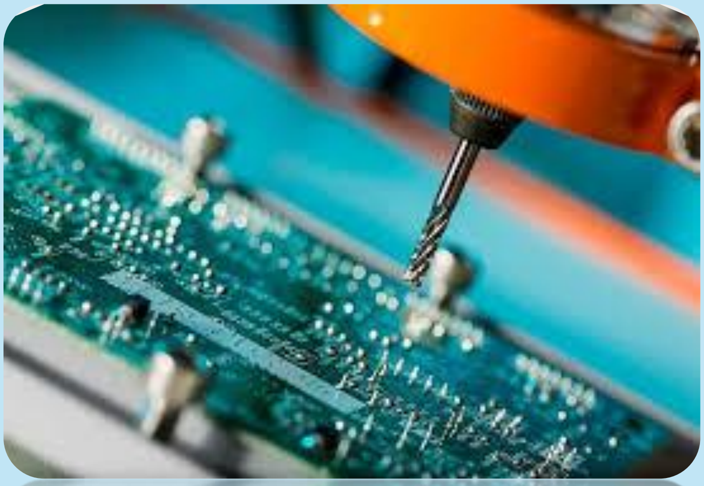
Drilling of printed circuit boards [TSP Problem] [PSO Algo]



Cairo University

Faculty of Computers and artificial intelligence

Department: Operations Research and Decision Support

Course Name: Computational Intelligence

Course Code: DS313/DS351

Instructors: Assoc. Prof. Ayman Ghoneim &

Assoc. Prof. Sally Kasem

NO.	NAME	ID
1	JANA RAAFAT ABDULHAMID	20221251
2	HAZEM MEDHAT ABDEL-AZIZ	20210116
3	MOHAMED ASHRAF RAMADAN	20210327
4	Mariam Mohamed Abd Elghany	20200529

Table Of Contents

Problem Formulation

Problem Definition

The PCB Drilling TSP (Traveling Salesman Problem) involves determining the most efficient route for a drilling machine to visit a set of predefined points on a Printed Circuit Board (PCB), each representing the location for drilling a hole. The objective is to find the shortest path that visits each point exactly once, starting and ending at the same location. This problem is of critical importance in PCB manufacturing to minimize drilling time and optimize production efficiency.

Goal

The goal of the PCB Drilling TSP problem is to find the optimal or near-optimal drilling sequence that minimizes the total drilling time by finding the shortest possible path that visits each drilling point. By solving this problem efficiently, manufacturers can optimize the drilling process, reduce production time and costs, and improve the quality and reliability of printed circuit boards.

About PSO

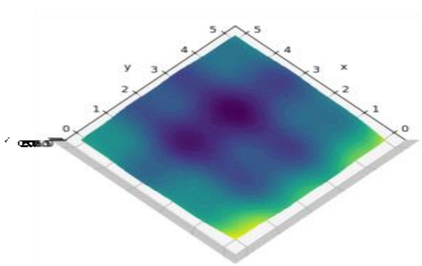
The program uses Particle Swarm Optimization (PSO) to solve the VRP, which is a metaheuristic optimization technique inspired by the social behavior of bird flocks or fish schools.

The PSO algorithm is a computational technique inspired by the collective behavior of natural organisms, such as birds or fishes, that move together to achieve a common goal. A group of particles (representing potential solutions) navigates through a problem's solution space to find the best possible solution. Each particle adjusts its position based on its own best-known solution (personal best) and the best solution discovered by the entire group (global best). This collaborative movement enables particles to converge toward optimal solutions over iterations.

During each iteration, particles update their positions and velocities according to mathematical formulas that balance exploration and exploitation. The velocity update equation incorporates inertia, cognitive, and social terms, which control the trade-off between exploration (searching for new promising areas) and exploitation (refining known solutions).

One of the key advantages of PSO is its ability to adaptively adjust its search behavior based on the dynamic characteristics of the optimization landscape. By dynamically tuning parameters such as inertia weight, acceleration coefficients, and neighborhood structures, PSO can effectively navigate complex and changing environments.

Despite its effectiveness, PSO has certain limitations, such as susceptibility to premature convergence, sensitivity to parameter settings, and difficulty in handling constraints. Researchers have developed various variants and hybrid approaches to address these challenges, including adaptive PSO, multi-objective PSO, and constrained PSO.



SYSTEM OVERVIEW

System Components:

System	Entity	Attributes	Activities	Events	State Variable
PCB Drilling Optimization System	-holes -Circuit Board	-Particles -Global Best (gbest) -Personal Best (pbest)	-Initialize velocity, Particles -Update Velocity -Update Position -Evaluate Personal and Global Best	-Calculate Fitness	-Best Route -Best Fit

System Analysis:

<u>Mathematical Model</u>

1- *Sets:*

• Let N be the set of drilling points.

• Let E be the set of edges connecting drilling points.

2- *Parameters:*

• Cij : Cost (time or distance) of traveling from drilling point ii to drilling point jj.

3- *Decision Variables:*

• Xij : Binary decision variable indicating whether the drill head travels directly from drilling point ii to drilling point jj.

$$x_{ij} = \begin{cases} 1 & \text{if the drill head travels from point } i \text{ to point } j \\ 0 & \text{otherwise} \end{cases}$$

4- *Objective Function:*

Minimize **∑i ∈N ∑j ∈N cij xij**

5- *Constraints:*

- Each drilling point is visited exactly once:

$$\sum_{j\in N, j\neq i} x_{ij} = 1 \quad \forall i \in N$$
$$\sum_{i\in N, i\neq j} x_{ij} = 1 \quad \forall j \in N$$

- The drill head returns to the starting point:

$$\sum_{j\in N, j\neq 1} x_{1j} = 1$$
$$\sum_{i\in N, i\neq 1} x_{i1} = 1$$

2. Encoders

The solution is encoded in real numbers encoding. Real number encoding offers flexibility in representing solutions, allowing for a wide range of possible values.

3. Operators

operators such as the velocity update formula and boundary handling mechanisms play a crucial role in guiding particles through the solution space. These operators enable effective exploration and exploitation, facilitating the convergence towards optimal solutions over iterative iterations.

The PSO algorithm used in the program uses two operators:

1. Mutation is applied in the **update_velocity** method through the random swapping of hole indices. This swapping operation introduces randomness and variability into the velocity update process, akin to the mutation operator in evolutionary algorithms.
2. In the **initialize_particles** method, the main operator used is **randomization**. This method initializes the positions of particles in the PSO population by randomly generating permutations of cities to visit. The permutation of cities ensures that each city is visited exactly once in the tour, which is a requirement of the TSP.

