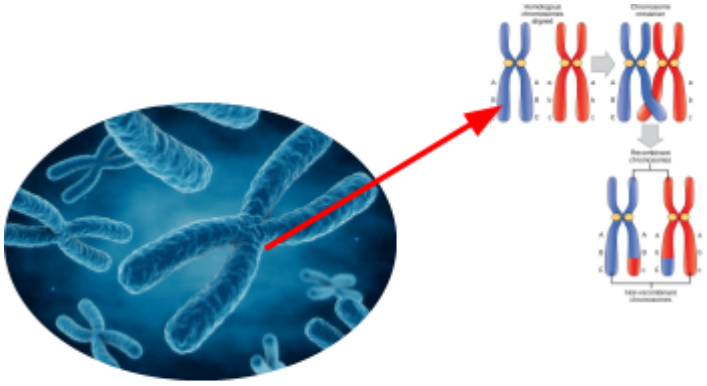


Figure 1. Initial Population

CROSSOVER PROCESS

If we take it from the biological part we can say that we take two chromosomes and we swap their parts. As seen in figure 2 we have two chromosomes blue and red and in the third phase ; we see that some portion of blue has gone into the red and some portion of red has gone into the blue so these two chromosomes have swapped parts.



OBJ K- initial chromosomes

Figure 2. Crossover Process

MUTATION PROCESS

To better explain mutation I have used the figure below (Fig. 3).So the first chromosome is the original one and next to it is the mutated one. Two parts of the original one have switched places , so the mutated chromosome so the orange has gone up and the blue has gone down. In the mutation process different from the crossover only one chromosome is needed for the process.

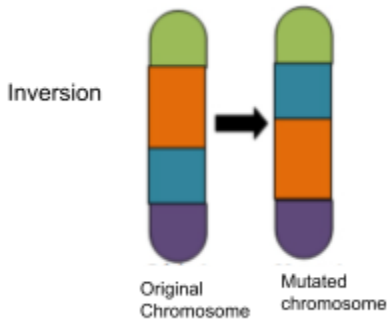


Figure 3. Mutation Process

These two are the methods on how the k-offspring chromosomes are created.This k-offspring has the crossover components which are created from the parents (crossover process) or that are created from the mutation process (Fig. 4).

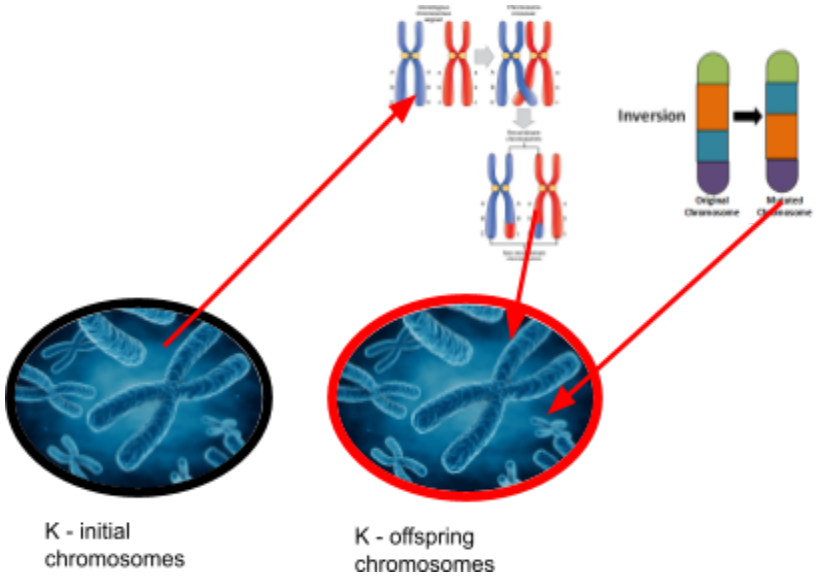


Figure 4. K-offspring chromosomes

So among any stage, (k) population and k-offspring chromosomes, a major fitness function is used to detect the best (k) chromosomes and those will become the next generation.This process of evolution (crossover ect.) will be repeated until the iteration threshold is reached.

3. TRAVELING SALESMAN PROBLEM

Formally, the TSP can be represented by a complete graph G=(V,E), where V is the set of vertices (cities) and E is the set of edges with associated weights (distances or costs between cities).[3]

For our TSP it will look like this :

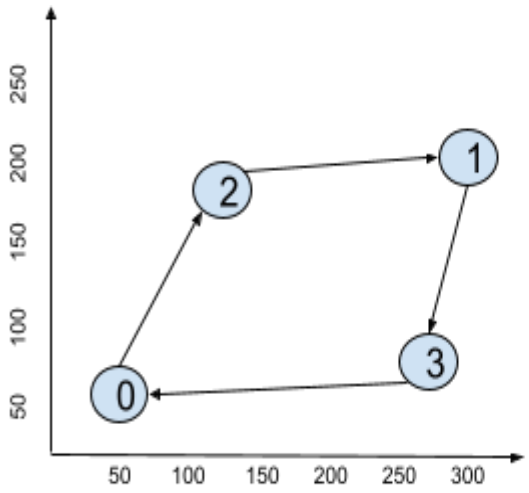


Figure 5. TSP graph where the cities are represented by a node from 0 to 3 and each city has its own coordinate.

4. LOCAL MINIMA

One of the biggest challenges when solving the Traveling Salesman Problem using optimization algorithms is avoiding local minima[4]. A local minimum is a solution that appears to be the best within a small range but is not the overall best solution. Both Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) can get stuck in these suboptimal solutions, preventing them from finding the shortest possible route.

Local Minima in Genetic Algorithm

In GA, local minima occur when the population converges too early, meaning the solutions become too similar and stop improving. This often happens when the mutation rate is too low or when selection favors the best solutions too strongly, reducing diversity in the population. In order to fix it we can increase the mutation rate so we can introduce more randomness in the population.

Local Minima in Particle Swarm Optimization

In PSO, particles follow the global best solution, which can lead them into a local minimum where they stop exploring better paths. This happens when the balance between personal best and global best is not properly set. In order to fix it we can adjust the best attractor value by lowering the influence of the personal best allowing particles to explore new possibilities rather than getting stuck.

5. GENETIC ALGORITHM WITH REGARDS TO TRAVELING SALESMAN PROBLEM

In regards with the traveling salesman problem the genetic algorithm is implemented following these steps [7,10]:

1. Finding the Population

So as I explained in section 1 the salesman has to start from one city and has to go to all other cities just once and then come back to the original city.
In figure 5 I have demonstrated a simple example where the salesman has to go through 4 cities 0, 1, 2 and 3. Based on this I created a solution : 0-2-1-3-0. This solution that I created is an example of a chromosome. This chromosome is part of the population so the population is made of the possible solutions where each solution represents a sequence of cities in a specific order.

2. Calculate Fitness

In order to evaluate how good each of the solutions are we use the fitness function. The fitness function is based on the total route distance. The shorter the route, the better the fitness.

3. Selection

We select the best solutions (highest fitness scores) to be parents for the next generation.This is an emulation of the concept of "the strong prevail" from natural selection, where those with beneficial features have a greater likelihood of passing down their genes to the next generation[3]. There are different methods for selection, but in my project I have used the fitness-proportionate selection.

4. Crossover Process

We create new solutions (children) by combining parts of two parent solutions(Fig. 6).

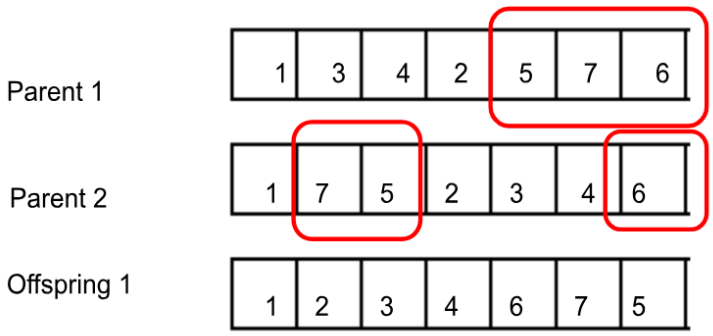


Figure 6. Crossover Process in TSP.

5. Mutation Process

To introduce randomness and avoid getting stuck in local optima, we randomly swap some cities in the child(Fig. 7).

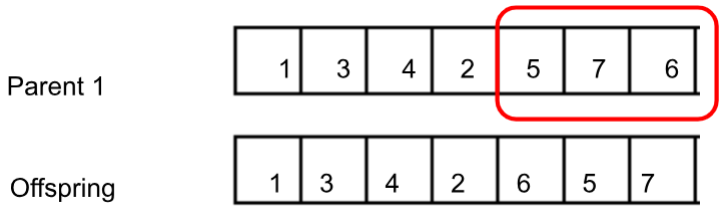


Figure 7. Mutation Process in TSP.

6. Replace the Population

The newly generated solutions replace the worst-performing solutions from the previous generation.

By repeatedly selecting, crossing over, and mutating solutions, we gradually evolve towards better solutions. Mutation ensures we explore different possibilities rather than just refining the same set of solutions over and over.

6. PARTICLE SWARM OPTIMISATION

Particle Swarm Optimization (PSO) [11] is an algorithm inspired by the movement of birds in flocks, schools of fish, and swarm behavior. It was first introduced by Kennedy and Eberhart[5] as a method for optimizing nonlinear functions.PSO works by using a group of particles, where each particle represents a possible solution to the problem[3]. These particles move through the search space to find the best solution.

Each particle has a position and velocity, which are updated during the optimization process based on two factors[3]:

1. The best position the particle has ever reached (personal best)
2. The best position found by the entire swarm (global best)

PSO is widely used in continuous optimization problems because it is simple and efficient. Over time, researchers have adapted it for discrete problems like the Traveling Salesman Problem (TSP)[3].

