



جامعة القاهرة

ECO-ROUTING

Marwan Mohamed	20200512
Diaa Maher	20200271
Ganna Ibrahim	20200124
Sara Yasser	20200774

Supervised By
Dr. Sally Kassem
T.A. Youssef Kamel

CAIRO UNIVERSITY
JULY/2024

ABSTRACT

The management of oil waste is a critical concern in the energy sector, requiring efficient logistical solutions to minimize environmental impact and operational costs. Traditional methods of route planning for waste collection trucks often result in suboptimal paths, leading to increased fuel consumption and greenhouse gas emissions. To address these issues, advanced optimization algorithms are necessary to enhance the efficiency of waste collection routes.

This project presents a Hybrid Particle Swarm Optimization and Simulated Annealing (HPSOSA) algorithm designed to solve the Multiple Traveling Salesman Problem (MTSP) for oil waste management. The HPSOSA algorithm combines the global search capabilities of Particle Swarm Optimization (PSO) with the local search proficiency of Simulated Annealing (SA) to optimize the routes for multiple waste collection trucks. The objective function, focused on minimizing the summation of distances traveled by the fleet, was implemented and tested using a dataset of geographical coordinates. The algorithm was developed and executed in PyCharm, ensuring optimal performance and accuracy in solving the MTSP. The optimized routes were then exported dynamically to CSV files, each corresponding to a specific driver, and imported into a Flutter-based mobile application built in Android Studio for real-time route tracking.

The system was rigorously tested using various datasets from CVRPLIB to validate the efficiency and robustness of the HPSOSA algorithm. The mobile application effectively displayed the optimized routes on a map, assigning unique IDs to each driver for clarity. This comprehensive solution not only reduced the total distance traveled by the fleet but also contributed to significant fuel savings and lower emissions. The combination of PyCharm for algorithm development and Android Studio for mobile app deployment provided a seamless integration of optimization and real-time tracking, demonstrating the potential of HPSOSA in revolutionizing oil waste management logistics.

DECLARATION

We hereby declare that our dissertation is entirely our work and genuine / original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F (FAIL) grade, and it may result in withdrawal of our Bachelor's degree.

Group members:

Name

Signature

Marwan Mohamed

Diaa Maher

Ganna Ibrahim

Sara Yasser

PLAIGRISM CERTIFICATE

This is to certify that the project entitled “Eco Routing which is being submitted here with for the award of the “**Bachelor of Computer and Artificial Intelligence Degree**” in “**Operations Research and Decision Support**”. This is the result of the original work by **Marwan Mohamed, Diaa Maher, Ganna Ibrahim** and **Sara Yasser**, under my supervision and guidance. The work embodied in this project has not been done earlier for the basis of award of any degree or compatible certificate or similar title of this for any other diploma/examining body or university to the best of my knowledge and belief.

Turnitin Originality Report

Processed on 08-July-2024

00:14 PKT ID: 300502964

Word Count: 9300

Similarity Index

10%

Similarity by Source

Internet Sources: 06%

Publications: 0 %

Student Papers: 08%

Date: 08/07/2024

Dr. Sally Kassem

TABLE OF CONTENTS

Contents

CHAPTER 1	10
INTRODUCTION	10
1.1 Problem Domain	10
1.2 Problem Statement	11
1.3 Proposed System	11
1.3.1 Aims and Objectives	11
1.3.2 Proposed System Features	12
1.4 Development Methodology	12
1.5 Resource Requirements	12
1.6 Report Layout	13
CHAPTER 2	14
BACKGROUND/ EXISTING WORK.....	14
2.1 Theoretical Foundations.....	14
2.2 Literature Review.....	15
2.3 Existing work.....	17
2.4 Applications in Oil Waste Management	18
2.5 Challenges and Limitations.....	18
2.6 Innovations and Future Directions	19
CHAPTER 3	21
OPTIMIZATION MODEL DESIGN AND IMPLEMENTATION	21
3.1 Problem Formulation	21
3.1.1 Type of Problem.....	21
3.1.2 Objective Function.....	22
3.1.3 Constraints	22
3.1.4 Variables	23
3.1.5 Representation.....	23
3.2 Particle Swarm Optimization (PSO)	24
3.2.1 Initialization	24
3.2.2 Velocity and Position Update	24
3.2.3 Fitness Evaluation.....	25
3.2.4 Convergence Criteria	25
3.2.5 Advantages and Limitations of PSO	26
3.3 Simulated Annealing (SA)	27