4. Constraint Handling

Constraint handling in PSO is crucial as it ensures that solutions generated by particles adhere to problem constraints. By incorporating constraint handling mechanisms, such as penalty functions or repair methods, PSO can effectively navigate constrained solution spaces, maintain feasibility, and prevent infeasible solutions. This enhances the algorithm's robustness and ability to find viable solutions in real-world optimization problems.

Here’s how we implement these constraints within the given PSO framework:

1. **Initialize Particles with Valid Routes:**

   The initialization ensures that each route starts and ends at the start_hole and visits each hole exactly once.

2. **Update Velocity and Position:**

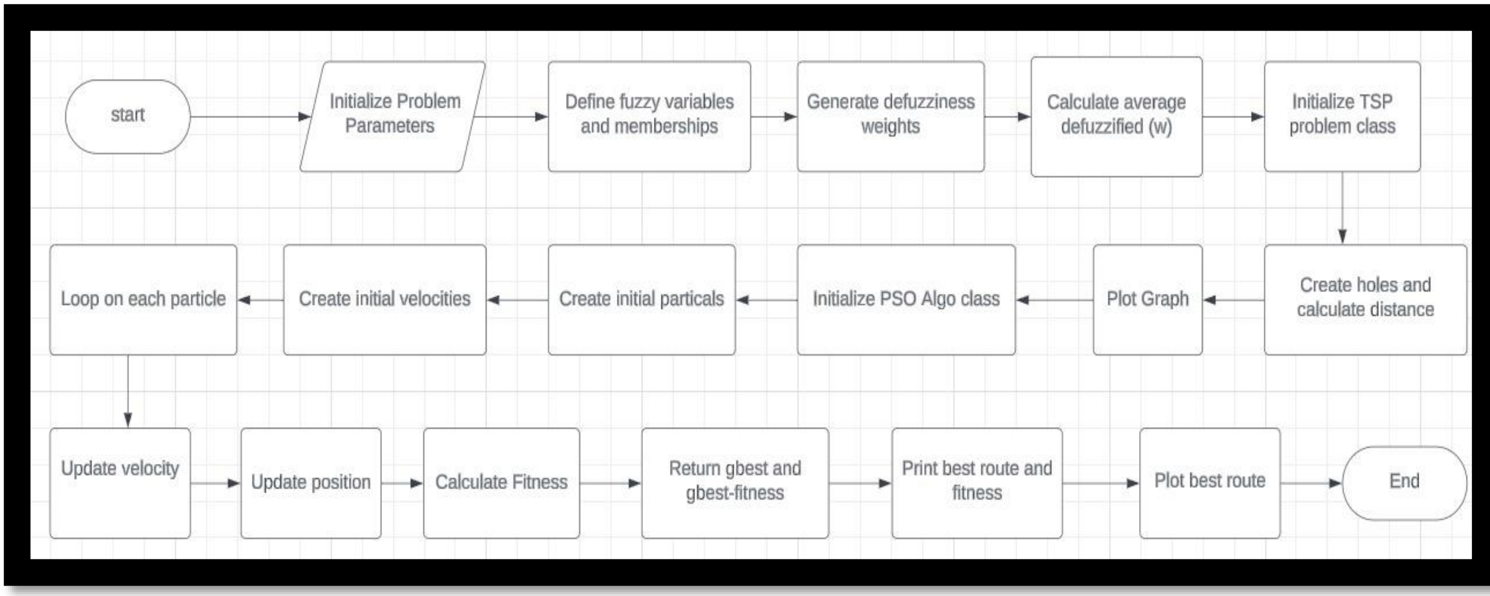
   The velocity update process introduces swaps based on the cognitive and social components. These swaps are constrained to avoid the first and last elements (which are the start hole and ensure a round trip).

5. Algorithm Parameters

Algorithm parameters in PSO govern the exploration and exploitation balance, convergence speed, and solution quality. Optimizing these parameters effectively influences the algorithm's performance and convergence behavior. These parameters are as follows:

- Number of iterations: It determines the duration of the optimization process, affecting the algorithm's convergence behavior and its ability to find optimal or near-optimal solutions.
- Number of Particles: It influences the diversity of the population and the exploration-exploitation trade-off, directly impacting the algorithm's convergence speed and solution quality.
- Number of holes: The number of holes in the PCB drilling TSP problem directly affects the complexity of the optimization task, influencing the solution space size and the computational resources required for the optimization process.
- Inertia weight: It controls the balance between exploration and exploitation, affecting the algorithm's ability to explore diverse regions of the solution space and converge towards optimal solutions.
- Cognitive component: It controls the particle's reliance on its personal best solution. A higher value of c_1 encourages particles to prioritize their own historical best solutions, leading to more exploitation of promising regions in the solution space.
- Social component: It governs the influence of the global best solution on each particle's velocity update. A higher value of c_2 encourages particles to explore the collective knowledge of the swarm by adjusting their velocities towards the global best solution.