KEY COMPONENTS OF PARTICLE SWARM OPTIMISATION FOR TSP

1. Particles (Solutions)

In Particle Swarm Optimization (PSO) applied to the Traveling Salesman Problem (TSP), the algorithm represents each possible route as a particle in a swarm. Each particle is a candidate solution that evolves over multiple iterations to find the shortest path.

2. Global Best Route

The global best route is the best solution found by any particle in the swarm so far. Every particle in the swarm is influenced by this solution and gradually moves towards it, helping the entire swarm converge toward an optimal or near-optimal route[3].

3. Personal Best Route

Each particle also maintains a personal best route, which is the best route that the individual particle has discovered through its own exploration. While the particle is influenced by the swarm's global best, it also retains some independence by attempting to refine its own best solution[3].

4. Velocity Updates

Unlike traditional PSO, where velocity is a mathematical vector, the velocity updates in TSP are handled through swap operations. Instead of moving in continuous space, particles adjust their positions by swapping cities within their routes. These swaps are based on both their personal best and the global best , allowing them to improve their routes step by step while still maintaining diversity within the swarm [3].

7. GENETIC ALGORITHM EXPERIMENTS AND THEIR RESPECTIVE RESULTS

1. First experiment (10 cities)

For the first experiments I have taken in consideration 10 cities and the coordinates are : (0, 0), (1, 3), (4,3), (6, 1), (3, 0), (5, 5), (2, 6), (8, 8), (6, 4), (7, 1) .

First experiment :

- ➔ Population was set to 50.
- ➔ Generations were set to 10.
- ➔ And the mutation rate was set to 0.01.

This is the initial experiment.

Population	Generations	Mutation	Best route found:	Best route distance:
50	10	0.01	[2, 1, 0, 9, 3, 4, 6, 5, 7, 8]	37.591507942287 48
50	10	0.01	[0, 1, 6, 5, 8, 2, 7, 3, 9, 4]	34.943454272709 05
50	10	0.01	[5, 2, 7, 8, 9, 3, 4, 0, 1, 6]	32.922716470774 11

The method I followed is as follows. First I ran the first experiment and saved its data :

1. The best route found,

2. And the best route distance. The first row contains the data after the first iteration. Then I tried to find the shortest path for the corresponding experiment by keeping track of the results. So, whenever I found a path that was shorter than the previous path, I noted it in the table. I repeated this several times until I found the shortest path from the iterations I had done.In this case the shortest distance is : 32.92271647077411. And it has this graph (Fig. 8) :

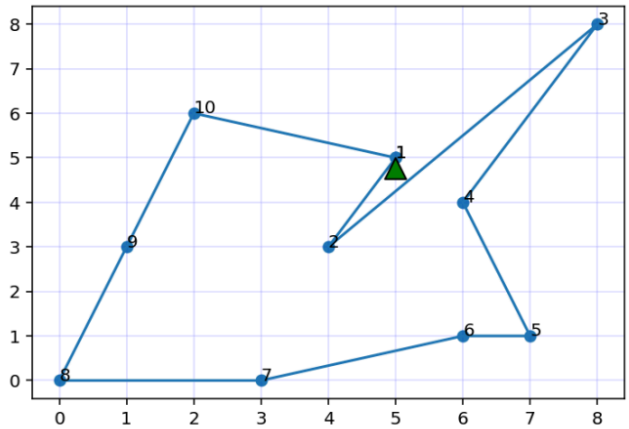


Figure 8. The graph of the shortest path of GA with population=50, generations = 10 and mutation = 0.01.

Second experiment :

- ➔ Population was set to 100.
- ➔ Generations were set to 10.
- ➔ And the mutation rate was set to 0.01.

So I have increased the population while keeping the generations and mutation rate the same.

Population	Generation	Mutation	Best route found:	Best route distance:
100	10	0.01	[9, 3, 8, 7, 5, 2, 6, 0, 1, 4]	35.771885776669 43
100	10	0.01	[0, 2, 1, 6, 7, 5, 8, 9, 3, 4]	33.468242550334 274
100	10	0.01	[9, 3, 4, 0, 1, 2, 6, 5, 7, 8]	31.969438558256 368

For the second experiment I found out that the shortest path based on the parameters is : 31.969438558256368 .

In the figure (Fig. 9) below I have modified the code to make 100 iterations for these specific parameters of this experiment. As seen, increasing the population provides more genetic diversity, leading to better solutions.The histogram has a shift to the left (towards shorter distances), with more solutions clustering around the optimal range.

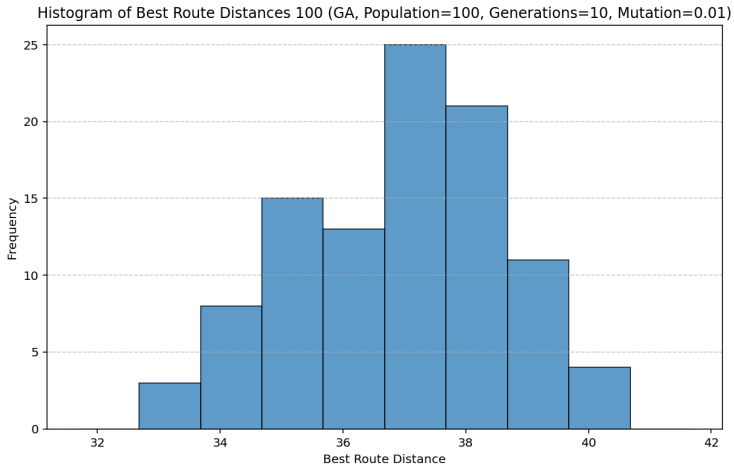


Figure 9. Histogram of the best route distance of GA with population = 100, generations=10 and mutation = 0.01.

Third experiment :

- ➔ Population was set to 500.
- ➔ Generations were set to 10.
- ➔ And the mutation rate was set to 0.01.

So I have increased the population while keeping the generations and mutation rate the same.

Population	Generation	Mutation	Best route found:	Best route distance:
500	10	0.01	[8, 2, 4, 9, 3, 0, 1, 6, 7, 5]	34.91017868374994
500	10	0.01	[0, 4, 2, 3, 9, 5, 7, 8, 6, 1]	33.97430865736935
500	10	0.01	[7, 8, 5, 6, 1, 0, 4, 3, 9, 2]	32.54413567094303
500	10	0.01	[8, 5, 2, 3, 9, 4, 0, 1, 6, 7]	31.7230608859
500	10	0.01	[1, 0, 4, 2, 3, 9, 8, 5, 7, 6]	31.458947335248848
500	10	0.01	[8, 5, 7, 6, 1, 0, 4, 3, 9, 2]	31.30986180329806

For the third experiment I found out that the shortest path based on the parameters is : 31.30986180329806.

In the figure (Fig.10) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 30 is for the distance around 34 to 36 and while for the shortest distance of less than 32 it has a frequency of less than 10.

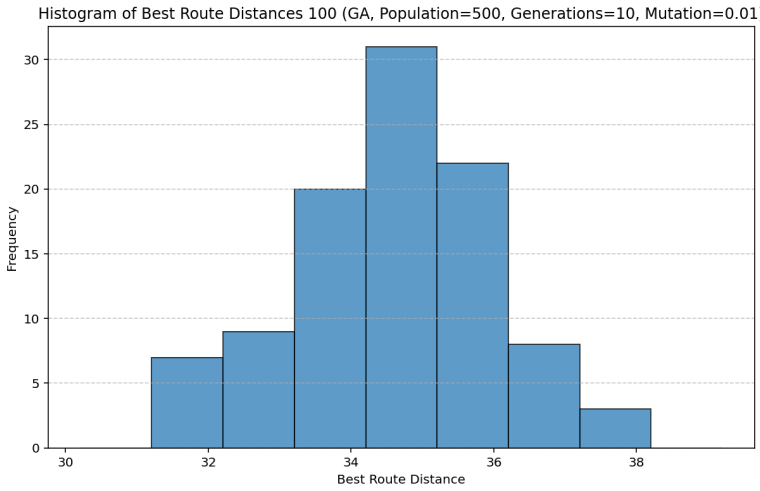


Figure 10. Histogram of the best route distance of GA with population = 500, generations=10 and mutation = 0.01.

Fourth experiment :

- ➔ Population was set to 50.
- ➔ Generations were set to 50.
- ➔ And the mutation rate was set to 0.01.

So I have increased the generations while keeping the population and mutation rate the same as in the initial experiment.

Population	Generations	Mutation	Best route found:	Best route distance:
50	50	0.01	[8, 5, 7, 2, 9, 4, 0, 1, 6, 3]	38.516314945776486
50	50	0.01	[3, 9, 7, 6, 0, 4, 1, 2, 5, 8]	36.97601126787586
50	50	0.01	[9, 3, 6, 7, 5, 8, 2, 1, 0, 4]	34.905985070547814
50	50	0.01	[1, 2, 0, 4, 9, 3, 8, 5, 7, 6]	34.26679285561518
50	50	0.01	[4, 0, 1, 6, 5, 7, 8, 3, 9, 2]	31.969438558256368

For the fourth experiment I found out that the shortest path based on the parameters is : 31.969438558256368.

In the figure (Fig. 11) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 20 is for the distance around 38 to 40 and while for the shortest distance of less than 32 it has a frequency of less than 2.5.