3.3.1 Initialization:	27
3.3.2 Iteration Process:	27
3.3.3 Temperature Cooling:	27
3.3.4 Stopping Criteria:	27
3.3.5 Advantages and Limitations of SA:	28
3.4 HPSOSA	29
3.4.1 Algorithm Design	29
3.5 Case Study: Oil Waste Collection	31
3.5.1 Introduction	31
3.5.2 Environmental Impact	32
3.5.3 Current Practices	32
3.5.4 Challenges	32
CHAPTER 4	33
TESTING AND RESULTS	33
4.1 Performance Metrics	33
4.3 Experimental Design	35
4.3.1 Datasets	35
4.3.2 Algorithm Configuration	35
4.3.3 Experimental Environment	36
4.3.4 Experimental Procedure	36
4.3.5 Data Collection	37
4.4 Baseline Comparisons	38
4.5 Sensitivity Analysis	40
4.5.1 Reasons for Performing Sensitivity Analysis:	40
4.5.2 Steps to Perform Sensitivity Analysis:	40
4.5.3 Benefits of Sensitivity Analysis:	44
4.6 Case Study Results	44
4.7 Mobile Application:	46
4.1 Introduction	47
4.2 Objectives	47
4.3 Features	47
CHAPTER 5	51
ANALYTICAL DISCUSSION AND STRATEGIC INSIGHTS	51
5.1 Competitive Profile Matrix (CPM):	51
5.2 SWOT Analysis:	53
	53

.....	53
5.3. Strategic Positioning and Recommendation	53
CHAPTER 6	54
CONCLUSION AND FUTURE WORK	54
6.1 Conclusion	54
6.2 Future Work	54
6.3 Final Remarks	55
REFERENCES	56

LIST OF FIGURES

Table 1	29
Figure 1	37
Figure 2	37
Figure 3	38
Figure 4	41
Figure 5	42
Figure 6	43
Figure 7	43
Figure 8	44
Figure 9	45
Figure 10	45
Figure 11	46
Figure 12	48
Figure 13	51
Figure 14	52
Figure 15	53
.....	30

LIST OF TABLES

Table 1 29

Table 2 30

Table 3 35

Table 4 37

Table 5 38

Table 6 39

Table 7 39

Table 8 39

Table 9 41

Table 10 42

Table 11 45

CHAPTER 1

INTRODUCTION

Effective management of oil waste is a critical concern in the contemporary industrial landscape. The transportation and disposal of oil waste involves complex logistical challenges that require optimized routing solutions to enhance efficiency and reduce operational costs. This project addresses these challenges by developing an advanced route optimization system using a hybrid approach that combines Particle Swarm Optimization (PSO) and Simulated Annealing (SA) to solve the Multiple Traveling Salesman Problem (MTSP). This chapter provides an overview of the project, its significance, and its relevance to the field of operations research and oil waste management.

1.1 Problem Domain

The problem domain for this project lies at the intersection of logistics, waste management, and operations research. Oil waste, generated from various industrial processes, needs to be collected and transported to disposal or recycling facilities. This process must be carried out efficiently to minimize environmental impact, reduce costs, and comply with regulatory requirements.

Traditional methods of route planning and optimization often fall short in addressing the complexities of real-world scenarios involving multiple vehicles and varying waste collection points.