6. Program flow



RESULTS & CONCLUSIONS

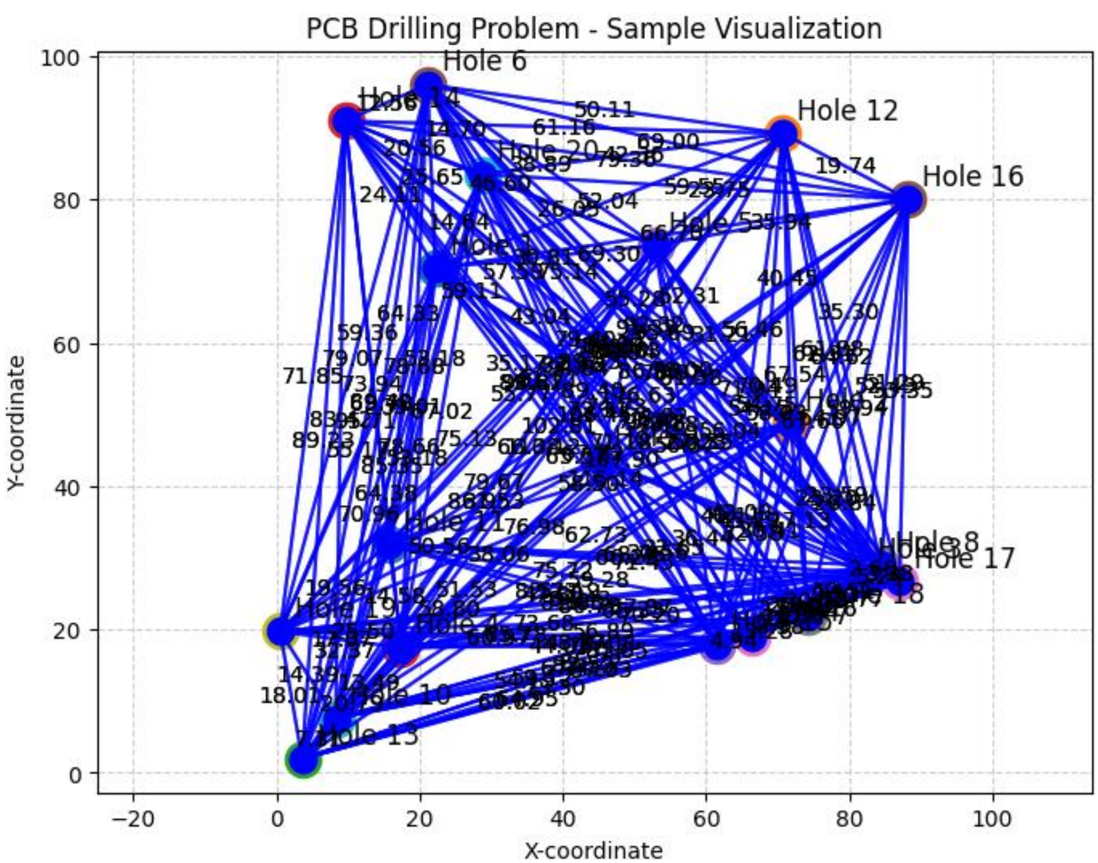
Output Testing:

In this part, we will Present the results of our implementation using three examples with different sizes of the problem, with fixed number of iterations.

1. Small Scale

Number of holes = 20

Number of Particles = 10

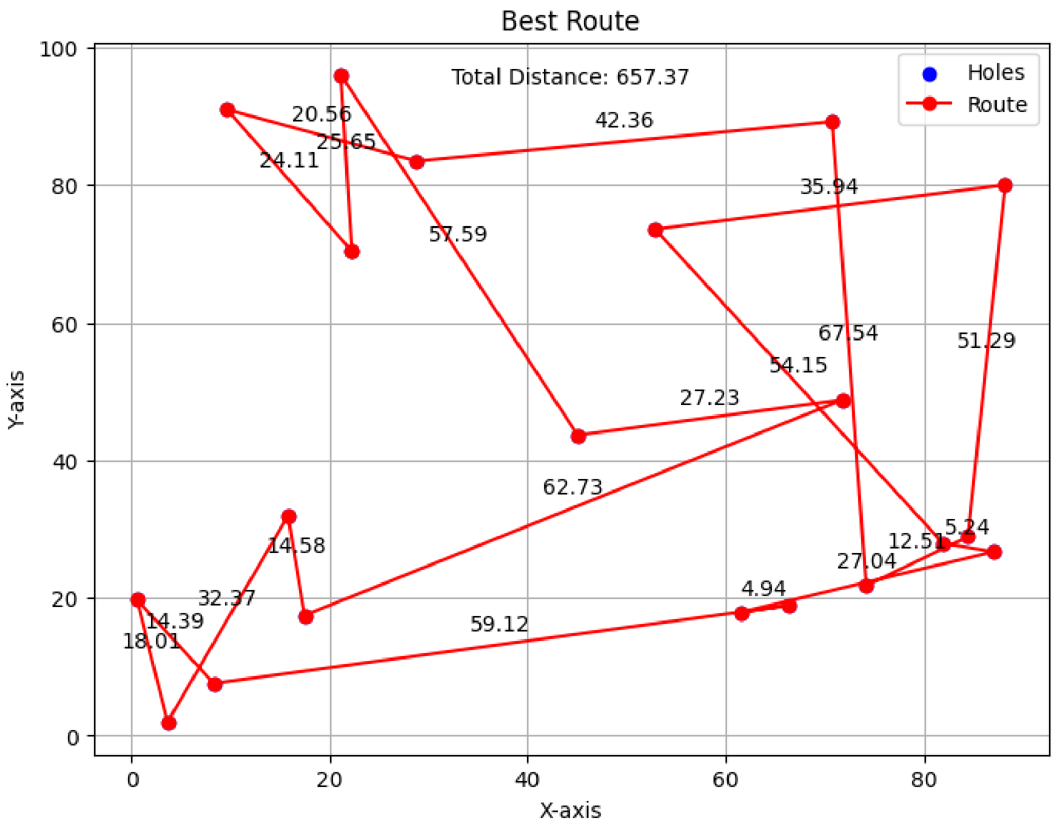


Results

The best route is as follows:

Best Route: [0, 13, 19, 11, 17, 7, 15, 4, 2, 16, 14, 6, 9, 18, 12, 10, 3, 1, 8, 5, 0]