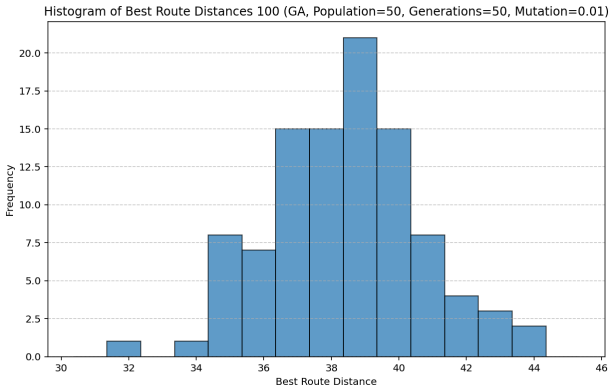


Figure 11. Histogram of the best route distance of GA with population = 50, generations=50 and mutation = 0.01.

Fifth experiment :

- ➔ Population was set to 50.
- ➔ Generations were set to 100.
- ➔ And the mutation rate was set to 0.01.

So I have increased the generations while keeping the population and mutation rate the same as in the initial experiment.

Population	Generation	Mutation	Best route found:	Best route distance:
50	100	0.01	[5, 7, 6, 8, 9, 3, 2, 1, 0, 4]	36.577479214673076
50	100	0.01	[3, 2, 9, 5, 8, 7, 6, 1, 0, 4]	35.60385217342433
50	100	0.01	[2, 8, 3, 9, 4, 0, 1, 6, 7, 5]	32.486992908410045

For the fifth experiment I found out that the shortest path based on the parameters is : 32.486992908410045.

In the figure (Fig. 12) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 20 is for the distance around 36 to 40 and while for the shortest distance of less than 32 it has a frequency of less than 2.5.

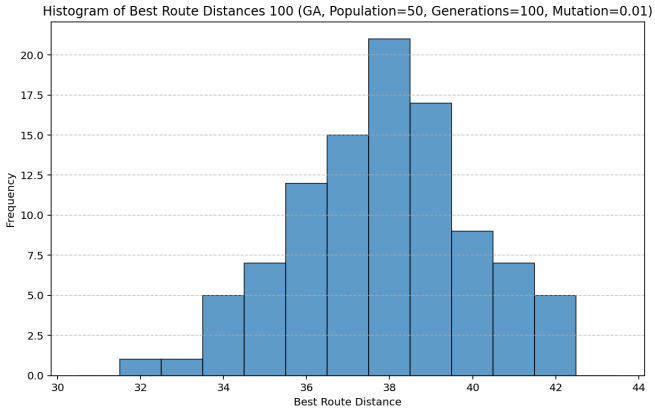


Figure 12. Histogram of the best route distance of GA with population = 50, generations=100 and mutation = 0.01.

Sixth experiment :

- ➔ Population was set to 50.
- ➔ Generations were set to 1000.
- ➔ And the mutation rate was set to 0.01.

So I have increased the generations while keeping the population and mutation rate the same as in the initial experiment.

Population	Generation	Mutation	Best route found:	Best route distance:
50	1000	0.01	[5, 8, 7, 6, 9, 3, 2, 4, 0, 1]	36.907091049657 44
50	1000	0.01	[3, 4, 0, 1, 2, 5, 7, 8, 6, 9]	35.818603706820 47
50	1000	0.01	[6, 5, 2, 8, 7, 3, 9, 4, 0, 1]	33.834320405402 48

For the sixth experiment I found out that the shortest path based on the parameters is : 33.83432040540248.

In the figure (Fig. 13) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 20 is for the distance around 36 to 38 and while for the shortest distance of less than 32 it has a frequency of less than 5.

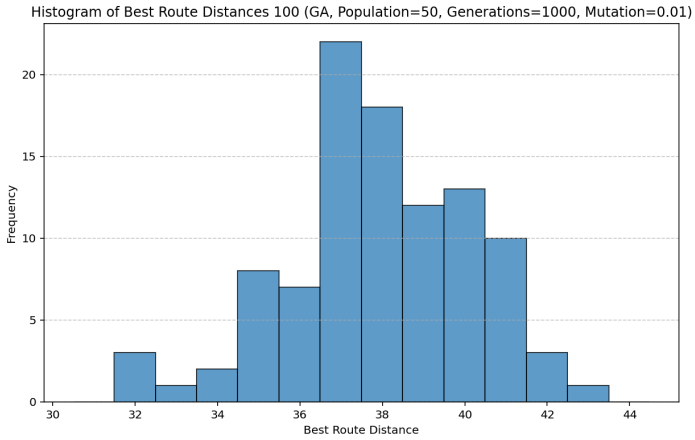


Figure 13. Histogram of the best route distance of GA with population = 50, generations=1000 and mutation = 0.01.

Seventh experiment :

- ➔ Population was set to 50.
- ➔ Generations were set to 10.
- ➔ And the mutation rate was set to 0.05.

So I have increased the mutation rate while keeping the generations and the population the same as in the initial experiment.

Population	Generation	Mutation	Best route found:	Best route distance:
50	10	0.05	[3, 5, 6, 7, 8, 0, 1, 2, 4, 9]	39.740838058004776
50	10	0.05	[0, 4, 2, 3, 9, 7, 5, 8, 1, 6]	37.30447934037034
50	10	0.05	[8, 5, 2, 7, 6, 1, 0, 4, 9, 3]	33.82562204359691

For the seventh experiment I found out that the shortest path based on the parameters is : 33.82562204359691.

In the figure (Fig. 14) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of 20 is for the distance around 38 to 40 and while for the shortest distance of 32 it has a frequency of less than 2.5.

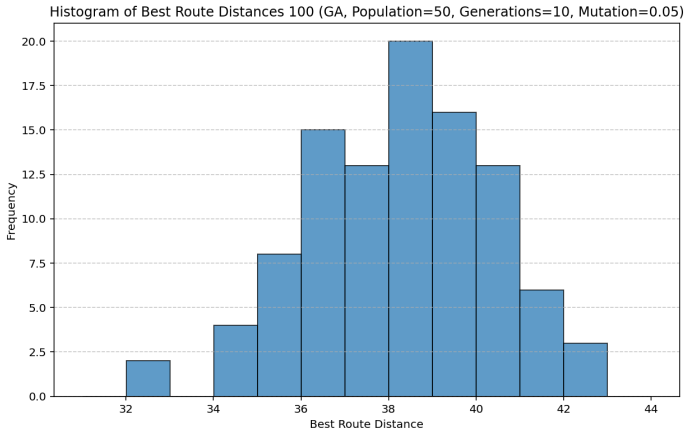


Figure 14. Histogram of the best route distance of GA with population = 50, generations=10 and mutation = 0.05.

Eighth experiment :

- ➔ Population was set to 50.
- ➔ Generations were set to 10.
- ➔ And the mutation rate was set to 0.09.

So I have increased the mutation rate while keeping the generations and the population the same as in the initial experiment.

Population	Generation	Mutation	Best route found:	Best route distance:
50	10	0.09	[8, 7, 1, 0, 4, 2, 9, 3, 6, 5]	37.98418327781727
50	10	0.09	[3, 0, 4, 1, 6, 7, 5, 2, 8, 9]	35.05220108855459
50	10	0.09	[8, 9, 3, 0, 4, 1, 6, 2, 5, 7]	34.56926502118161

For the eighth experiment I found out that the shortest path based on the parameters is : 33.82562204359691.

In the figure (Fig. 15) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 20 is for the distance around 38 to 40 and while for the shortest distance of less than 34 it has a frequency of less than 5.

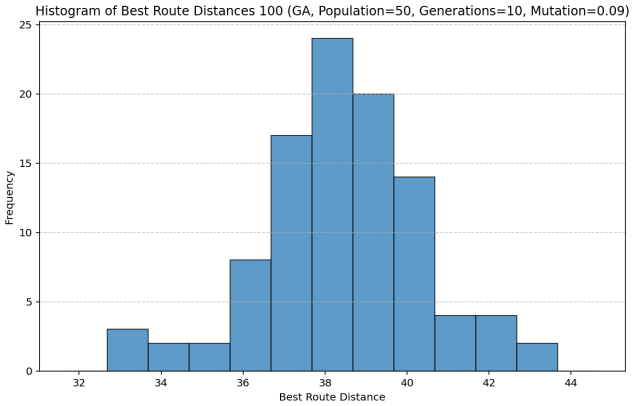


Figure 15. Histogram of the best route distance of GA with population = 50, generations=10 and mutation = 0.09.

Ninth experiment :

- ➔ Population was set to 500.
- ➔ Generations were set to 30.
- ➔ And the mutation rate was set to 0.01.

So I have increased the population and the generation and I haven't changed the mutation rate as in the initial experiment.

population	generation	mutation	Best route found:	Best route distance:
500	30	0.01	[5, 8, 2, 9, 3, 0, 4, 1, 6, 7]	34.673620288723505
500	30	0.01	[1, 0, 4, 2, 5, 8, 9, 3, 7, 6]	34.020752829968686
500	30	0.01	[2, 8, 5, 7, 6, 1, 0, 4, 9, 3]	31.20550653575616

For the ninth experiment I found out that the shortest path based on the parameters is : 31.20550653575616.

In the figure(16) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 20 is for the distance around 34 to 36 and while for the shortest distance of less than 32 it has a frequency of less than 10.

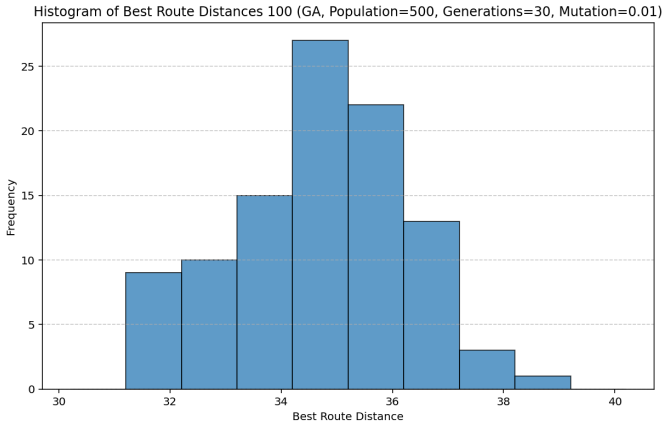


Figure 16. Histogram of the best route distance of GA with population = 500, generations=30 and mutation = 0.01.