In operations research, the MTSP is a well-known combinatorial optimization problem that extends the classic Traveling Salesman Problem (TSP) to scenarios involving multiple salesmen (or vehicles) starting from a common depot and visiting a set of locations. Solving MTSP is crucial for optimizing routes in multi-vehicle contexts, such as oil waste collection, where multiple trucks need to cover different routes without overlap, ensuring that all waste collection points are visited in the most efficient manner.

1.2 Problem Statement

The primary problem addressed by this project is the optimization of oil waste collection routes in a multi-vehicle environment. The specific challenges include:

1. Efficiently assigning collection points to multiple trucks to minimize the total distance traveled.
2. Ensuring that all collection points are covered without overlap.
3. Adapting the route optimization process to real-world constraints such as varying collection times, vehicle capacities, and traffic conditions.

Current methods often lack the flexibility and computational efficiency required to handle these challenges effectively. This project proposes a novel hybrid approach combining PSO and SA to develop a robust and scalable solution for the MTSP in the context of oil waste management.

1.3 Proposed System

The proposed system aims to address the outlined problem by developing an advanced optimization algorithm that leverages the complementary strengths of Particle Swarm Optimization (PSO) and Simulated Annealing (SA). These algorithms are chosen for their ability to handle complex optimization problems effectively

1.3.1 Aims and Objectives

The primary aims and objectives of the project are as follows:

- To develop a hybrid optimization algorithm that effectively solves the MTSP for oil waste collection.
- To validate the effectiveness of the proposed system through testing and performance evaluation.
- To implement this algorithm in a software application that provides optimized routing solutions.

1.3.2 Proposed System Features

The key features of the proposed system include:

- An algorithm that combines PSO and SA to optimize routes for multiple vehicles.
- Exporting optimized routes to CSV files for integration with other systems.
- A user-friendly interface for inputting collection points, vehicle capacities, and other relevant parameters.

1.4 Development Methodology

The development of the proposed system will follow an iterative and incremental methodology. The key phases include:

- **Algorithm Design:** Developing the hybrid PSO-SA algorithm, including mathematical modeling and simulation.
- **Implementation:** Coding the algorithm and integrating it with a user interface using Python for the backend and Flutter for the mobile application.
- **Testing and Validation:** Conducting extensive testing to ensure the accuracy and efficiency of the algorithm, followed by performance evaluations against benchmark datasets.

1.5 Resource Requirements

The successful execution of this project requires the following resources:

- **Software:** Python (for algorithm development), PyCharm (IDE for coding and testing), Flutter and Android Studio (for mobile application development), CSV libraries (for data handling) Matplotlib (for data visualization).
- **Hardware:** High-performance computing systems for algorithm development and testing.
- **Human Resources:** Expertise in operations research, software development, mobile application development, and project management.

1.6 Report Layout

The report is structured as follows:

- **Chapter 1: Introduction:** Provides an overview of the project, problem domain, problem statement, proposed system, development methodology, resource requirements, and report layout.
- **Chapter 2: Background and Existing Work:** Reviews relevant literature, existing solutions, and theoretical foundations related to the MTSP and optimization algorithms.
- **Chapter 3: Optimization Model Design and Implementation:** Details the design and development of the hybrid PSO-SA algorithm, including mathematical models and implementation specifics.
- **Chapter 4: Validation and Testing:** Discusses the testing methodology, validation processes, results, performance comparisons, and analysis.
- **Chapter 5: Analytical Discussion and Strategic Insights:** Provides a comprehensive discussion of the project's findings, challenges encountered, lessons learned, and strategic insights.
- **Chapter 6: Conclusion and Future Work:** Summarizes the project's contributions, conclusions, and potential directions for future research and development.

CHAPTER 2

BACKGROUND/ EXISTING WORK

In the domain of operations research and logistics, the optimization of routing problems has been a focal point of research and application. This chapter provides a comprehensive review of the theoretical foundations, existing methodologies, and relevant literature pertaining to the Multiple Traveling Salesman Problem (MTSP) and its applications in various industries, particularly in the context of oil waste management.