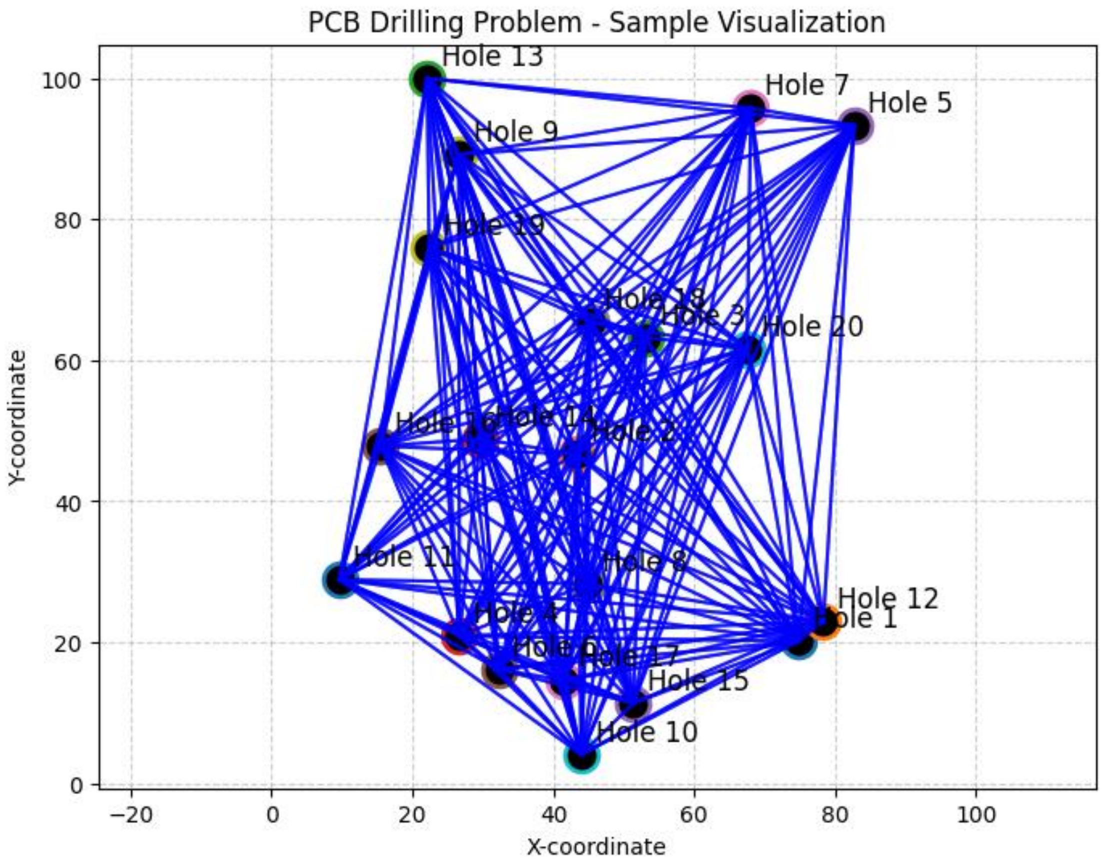
Best Fitness (Total Distance): 657.3679003869742



11. Small Scale

Number of holes = 20

Number of Particles = 20

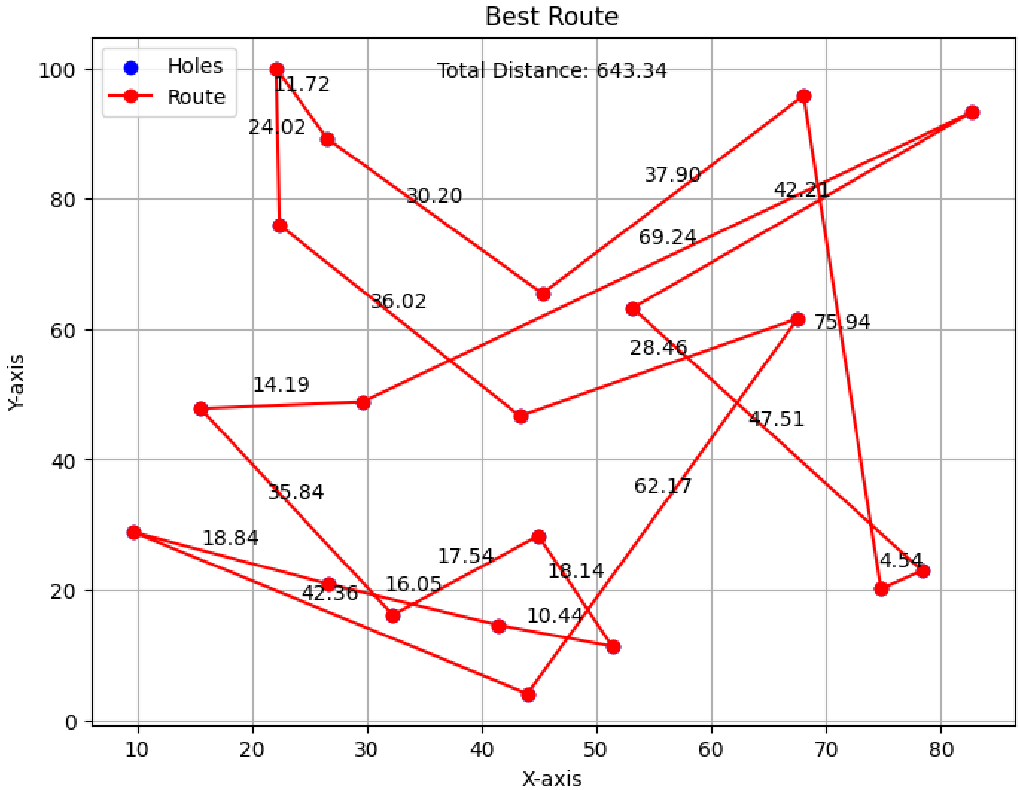


Results

The best route is as follows:

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

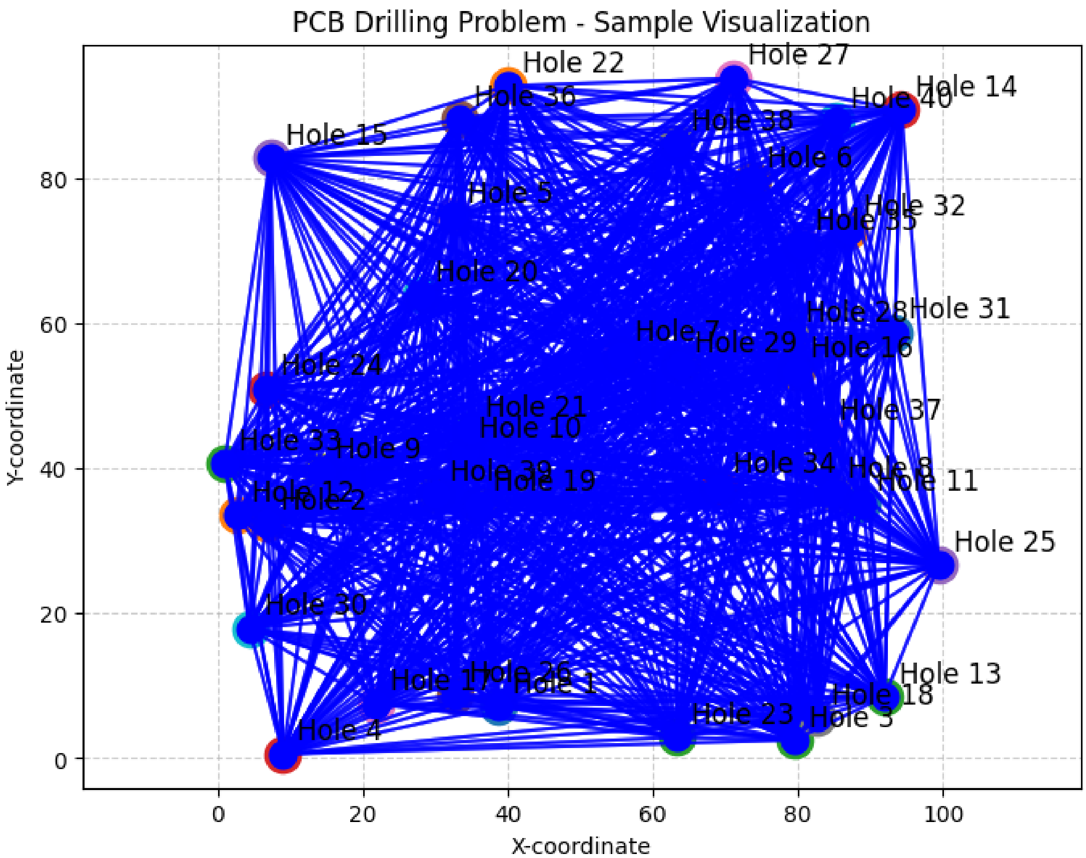
Best Fitness (Total Distance): 1749.5863706198163