Tenth experiment :

- ➔ Population was set to 300.
- ➔ Generations were set to 50.
- ➔ And the mutation rate was set to 0.05.

In this experiment I have changed all the parameters.

population	generation	mutation	Best route found:	Best route distance:
300	50	0.05	[5, 8, 2, 7, 6, 1, 0, 4, 3, 9]	34.336930033147 21
300	50	0.05	[7, 6, 1, 0, 4, 9, 3, 8, 2, 5]	32.486992908410 045
300	50	0.05	[1, 0, 4, 3, 9, 2, 8, 5, 7, 6]	31.309861803298 06

For the tenth experiment I found out that the shortest path based on the parameters is : 31.30986180329806.

In the figure (Fig. 17) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of 30 is for the distance around 36 to 38 and while for the shortest distance of less than 32 it has a frequency of less than 10.

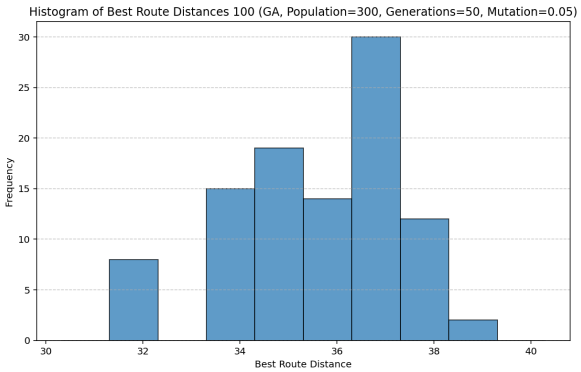


Figure 17. Histogram of the best route distance of GA with population = 300, generations=50 and mutation = 0.05.

2. Second Experiment (20 cities)

For the next experiments I added more cities by increasing the number of the coordinates so now the new coordinates are : (0, 0), (1, 3), (4, 3), (6, 1), (3, 0),(5, 5), (2, 6), (8, 8), (6, 4), (7, 1),(2, 3), (8, 7), (5, 1), (1,1), (3, 4),(7, 5), (2, 5), (6, 2), (4, 7), (4, 4).

First experiment :

- ➔ Population was set to 50.
- ➔ Generations were set to 10.
- ➔ And the mutation rate was set to 0.1.

This is the initial experiment.

population	generation	mutation	Best route found:	Best route distance:
50	10	0.1	[16, 3, 8, 19, 1, 0, 14, 15, 17, 12, 2, 9, 4, 6, 10, 13, 18, 5, 7, 11]	72.476029721822 93
50	10	0.1	[2, 16, 19, 17, 3, 18, 14, 6, 5, 8, 4, 0, 1, 10, 12, 9, 13, 15, 7, 11]	65.990377803759 67
50	10	0.1	[15, 8, 5, 7, 11, 17, 2, 16, 14, 19, 18, 10, 1, 13, 4, 12, 3, 0, 9, 6]	62.203130861240 16

For the first experiment I found out that the shortest path based on the parameters is : 62.20313086124016.

In the figure (Fig. 18) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of 14 is for the distance around 70 to 75 meanwhile for the shortest distance of less than 60 we have a frequency of 1.

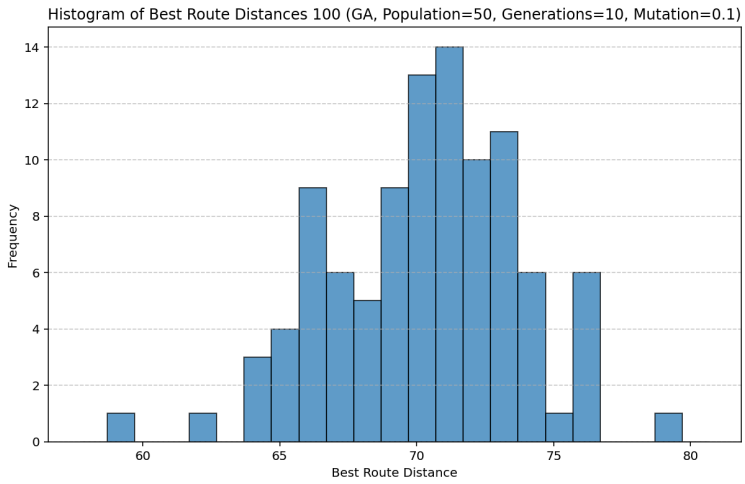


Figure 18. Histogram of the best route distance of GA with population = 50, generations=10 and mutation = 0.01.

Second experiment :

- ➔ Population was set to 300.
- ➔ Generations were set to 10.
- ➔ And the mutation rate was set to 0.01.

So I have increased the population and I have decreased the mutation rate while keeping the same generations as in the initial experiment.

Population	Generation	Mutation	Best route found:	Best route distance:
300	10	0.01	[15, 18, 7, 11, 8, 2, 9, 17, 6, 13, 0, 14, 12, 4, 10, 19, 3, 1, 16, 5]	64.22687726807101
300	10	0.01	[3, 8, 15, 5, 4, 9, 12, 17, 7, 18, 11, 19, 16, 6, 14, 0, 13, 10, 1, 2]	59.73520312307142
300	10	0.01	[12, 17, 9, 5, 19, 2, 13, 4, 0, 10, 18, 14, 1, 16, 6, 15, 7, 11, 8, 3]	52.13534518993933

For the second experiment I found out that the shortest path based on the parameters is : 52.13534518993933.

In the figure (Fig. 19) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of almost 25 is for the distance around 62 to 63 while for the shortest distance of 58 we have a frequency of less than 5.

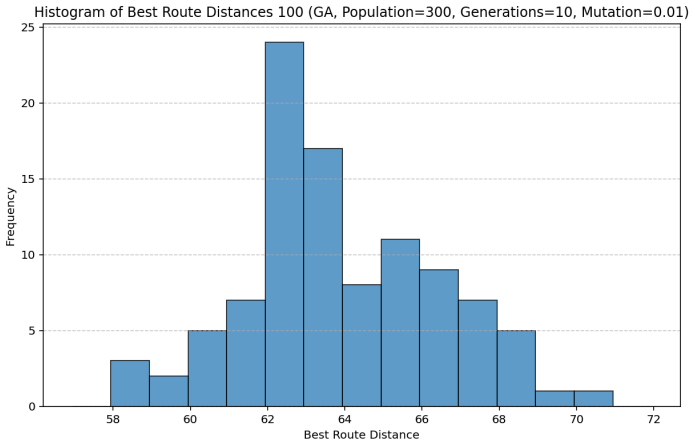


Figure 19. Histogram of the best route distance of GA with population = 300, generations=10 and mutation = 0.01.

Third experiment :

- ➔ Population was set to 1000.
- ➔ Generations were set to 100.
- ➔ And the mutation rate was set to 0.01.

So I have increased the population and I have decreased the mutation rate while keeping the same generations as in the initial experiment.

Population	Generation	Mutation	Best route found:	Best route distance:
1000	100	0.01	[4, 1, 2, 3, 12, 5, 8, 15, 18, 19, 7, 11, 14, 0, 16, 6, 10, 13, 17, 9]	63.613334430975875
1000	100	0.01	[4, 10, 0, 2, 6, 16, 5, 18, 11, 15, 12, 3, 17, 9, 8, 7, 19, 14, 1, 13]	56.49526950312853

For the third experiment I found out that the shortest path based on the parameters is : 56.49526950312853. And it has the following graph (Fig. 20):

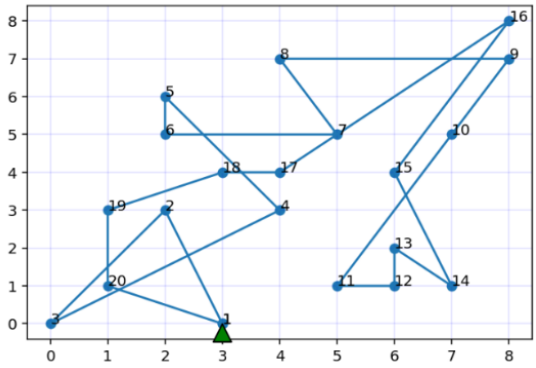


Figure 20. The graph of the shortest path of GA with population=1000, generations = 100 and mutation = 0.01.

Fourth experiment :

- ➔ Population was set to 500.
- ➔ Generations were set to 50.
- ➔ And the mutation rate was set to 0.09.

In this experiment I have increased all the parameters.

Population	Generation	Mutation	Best route found:	Best route distance:
500	50	0.09	[9, 17, 8, 5, 19, 0, 10, 16, 6, 15, 18, 11, 7, 12, 4, 14, 13, 1, 2, 3]	58.4954364847063

For the fourth experiment I found out that the shortest path based on the parameters is : 58.4954364847063. And it has the following graph (Fig 21):

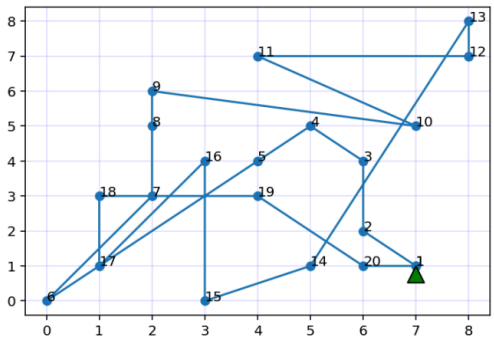


Figure 21. The graph of the shortest path of GA with population=500, generations = 50 and mutation = 0.09.