The optimization of routes for multiple vehicles, known as the MTSP, extends the classic Traveling Salesman Problem (TSP) to scenarios where multiple salesmen (vehicles) must start from a common depot, visit a set of locations, and return to the depot, each exactly once. This combinatorial optimization problem is critical in industries where efficient routing is essential for minimizing costs, reducing environmental impact, and maximizing resource utilization.

2.1 Theoretical Foundations

This section delves into the mathematical foundations underlying the MTSP and its variants. It explores the complexities involved in modeling route optimization problems, including graph theory, combinatorial optimization techniques, and heuristic algorithms commonly used to solve NP-hard problems like the MTSP.

- 1- Problem Formulation:** The MTSP involves finding optimal routes for a fleet of m salespersons (or vehicles) to visit a set of n cities (or locations), where each salesperson starts and ends their route at a designated depot. The objective is to minimize total travel distance, time, or another cost metric while satisfying constraints such as visitation precedence, time windows, and vehicle capacities.
- 2- Graph Theory Representation:** The graph representation of Cities and depots are represented as nodes in a complete graph, where each edge between nodes represents a

possible direct route between cities. The Cost Matrix: A cost matrix where represents the cost distance and time of traveling directly from city to another city.

- 3- **Combinatorial Optimization Techniques:** Exact methods such as Integer Linear Programming (ILP) formulations can provide optimal solutions for small instances of MTSP. These formulations typically include variables to denote the sequence of visits and decision about which salesperson visits each city. Heuristic Approaches due to the NP-hard nature of MTSP and its variants, heuristic methods are commonly employed:
 - a. **Nearest Neighbor:** Each salesperson starts from a depot and selects the nearest unvisited city until all cities are visited.
 - b. **Genetic Algorithms:** Population-based metaheuristics that iteratively evolve solutions through crossover and mutation operations.
 - c. **Ant Colony Optimization:** Inspired by the foraging behavior of ants, where pheromone trails guide the construction of solutions.
 - d. **Simulated Annealing:** A probabilistic technique that allows for exploration of suboptimal solutions to escape local optima.
- 4- **Vehicle Routing Problem (VRP) Variants:** Time-Dependent VRP, Capacitated VRP, Stochastic VRP and Split Delivery VRP
- 5- **Complexities and Trade-offs:** NP-hard nature of MTSP variants implies that exact solutions may be impractical for large instances, necessitating heuristic or metaheuristic approaches. The Balance that needed for high-quality solutions with feasible computational requirements is critical, particularly in real-time or dynamic environments. Choosing and adapting algorithms based on problem characteristics and available computational resources are essential for practical implementation.

2.2 Literature Review

Reference [1], we gained valuable insights from the study on hybrid particle swarm optimization with genetic algorithms (HPSOGA) for solving capacitated vehicle routing problems with fuzzy demand (CVRPFD). This approach combines PSO's local and global search capabilities with

GA's crossover and mutation operations, ensuring feasible solutions through modified particle coding. The study demonstrated HPSOGA's effectiveness using modified CVRP datasets and practical application to a garbage collection system in Indonesia. Results showed HPSOGA outperformed standalone PSO and GA in terms of solution quality and efficiency. This reference significantly influenced our methodology, providing a robust framework for our project.

We utilized various datasets from the Vehicle Routing Problem (VRP) repository provided by PUC-Rio to test our algorithm. This extensive collection of benchmark instances is widely recognized in the research community and provided a solid foundation for evaluating the performance and robustness of our approach. The diverse and well-documented datasets from this repository were instrumental in ensuring the validity and reliability of our results. [2]

In reference [3], we drew valuable insights from a study on biologically inspired algorithms for the Multiple Travelling Salesman Problem (MTSP). The research introduced two PSO-based algorithms, APSO and HAPSO, to solve the MTSP, which involves finding minimal cost tours for multiple salesmen starting and ending at a depot. APSO combines PSO with 2-opt, path-relink, and swap operators, while HAPSO integrates GRASP, PSO, and similar operators. Experimental comparisons using five TSP instances showed that HAPSO outperforms both APSO and traditional algorithms like GA and ACO in terms of performance and stability. This study's findings significantly influenced our approach, offering a robust and effective methodology for our project.