1. Medium Scale

Number of holes = 40

Number of Particles = 10

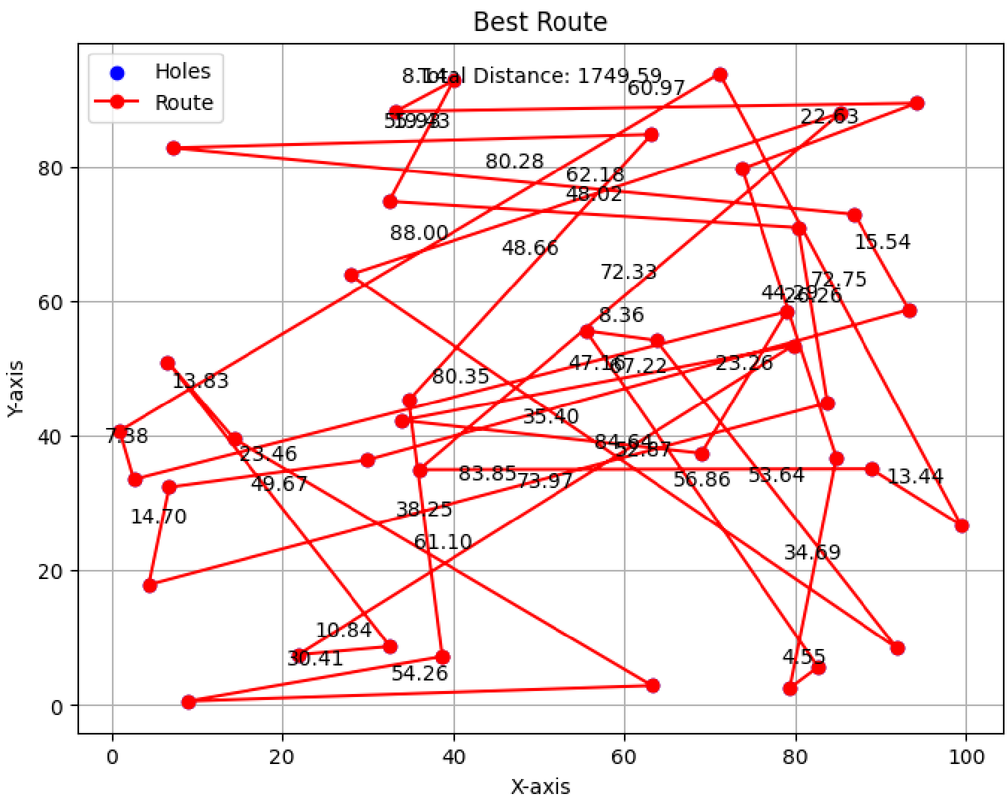


Results

The best route is as follows:

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

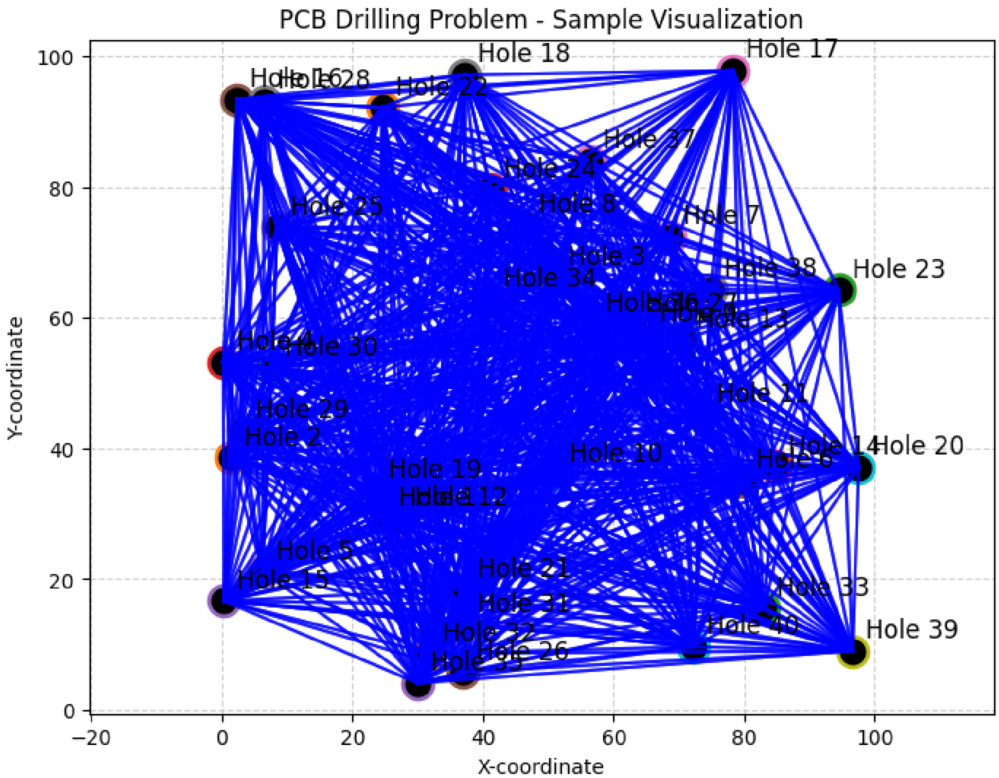
Best Fitness (Total Distance): 1749.5863706198163