This experiment was a bit strange because no matter how many iterations I made, none of them produced a better result than the first iteration.

Fifth experiment :

- ➔ Population was set to 500.
- ➔ Generations were set to 50.
- ➔ And the mutation rate was set to 0.01.

So I have increased the population and I have decreased the mutation rate and also I have increased the generations.

population	generation	mutation	Best route found:	Best route distance:
500	50	0.01	[3, 17, 4, 2, 13, 10, 19, 0, 14, 15, 5, 8, 11, 7, 16, 18, 6, 1, 9, 12]	62.904772894293 55
500	50	0.01	[17, 19, 18, 6, 16, 0, 13, 2, 12, 5, 15, 11, 7, 14, 8, 10, 4, 1, 9, 3]	59.560174821236 78
500	50	0.01	[14, 16, 19, 10, 8, 9, 3, 6, 18, 7, 11, 15, 13, 4, 0, 1, 2, 5, 17, 12]	59.197657285310 2

For the fifth experiment I found out that the shortest path based on the parameters is : 59.1976572853102.

In the figure (Fig. 22) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 16 is for the distance around 61 to 63 while for the shortest distance of 52.5 it has a frequency of 1.

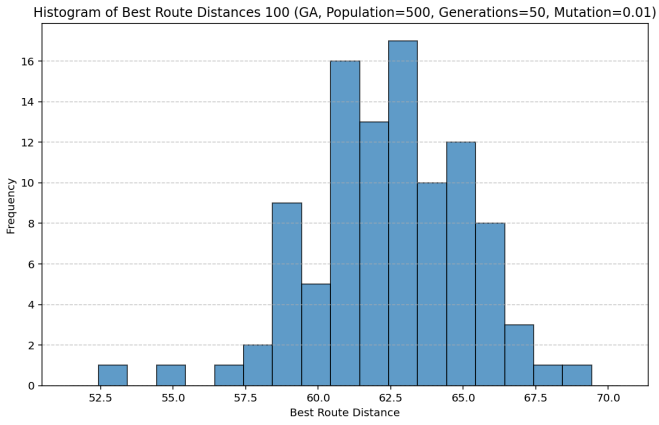


Figure 22. Histogram of the best route distance of GA with population = 500, generations=50 and mutation = 0.01.

8. INTERPRETATION OF THE GENETIC ALGORITHM EXPERIMENTS RESULTS

1. Increasing the Population Size Improves Results

- ➔ In the first experiment (50 population, 10 generations, 0.01 mutation rate), the best distance was 32.92.
- ➔ Then when the population increased to 500, the best distance improved to 31.20.

This happens because a larger population means more diverse solutions, helping GA avoid local optima.

2. Increasing Generations Helps, but Has Diminishing Returns

- ➔ 50 generations (50 population, 50 generations, 0.01 mutation rate) was 31.96.
- ➔ 100 generations (50 population, 100 generations, 0.01 mutation rate) was 32.48.
- ➔ 1000 generations (50 population, 1000 generations, 0.01 mutation rate) was 33.83

We see that the distance is growing as the generations grow. Also when increasing the generations the GA took a lot longer to compute especially when the number of cities increased. As more generations might help, but after a certain point, GA converges early and stagnates.

3. Mutation Rate

- ➔ 0.01 mutation rate (best results: 31.30).
- ➔ 0.05 mutation rate (best result: 33.82).
- ➔ 0.09 mutation rate (best result: 34.56).

As mutation rate increases the results tend to get worse, this happens because a high mutation rate makes GA too random, preventing convergence.

4. GA Scales Poorly with More Cities

- ➔ Combinatorial Explosion

With 10 cities, there are 10! (3.6 million) possible routes.With 20 cities, there are 20! (2.43×10^{18}) possible routes.As the number of possible solutions increases, it makes it harder for GA to explore all good possibilities.

- ➔ More Local Minima

With more cities, GA is more likely to get stuck in suboptimal solutions (local minima). So a mutation rate that works well for 10 cities is not enough for 20 cities or more to escape these traps.

- ➔ Higher Generations Needed for Convergence

When cities increase, the algorithm might need more generations to find a good solution.As shown in my experiments 50 generations were good for 10 cities, but for 20 cities, GA needed 100 or more generations for significant improvements.

9. PARTICLE SWARM OPTIMISATION EXPERIMENTS AND THEIR RESPECTIVE RESULTS

1. **First experiment (10 cities)**

For the first experiments I have taken in consideration 10 cities and the coordinates are : (0, 0), (1, 3), (4,3), (6, 1), (3, 0), (5, 5), (2, 6), (8, 8), (6, 4), (7, 1) .

First experiment :

- ➔ Particle Best Attractor was set to 0.1.
- ➔ Particles were set to 20.
- ➔ And the iterations were set to 10.

This is the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:
0.1	20	10	[0, 4, 1, 2, 6, 8, 5, 7, 3, 9]	38.691270456565 93
0.1	20	10	[6, 5, 8, 3, 9, 7, 2, 4, 0, 1]	34.537516252344 936
0.1	20	10	[9, 7, 8, 5, 6, 1, 0, 4, 2, 3]	32.434955094657 85

The method I followed is as follows. First I ran the first experiment and saved its data :

1. The best route found .
2. And the best route distance. The first row contains the data after the first iteration. Then I tried to find the shortest path for the corresponding experiment by keeping track of the results. So, whenever I found a path that was shorter than the previous path, I noted it in the table. I repeated this several times until I found the shortest path from the iterations I had done.In this case the shortest distance is : 32.43495509465785. And it has this graph(Fig. 23) :

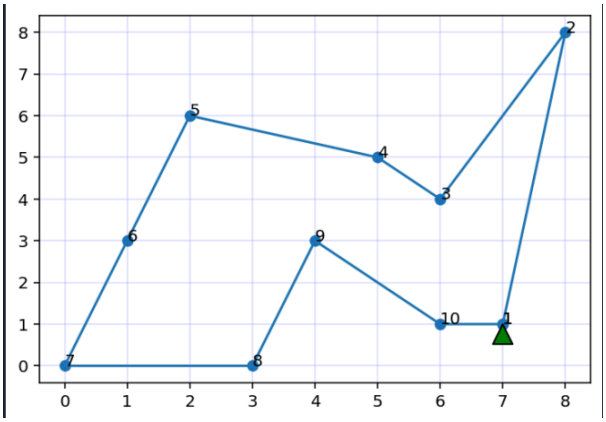


Figure 23. The graph of the shortest path of PSO with Particle Best Attractor =0.1, Particle = 20 and Iterations= 10.

Second experiment :

- ➔ Particle Best Attractor was set to 0.1.
- ➔ Particles were set to 70.
- ➔ And the iterations were set to 10.

So I have increased the particles while keeping the particle best attractor and the iterations the same as in the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:
0.1	70	10	[8, 6, 7, 5, 2, 1, 0, 4, 3, 9]	33.762232920460 55
0.1	70	10	[9, 2, 8, 7, 5, 6, 1, 0, 4, 3]	31.205506535756 157

For the second experiment I found out that the shortest path based on the parameters is : 31.205506535756157.

In the figure (Fig. 24) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 25 is for the distance around 35 to 36 while for the shortest distance of 32 it has a frequency of 5.

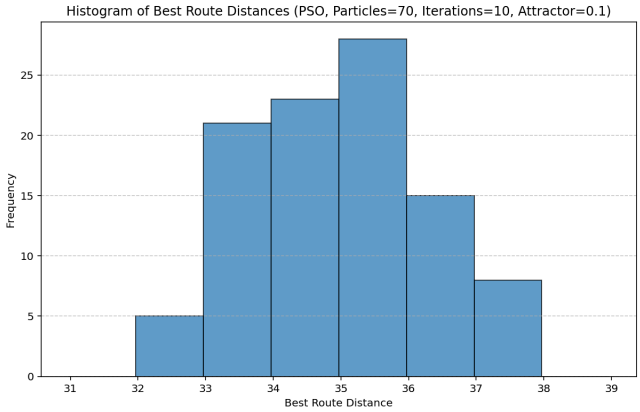


Figure 24. Histogram of the best route distance of PSO with Particle Best Attractor = 0.1, Particle =70 and Iterations= 10.

Third experiment :

- ➔ Particle Best Attractor was set to 0.1.
- ➔ Particles were set to 150.
- ➔ And the iterations were set to 50.

So I have increased the particles and the iterations while keeping the particle best attractor the same as in the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:
0.1	150	50	[2, 4, 0, 1, 6, 5, 7, 8, 3, 9]	31.96943855825637
0.1	150	50	[9, 8, 7, 5, 2, 6, 1, 0, 4, 3]	31.205506535756154

For the third experiment I found out that the shortest path based on the parameters is : 31.205506535756154.

In the figure (Fig. 25) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 50 is for the distance around 31 to 32 and this group also represents the shortest distance of this experiment.