The Capacitated Vehicle Routing Problem (CVRP) involves optimizing the delivery routes of a fleet of vehicles from a common depot to meet customer demands at minimum transit cost while respecting vehicle capacity constraints. Reference [4] presents a meta-heuristic approach using simulated annealing to solve the CVRP. This algorithm combines random and deterministic operators informed by problem-specific knowledge. Experimental results demonstrate the effectiveness of this approach, providing favorable comparisons to other methods. The insights from this study significantly contributed to the development and validation of our model.

To address the MTSP-CVRP in oil waste management, our objective is to minimize the total travel distance for multiple vehicles, ensuring each location is visited exactly once by one vehicle, and adhering to capacity constraints. The model's objective function calculates the total travel distance based on the routes taken by the vehicles. Key constraints include the Single Visit Constraint, ensuring each location is visited only once; the Flow Conservation Constraint, ensuring each vehicle leaves and arrives at each location exactly once; the Depot Constraint, requiring each vehicle to start and end its route at a common depot; and the Capacity Constraint, limiting the total demand serviced by each vehicle to not exceed its capacity. These equations and constraints, derived from references [5][6], form the foundation of our optimization model, ensuring an efficient and feasible solution for the problem at hand.

2.3 Existing work

Example Application: Hybrid Particle Swarm Optimization and Simulated Annealing (HPSOSA)

- **An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse:** This project discusses a hybrid algorithm combining Particle Swarm Optimization (PSO) and Simulated Annealing (SA) to improve the efficiency of path planning for Automated Guided Vehicles (AGVs) in warehouses. PSO is used in determining each particle represents a potential solution exploring the search space. The SA Integration adds a mechanism to help particles escape local optima and converge to global optima. [9]
- **Hybrid PSO – SA Algorithm for Achieving Partitioning Optimization in Various Network Applications:** This study introduces a hybrid algorithm combining Particle Swarm Optimization (PSO) and Simulated Annealing (SA) for optimizing partitioning in network applications. The goal is to minimize both cut size and circuit delay. PSO combines local and global search strategies, offering high search efficiency. SA avoids local optima through a probability-based search process controlled by a cooling schedule. Experimental results indicate that the hybrid PSO-SA algorithm outperforms other algorithms in achieving better partitioning results. [8]

- **Hybrid PSO-SA Type Algorithms for Multimodal Function Optimization and Reducing Energy Consumption in Embedded Systems:** This paper presents a novel hybrid evolutionary algorithm that combines Particle Swarm Optimization (PSO) and Simulated Annealing (SA). The hybrid algorithm, named HPSO-SA, addresses PSO's premature convergence by incorporating SA's local search capabilities. This combination ensures fast convergence through PSO while SA helps escape local optima. [7]

2.4 Applications in Oil Waste Management

This section specifically examines how optimization techniques, including the MTSP, are applied in the field of oil waste management. It reviews case studies, industry practices, and challenges faced in optimizing waste collection routes, considering factors such as varying collection schedules, vehicle capacities, environmental regulations, and operational constraints.

- **Oil & Gas Industry Waste Management:** The paper addresses the significant challenge in the Oil & Gas (O&G) sector concerning waste streams that can become contaminated by oily or hazardous fluids and radioactivity. This requires careful handling, treatment, and disposal. Common wastes generated at terminals include tank bottom sludge, spill cleanup materials, and soils contaminated with oil. The tank sludge typically consists of water, residual product, and various solids like sand, scale, and rust. Effective management involves either re-processing for product recovery or disposal at licensed facilities.