11. Medium Scale

Number of holes = 40

Number of Particles = 20

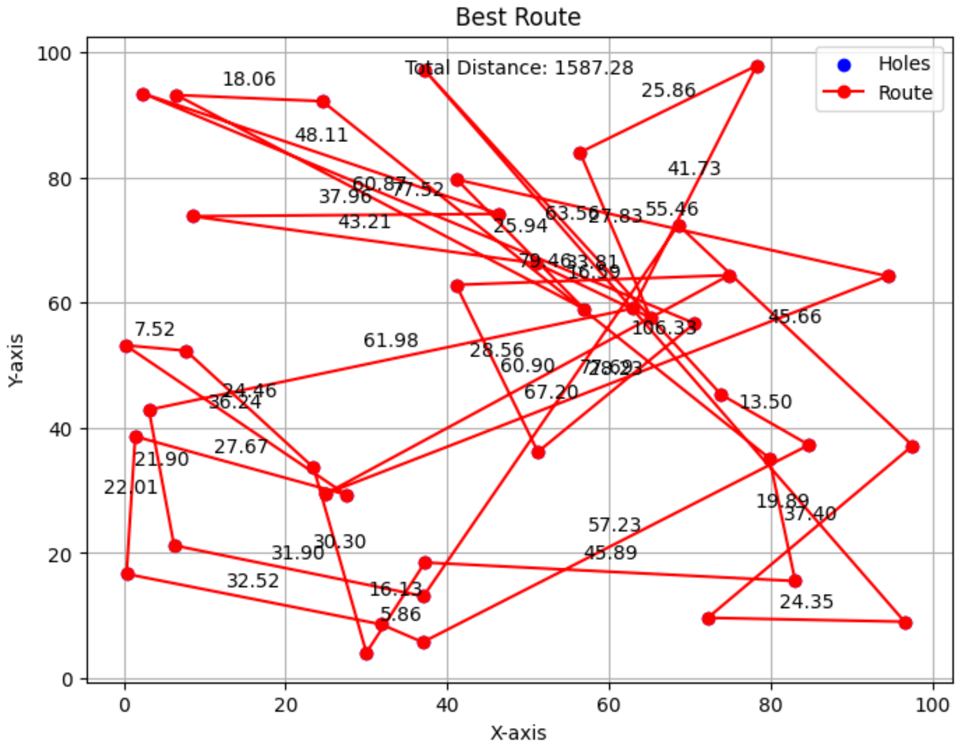


Results

The best route is as follows:

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

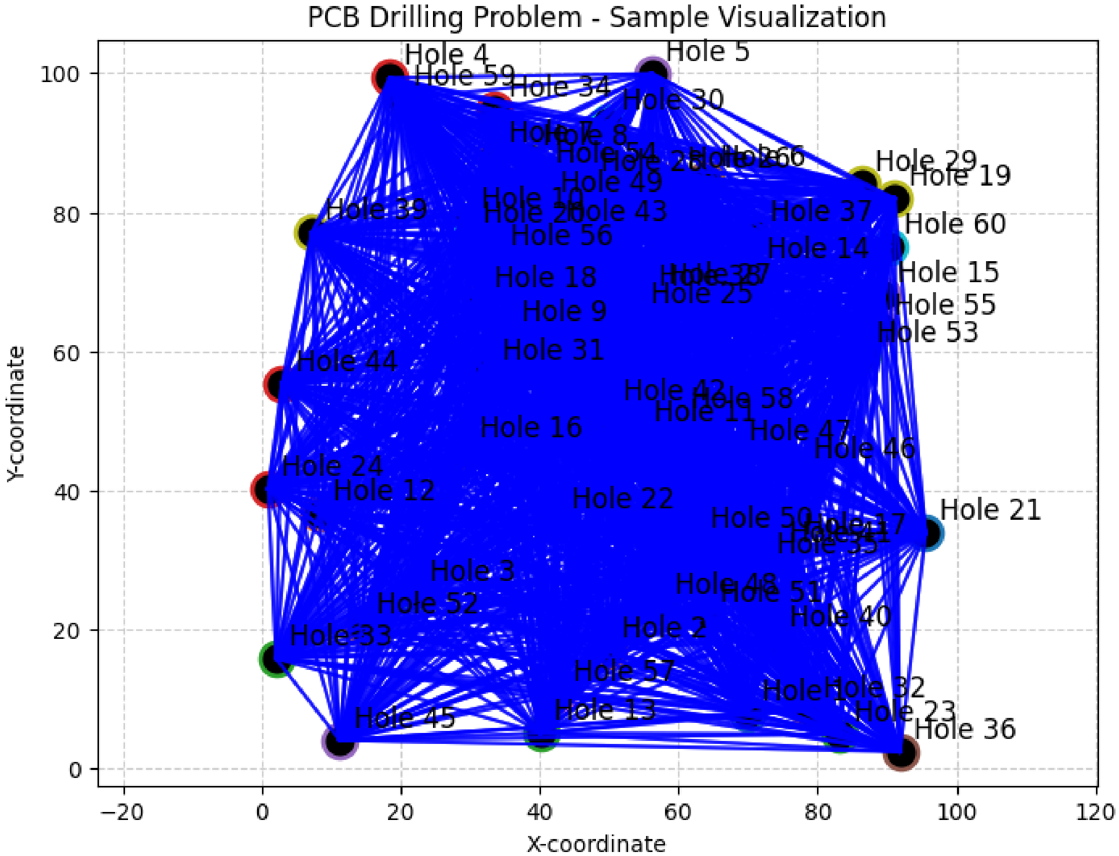
Best Fitness (Total Distance): 1749.5863706198163