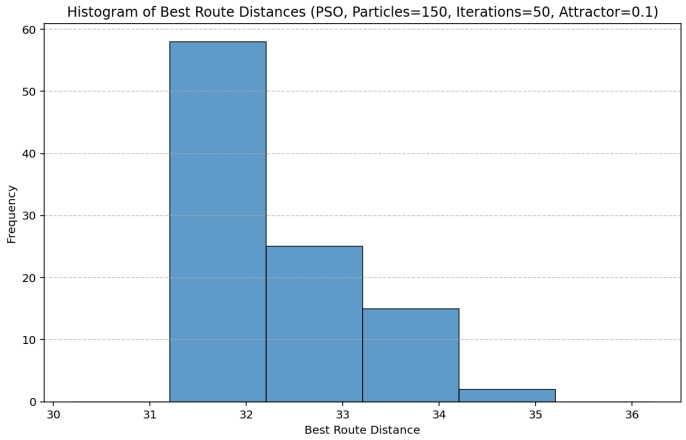


Figure 25. Histogram of the best route distance of PSO with Particle Best Attractor = 0.1, Particle =150 and Iterations= 50.

Fourth experiment :

- ➔ Particle Best Attractor was set to 0.5.
- ➔ Particles were set to 150.
- ➔ And the iterations were set to 50.

So I have increased the particles , the particle best attractor and the iterations in this experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:
0.5	150	50	[2, 4, 0, 1, 6, 7, 5, 8, 9, 3] Best Route Distance: 31.458947335248848	
0.5	150	50	[1, 6, 5, 7, 8, 2, 3, 9, 4, 0] Best Route Distance: 31.389210350487645	
0.5	150	50	[0, 1, 6, 2, 5, 7, 8, 9, 3, 4] Best Route Distance: 31.20550653575616	

0.5	150	50	[1, 6, 2, 5, 7, 8, 9, 3, 4, 0]	31.205506535756154

For the fourth experiment I found out that the shortest path based on the parameters is : 31.205506535756154.

In the figure(Fig. 26) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 50 is for the distance around 31 to 32 and this group also represents the shortest distance of this experiment.

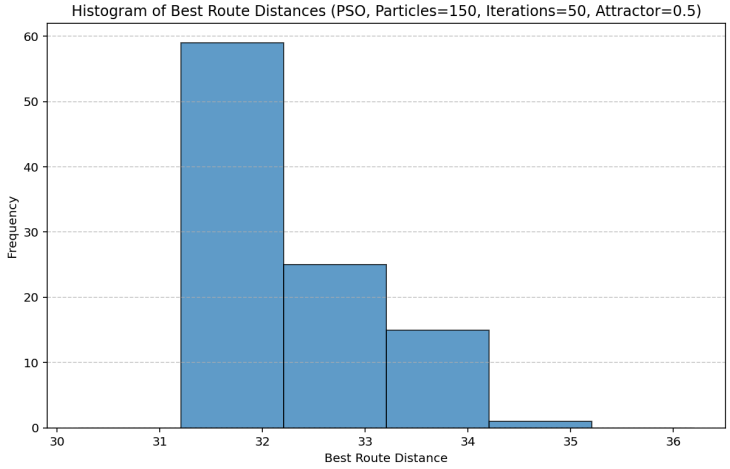


Figure 26. Histogram of the best route distance of PSO with Particle Best Attractor = 0.5, Particle =150 and Iterations= 50.

2.First experiment (20 cities)

For the next experiments I added more cities by increasing the number of the coordinates so now the new coordinates are : (0, 0), (1, 3), (4, 3), (6, 1), (3, 0),(5, 5), (2, 6), (8, 8), (6, 4), (7, 1),(2, 3), (8, 7), (5, 1), (1,1), (3, 4),(7, 5), (2, 5), (6, 2), (4, 7), (4, 4).

First experiment :

- ➔ Particle Best Attractor was set to 0.1.
- ➔ Particles were set to 20.
- ➔ And the iterations were set to 10.

This is the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:
0.1	20	10	[9, 12, 3, 16, 10, 2, 17, 13, 0, 4, 8, 18, 6, 19, 15, 14,	65.27766544146006

			1, 7, 11, 5]	
0.1	20	10	[15, 11, 9, 3, 12, 5, 7, 18, 6, 4, 13, 0, 2, 19, 16, 14, 17, 8, 1, 10]	61.39370600427
0.1	20	10	[16, 3, 12, 9, 17, 18, 10, 0, 1, 4, 13, 14, 15, 11, 7, 5, 8, 6, 2, 19]	60.473151098632876

For the first experiment I found out that the shortest path based on the parameters is : 60.473151098632876.

In the figure(Fig. 27) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 17.5 is for the distance around 65 to 70 and while for the shortest distance of less than 55 it has a frequency of less than 2.5.

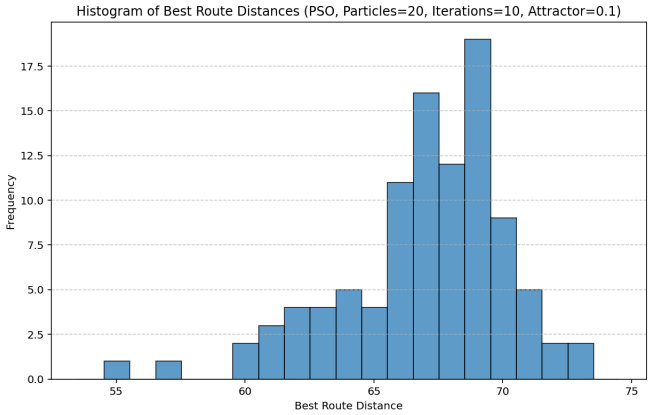


Figure 27. Histogram of the best route distance of PSO with Particle Best Attractor = 0.1, Particle =20 and Iterations= 10.

Second experiment :

- ➔ Particle Best Attractor was set to 0.9.
- ➔ Particles were set to 20.
- ➔ And the iterations were set to 10.

So I have increased the best particle attractor while keeping the iterations and the particles the same as in the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:
0.9	20	10	[10, 1, 14, 6, 0, 13, 12, 17, 19, 8,	62.330071402392704

			15, 7, 3, 2, 4, 9, 5, 11, 18, 16]	
0.9	20	10	[19, 9, 12, 16, 6, 17, 7, 11, 18, 15, 2, 14, 3, 8, 5, 10, 1, 13, 0, 4]	61.64909108319661
0.9	20	10	[6, 19, 17, 11, 7, 15, 1, 16, 18, 14, 2, 5, 8, 12, 4, 10, 13, 0, 3, 9]	61.18485508166548

For the second experiment I found out that the shortest path based on the parameters is : 61.18485508166548. And it has the following graph:

In the figure(Fig. 28) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 17.5 is for the distance around 65 to 70 and while for the shortest distance of less than 55 it has a frequency of less than 2.5.

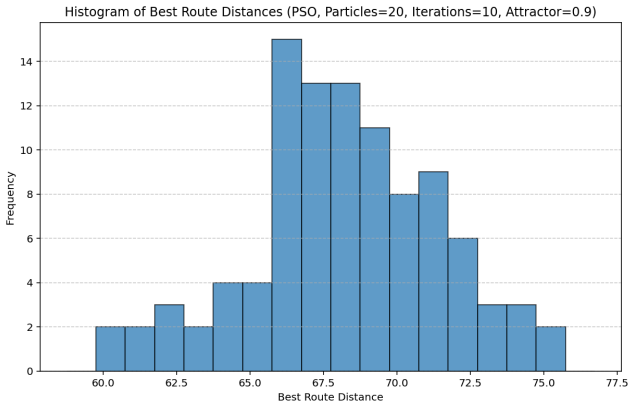


Figure 28. Histogram of the best route distance of PSO with Particle Best Attractor = 0.9, Particle =20 and Iterations= 10.

Third experiment :

- ➔ Particle Best Attractor was set to 0.1.
- ➔ Particles were set to 200.
- ➔ And the iterations were set to 10.

So I have increased the particles and I have kept the best particle attractor and the iterations the same as in the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:
0.1	200	10	[13, 0, 11, 7, 5, 8,	64.529283772094

			15, 10, 19, 1, 12, 9, 4, 17, 3, 14, 2, 16, 6, 18]	84
0.1	200	10	[13, 4, 3, 8, 11, 15, 7, 9, 17, 12, 1, 16, 18, 6, 10, 19, 5, 14, 0, 2]	61.566337314994
0.1	200	10	[1, 6, 14, 7, 11, 15, 2, 19, 17, 8, 3, 9, 10, 13, 0, 4, 12, 5, 18, 16]	54.043593657063 93

For the third experiment I found out that the shortest path based on the parameters is : 54.04359365706393.

In the figure (Fig. 29) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 20 is for the distance around 62 to 64 and while for the shortest distance of less than 54 it has a frequency of less than 2.5.

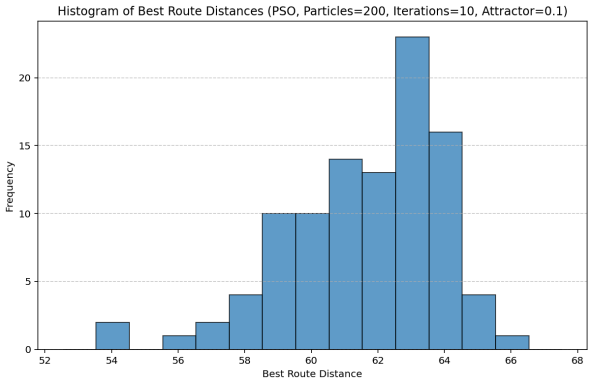


Figure 29. Histogram of the best route distance of PSO with Particle Best Attractor = 0.1, Particle =200 and Iterations= 10.

Fourth experiment :

- ➔ Particle Best Attractor was set to 0.1.
- ➔ Particles were set to 500.
- ➔ And the iterations were set to 100.