1. Large Scale

Number of holes = 60

Number of Particles = 10

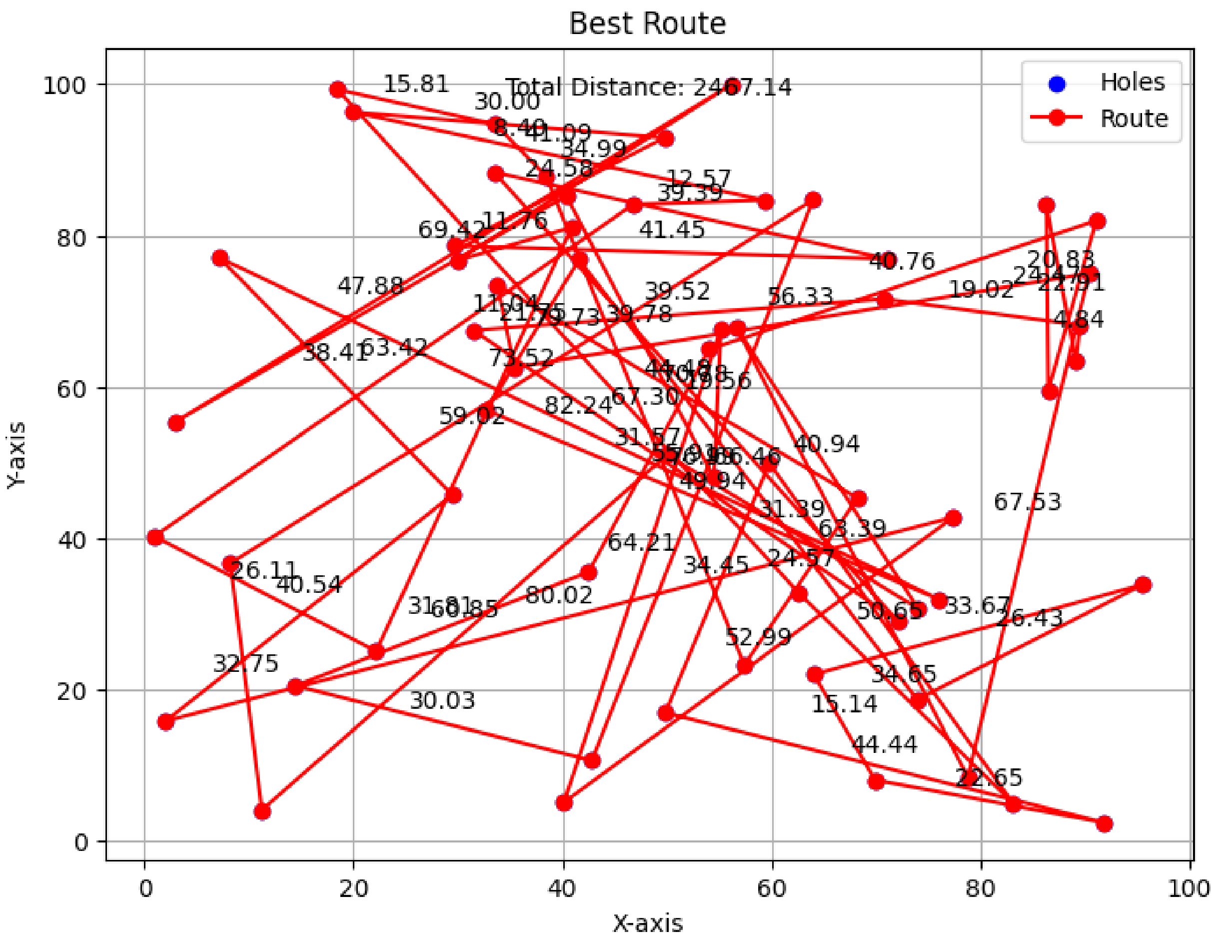


Results

The best route is as follows:

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

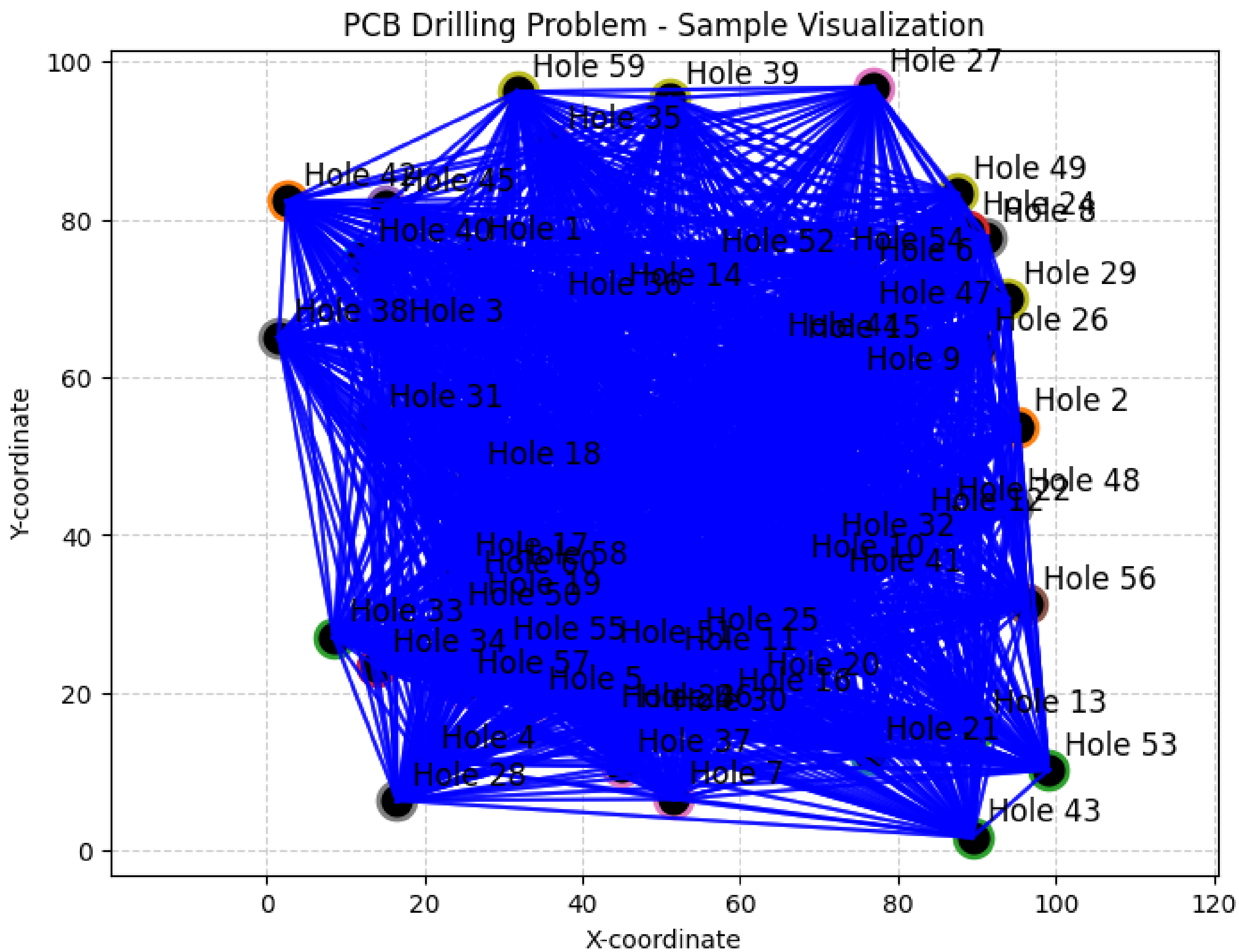
Best Fitness (Total Distance): 2467.13560485674



11. Large Scale

Number of holes = 60

Number of Particles = 20

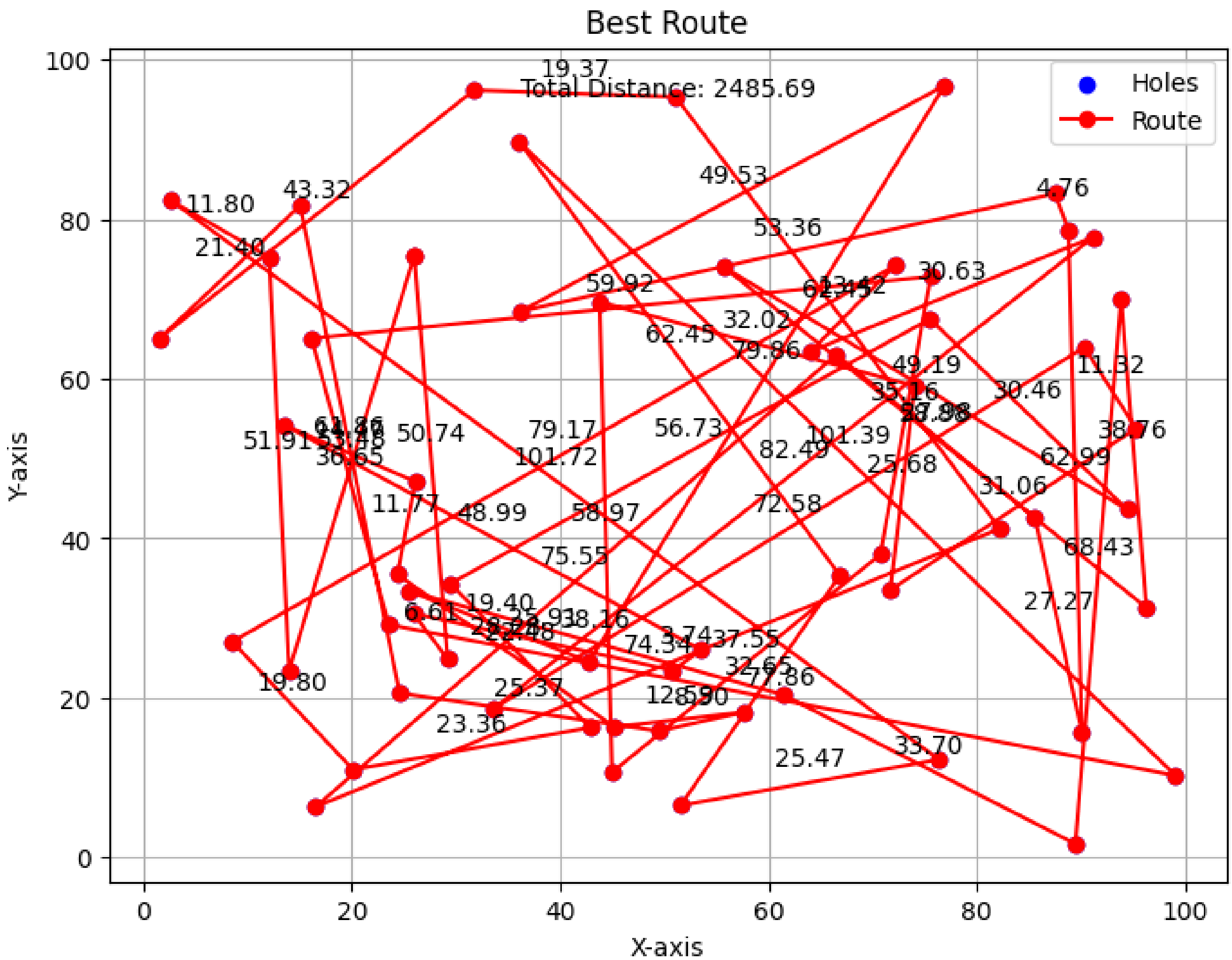


Results

The best route is as follows:

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

Best Fitness (Total Distance): 2485.6869252121974



Conclusion

In this project, we utilized the Particle Swarm Optimization (PSO) algorithm to tackle the PCB Drilling problem, a specific case of the Traveling Salesman Problem (TSP). The PSO algorithm, inspired by the social behaviors observed in natural swarms, efficiently explores the solution space to determine the shortest path that visits each drilling point exactly once and returns to the starting point.

Our results demonstrate that PSO is a robust and efficient technique for optimizing the drilling sequence in printed circuit boards. By carefully tuning the algorithm's parameters—such as the number of particles, inertia weight, cognitive coefficient, and social coefficient—we achieved a significant reduction in the total drilling path length. This optimization directly translates to reduced manufacturing time and cost, which are critical factors in the PCB production process.

Overall, the PSO algorithm proved to be a viable and effective method for solving the PCB Drilling problem, offering a practical approach to minimizing drilling path lengths and improving manufacturing efficiency. Further research and development can enhance its application, making it an indispensable tool in the PCB manufacturing industry.

REFERENCES:

HTTPS://DRIVE.GOOGLE.COM/FILE/D/1IVV8MJQAWKFJ90R2KQQPRYWRM1HNNO-F/VIEW?USP=SHARING

HTTPS://LINK.SPRINGER.COM/ARTICLE/10.1007/S11831-021-09694-4

HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/2021/10/AN-INTRODUCTION-TO-PARTICLE-SWARM-OPTIMIZATION-ALGORITHM/

HTTPS://WWW.TANDFONLINE.COM/DOI/FULL/10.1080/01969720903068435