So I have increased the particles and the iterations while keeping the best particle attractor the same as in the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:
0.1	500	100	[13, 0, 4, 9, 10, 15, 11, 7, 18, 8, 5, 6, 16, 14, 19, 1, 2, 17, 12, 3]	54.075637665797 03
0.1	500	100	[6, 16, 15, 11, 7, 18, 5, 9, 3, 4, 12, 0, 13, 10, 2, 1, 17, 8, 19, 14]	52.550111717843 23
0.1	500	100	[9, 17, 3, 8, 12, 19, 18, 14, 6, 16, 4, 1, 0, 13, 10, 2, 5, 7, 11, 15]	52.369021151594 92

For the third experiment I found out that the shortest path based on the parameters is : 52.36902115159492. And it has the following graph (Fig. 30):

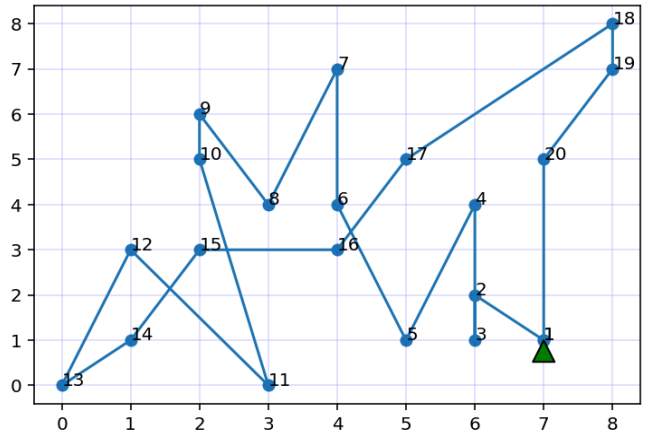


Figure 30. The graph of the shortest path of PSO with Particle Best Attractor =0.1, Particle = 500 and Iterations= 100.

Fifth experiment :

- ➔ Particle Best Attractor was set to 0.1.
- ➔ Particles were set to 500.
- ➔ And the iterations were set to 1000.

So I have increased the particles and the iterations while keeping the best particle attractor the same as in the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:

0.1	500	1000	[11, 7, 3, 9, 17, 12, 5, 2, 1, 0, 13, 4, 10, 16, 14, 19, 15, 8, 18, 6]	52.23452841991271
0.1	500	1000	[5, 18, 14, 19, 11, 7, 15, 8, 3, 2, 10, 0, 4, 13, 6, 16, 1, 12, 9, 17]	53.02859790655324
0.1	500	1000	[12, 4, 13, 0, 1, 14, 10, 17, 8, 2, 9, 3, 15, 11, 7, 5, 16, 6, 18, 19]	49.6637935239002

For the fifth experiment I found out that the shortest path based on the parameters is : 49.66379352390028. And it has the following graph(Fig. 31):

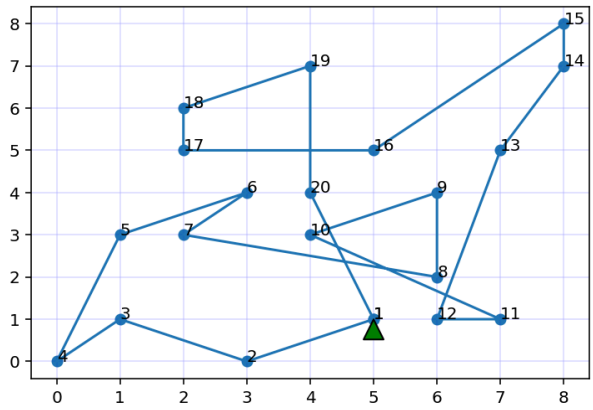


Figure 31. The graph of the shortest path of PSO with Particle Best Attractor =0.1, Particle = 500 and Iterations= 1000.

3. First experiment (25 cities)

For the next experiments I added more cities by increasing the number of the coordinates so now the new coordinates are : (0, 0), (1, 3), (4, 3), (6, 1), (3, 0),(5, 5), (2, 6), (8, 8), (6, 4), (7, 1), (2, 3), (8, 7), (5,1), (1, 1), (3, 4), (7, 5), (2, 5), (6, 2), (4, 7), (4, 4), (5, 6), (8, 9), (9, 1), (5, 3), (1, 2).

First experiment :

- ➔ Particle Best Attractor was set to 0.1.
- ➔ Particles were set to 20.
- ➔ And the iterations were set to 10.

This is the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:

0.1	20	10	[23, 19, 4, 0, 3, 1, 13, 21, 7, 20, 9, 12, 22, 24, 10, 6, 15, 11, 18, 17, 8, 5, 14, 16, 2]	88.71575425664749
0.1	20	10	[5, 16, 24, 1, 14, 2, 17, 9, 12, 23, 6, 11, 8, 3, 15, 19, 7, 21, 18, 20, 13, 4, 10, 0, 22]	85.34164282364154
0.1	20	10	Best Route Found: [19, 10, 4, 2, 12, 23, 14, 5, 6, 18, 16, 20, 15, 11, 8, 22, 3, 9, 24, 17, 21, 7, 1, 0, 13]	81.76253516678364

For the first experiment I found out that the shortest path based on the parameters is : 81.76253516678364.

In the figure(Fig. 32) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 16 is for the distance around 87.5 to 90 and while for the shortest distance of 80 it has a frequency of 2.

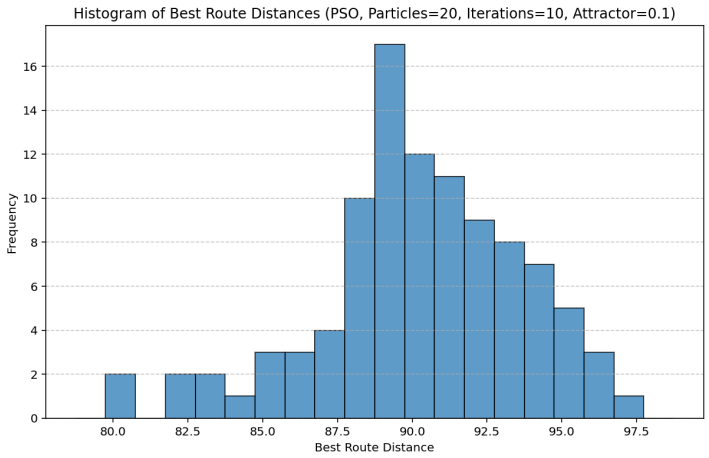


Figure 32. Histogram of the best route distance of PSO with Particle Best Attractor = 0.1, Particle =20 and Iterations= 10.

Second experiment :

- ➔ Particle Best Attractor was set to 0.9.
- ➔ Particles were set to 20.

➔ And the iterations were set to 10.

So I have increased the particle best attractor while keeping the particles and the iterations the same as in the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:
0.9	20	10	[2, 3, 9, 12, 24, 22, 4, 13, 16, 20, 18, 21, 17, 8, 6, 11, 7, 5, 23, 0, 14, 1, 10, 15, 19]	90.19646521096469
0.9	20	10	[21, 9, 22, 13, 0, 1, 14, 2, 19, 6, 23, 8, 16, 11, 5, 20, 18, 15, 24, 10, 3, 4, 17, 12, 7]	85.23965799794838
0.9	20	10	[19, 4, 8, 17, 12, 5, 21, 6, 16, 13, 24, 1, 22, 2, 14, 23, 15, 18, 20, 11, 7, 9, 3, 0, 10]	84.65620556222963

For the second experiment I found out that the shortest path based on the parameters is :84.65620556222963.

In the figure(Fig. 33) below I have modified the code to make 100 iterations for these specific parameters of this experiment.We can see that the highest frequency of more than 17.5 is for the distance around 90 to 95 and while for the shortest distance of less than 75 it has a frequency of less than 2.5.

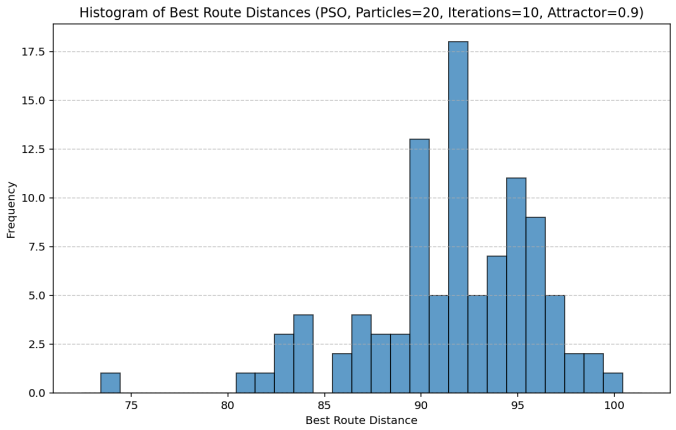


Figure 33. Histogram of the best route distance of PSO with Particle Best Attractor = 0.9, Particle =20 and Iterations= 10.

Third experiment :

- ➔ Particle Best Attractor was set to 0.1.
- ➔ Particles were set to 500.
- ➔ And the iterations were set to 100.

So I have increased the particles and the iterations while keeping the best particle attractor the same as in the initial experiment.

Particle Best Attractor	Particle	Iterations	Best Route Found:	Best Route Distance:
0.1	500	100	[16, 14, 22, 9, 1, 6, 5, 15, 20, 17, 24, 13, 10, 19, 2, 3, 11, 7, 21, 18, 8, 23, 12, 4, 0]	75.868953716216 91
0.1	500	100	[19, 12, 9, 6, 18, 20, 11, 7, 21, 13, 0, 1, 24, 10, 2, 16, 5, 8, 15, 17, 14, 23, 3, 22, 4]	73.769442565764 54

For the third experiment I found out that the shortest path based on the parameters is :73.76944256576454. And it has the following graph(Fig. 34):

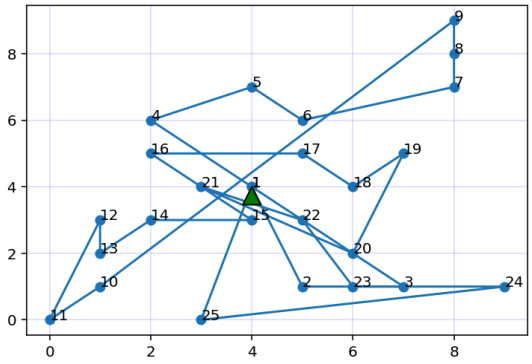


Figure 34. The graph of the shortest path of PSO with Particle Best Attractor =0.1, Particle = 500 and Iterations= 100.

10. INTERPRETATION OF THE PSO EXPERIMENTS RESULTS

1. Increasing Particles Improves Results

When I first started the experiments with 20 particles the best distance found was 32.43 but then when I increased the particles to 70 the best distance found improved to 31.20. This happens because a larger swarm size allows PSO to explore better solutions, but after a certain point, adding more particles has diminishing returns.

2. Increasing the Number of Iterations

Based on my results I think that increasing the number of iterations also improves the distance. For example when I set the number of iterations to 1000 for 20 cities a got the shortest path 49.66.

3. Increasing the Particle Best Attractor

As for increasing the attractor I found out that the best distance was when the attractor was low:

1. Attractor = 0.1 (balancing personal best & swarm best) performed best.
2. Attractor = 0.9 (focusing more on personal best) gave worse results.

This happens because if particles focus too much on their personal best (high attractor), they don't explore new solutions. A lower attractor makes them follow the global best more effectively.

4. Increasing the Number of Cities

- ➔ When the number of cities is 10 ; PSO quickly finds optimal paths with low particle count & iterations.
- ➔ When the number of cities is 20 ; Increasing particles and iterations helps, but with diminishing returns.
- ➔ When the number of cities is 25 For large cities ; PSO struggles more, requiring larger swarms and more iterations.

11. COMPARING GENETIC ALGORITHM WITH PARTICLE SWARM OPTIMISATION

In order to solve the TSP I have used both algorithms the GA and the PSO with different parameter configurations. The key goal was to find the shortest route while understanding how each algorithm performs under different conditions.

1. Best Performance Comparison

Number of Cities	Best Distance from GA	Best Distance from PSO
10	31.30	31.205506535756154
20	52.13	49.66

Based on these results the Particle Swarm Optimisation performed better than the Genetic Algorithm. As the number of cities increased, Genetic Algorithms performance tended to get lower, whereas Particle Swarm Optimisation, continued to improve with more iterations.

2.Effect of Increasing Population (GA) vs. Particles (PSO)

Population/Particles	Best Distance (GA)	Best Distance (PSO)
50 (GA) / 20 (PSO)	32.48	32.43
100 (GA) / 70 (PSO)	31.96	31.205506535756157

500 (GA) / 150 (PSO)	31.205506553575616	31.205506535756154

Based on these results again the PSO performed better than the GA by giving the shortest distance.

3. Effect of Increasing Generations (GA) vs. Iterations (PSO)

Generations (GA) / Iterations (PSO)	Best Distance (GA)	Best Distance (PSO)
10	31.30	31.205506535756157
50	31.20550653575616	31.205506535756154
100	32.48	31.20
1000	33.83	49.66 (for 20 cities)

Again we can see that the PSO performed better than the GA and also for PSO as the iterations increased the PSO performance was improved as for the GA as the generations increased the GA tended to perform worse.

4. Effect of Mutation Rate (GA) vs. Particle Best Attractor (PSO)

Mutation Rate (GA) / Particle Best Attractor (PSO)	Best Distance (GA)	Best Distance (PSO)
0.01 (GA) / 0.1 (PSO)	31.20550653575616	31.205506535756154
0.05 (GA) / 0.5 (PSO)	31.30	31.38
0.09 (GA) / 0.9 (PSO)	33.82	61.18 (20 cities)

As seen from the results GA was more sensitive to mutation rate changes. Meanwhile PSO's performance dropped when the best attractor was too high, making particles too focused on their own best route.

12. ADVANTAGES vs DISADVANTAGES

1. Genetic Algorithm (GA)

Advantages	**Disadvantages**
1.Good at Finding Near-Optimal Solutions: Since GA explores different possibilities through generations, it often finds a route that is close to the best one, even if it doesn't guarantee the optimal best solution.	**1.Slower as the Problem Grows**: As seen from the experiments when the number of cities increases, GA needed more generations and a bigger population to give us a solution, which makes it much longer to run.
2.Helps Avoid Local Minima : GA can avoid getting stuck in local minima by using mutation .	**2.Gets Stuck in Local Minima**: GA stops improving (gets stuck in a local minima) when the population becomes too similar too quickly (low diversity).
3.Can Be Adjusted for Different Problems: GA allows us to change the population size, mutation rate, generations, and selection methods to improve performance based on the problem we are solving.	**3.Mutation Rate Sensitivity**:If the mutation rate is too low, GA doesn't explore enough, but if it's too high, the solutions become too random and don't improve properly.
	4.Needs Careful Parameter Tuning:The performance of GA depends a lot on setting the right values for population size, mutation rate, and generations, which in some cases can be difficult.

2. Particle Swarm Optimization (PSO)

Advantages	Disadvantages
1.**Faster Than GA**: PSO generally finds good solutions faster than GA because particles adjust their paths instead of evolving through generations.	1.**Can Get Stuck in Local Minima**: If all the particles follow the best route too closely, they might stop exploring and get stuck in a less optimal solution.
2.**Keeps Improving Over Iterations**: Different from the GA, where performance can get worse after too many generations, PSO tries to improve as the number of iterations increases.	2.**Needs a Good Balance of Parameters**: The performance of PSO depends on finding the right balance between how much a particle follows its own best solution and how much it follows the global best.
3.**Less Randomness, More Efficiency**: Instead of	3.**Not as Flexible as GA**: While GA can be

depending on random mutations, PSO updates each particle's route based on its own best route and the best route found by others, making it more structured.	modified to fit different types of optimization problems, PSO is mainly designed for problems where solutions can be adjusted gradually.
4.**Performs Well on Large Problems**: When solving the Traveling Salesman Problem with many cities, PSO was found to give better results compared to GA.	4.**Performance Drops with High Attraction to Personal Best**: If particles focus too much on their own best solution instead of considering the global best, they might not explore enough new routes.

13. CONCLUSION

In conclusion, in this paper, I have analyzed the Traveling Salesman Problem and implemented both Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) to find solutions. Through multiple experiments, I tested different parameter settings and compared the results. The results show that increasing the population size in GA and the number of particles in PSO improves the solution, but only up to a certain point. GA showed that higher mutation rates negatively affect performance, while PSO performed better with a lower attractor value. When increasing the number of cities, both algorithms faced challenges, but PSO performed better overall in finding the shortest route.

Based on my findings, PSO consistently provided better results compared to GA, especially as the number of cities increased. This suggests that PSO is more effective in solving TSP when higher accuracy is needed. However, GA remains a useful approach, particularly when optimizing with constraints.

References

[1]“What is traveling salesman problem (TSP)? - Definition from WhatIs.com,” *WhatIs.com*[Online]. Available : https://www.techtarget.com/whatis/definition/traveling-salesman-problem

[2] M. Kuo, “Algorithms for the Travelling Salesman Problem,” *blog.routific.com* [Online], Nov. 08, 2023. Available: https://www.routific.com/blog/travelling-salesman-problem

[3] “PSO and the Traveling Salesman Problem: An Intelligent Optimization Approach,” *Arxiv.org*[Online], 2025. Available : https://arxiv.org/html/2501.15319v1 (accessed Feb. 11, 2025).

[4]M. Mishra, “The Curse of Local Minima: How to Escape and Find the Global Minimum,” *Medium*[Online], Jun. 01, 2023. Available: https://mohitmishra786687.medium.com/the-curse-of-local-minima-how-to-escape-and-find-the-global-minimum-fdabceb2cd6a

[5] “Particle Swarm Optimization - an overview | ScienceDirect Topics,” *www.sciencedirect.com*[Online]. Available: https://www.sciencedirect.com/topics/physics-and-astronomy/particle-swarm-optimization

[6] M. Z. Kagdi, “The P vs NP Problem”, *CEN 352 Lecture 6,* [PowerPoint slides]. Available: https://drive.google.com/file/d/1-smBz1P4C0xYRpD_PGd70sLfQV6n6MD9/view

[7]R. Shendy, “Traveling Salesman Problem (TSP) using Genetic Algorithm (Python),” **AI monks.io**[Online], Jan. 24, 2024. Available: https://medium.com/aimonks/traveling-salesman-problem-tsp-using-genetic-algorithm-fea640713758

[8]James Cutajar, “Evolutionary Algorithm for the Travelling Salesperson Problem (Genetic Algorithm),” *YouTube*, Dec. 03, 2023. Available : https://www.youtube.com/watch?v=Wgn_aPH3OEk

[9] A. Mirjalili, “Learn Particle Swarm Optimization (PSO) in 20 minutes,” *YouTube*. Mar. 30, 2018. Accessed: Nov. 01, 2020. [YouTube Video]. Available: https://www.youtube.com/watch?v=JhgDMAm-imI

[10] M. Z. Kagdi, “Evolutionary Computation”, *CEN 352 Lecture 7,* [PowerPoint slides]. Available: https://drive.google.com/file/d/1QisImbuyGFULnLlW1Za-ZcwuCX-d-o_j/view

[11] M. Z. Kagdi, “Swarm Intelligence”, *CEN 352 Lecture 9,* [PowerPoint slides]. Available: https://drive.google.com/file/d/1cTU6tvVBR-xxjOG5uoMD3HDvvhemKMdd/view