-

2.5 Challenges and Limitations

An analysis of the challenges and limitations associated with existing approaches to solving routing problems in logistics and waste management contexts. This section explores issues such as scalability, computational complexity, and the trade-offs between solution quality and computational resources.

- **Scalability:** Complexity Increase with Size as the size of the problem as the number of locations, trucks, and constraints increases, the complexity of finding an optimal solution grows exponentially. Talking about the Computational Resources leads we to larger problems require more memory and processing power.
- **Computational Complexity:** Due to Non-Linear Hard Nature, many routing problems are NP-hard, meaning no known polynomial-time algorithms can solve all instances optimally.
- **Trade-offs Between Solution Quality and Computational Resources:** Achieving an optimal solution often requires significant computational time and resources. In many real-world applications, near-optimal solutions obtained quickly are preferred over optimal solutions that take a long time to compute. Moreover, Heuristics and metaheuristics need careful tuning of parameters, which can be time-consuming and require expertise.
- **Data Quality and Availability:** Accurate Data is needed. High-quality, accurate data on locations, distances, and times are crucial for effective routing. Inaccurate or incomplete data can lead to suboptimal solutions. While Integration with Systems, Routing solutions need to integrate with other systems. Ensuring seamless integration can be challenging and requires robust data handling and interoperability.

2.6 Innovations and Future Directions

Lastly, this section outlines potential innovations and future research directions in the optimization of routing problems. It discusses emerging trends, technological advancements, and opportunities for enhancing the efficiency and effectiveness of routing algorithms in complex operational environments.

- **Artificial Intelligence and Machine Learning:** Utilizing machine learning models to predict traffic patterns, demand fluctuations, and potential disruptions can improve routing decisions. These models can provide real-time insights and anticipate future conditions. The deep learning techniques can handle complex data structures and identify

patterns that traditional methods might miss, improving the accuracy of route optimization.

- **Advanced Heuristic and Metaheuristic Approaches:** Combining different heuristic and metaheuristic approaches as Particle Swarm Optimization with Simulated Annealing can leverage the strengths of each method, leading to better solutions. Developing algorithms that adapt their parameters based on problem characteristics and real-time feedback can enhance performance and robustness.
- **Autonomous Vehicles and Drones:** Algorithms specifically designed for autonomous vehicles and drones can optimize routes without human intervention. This includes handling complex environments and dynamically adjusting routes based on real-time data. Coordinating multiple autonomous agents as vehicles and drones to work together efficiently requires sophisticated algorithms capable of handling decentralized decision-making.

CHAPTER 3

OPTIMIZATION MODEL DESIGN AND IMPLEMENTATION

Provide an overview of Chapter 3, outlining its importance as the core technical section where the optimization model, based on Hybrid Particle Swarm Optimization and Simulated Annealing (HPSOSA), is designed, implemented, and validated.

3.1 Problem Formulation

The Hybrid Particle Swarm Optimization and Simulated Annealing (HPSOSA) project addresses a complex optimization problem known as the Multiple Traveling Salesman Problem (MTSP) in the context of oil waste management. The MTSP is a generalization of the classic Traveling Salesman Problem (TSP), where instead of a single salesman, multiple salesmen (or vehicles) must visit a set of locations. Each vehicle starts and ends at a common depot, and each location must be visited exactly once by one of the vehicles. Additionally, the problem incorporates elements of the Capacitated Vehicle Routing Problem (CVRP), which introduces constraints related to the capacity of each vehicle.

3.1.1 Type of Problem

Multiple Traveling Salesman Problem (MTSP): The MTSP involves finding the optimal set of routes for multiple vehicles, ensuring that each vehicle starts and ends at a common depot and that all locations are visited exactly once. The objective is typically to minimize the total travel distance or cost.

Capacitated Vehicle Routing Problem (CVRP): The CVRP adds another layer of complexity by introducing capacity constraints for each vehicle. Each vehicle has a limited capacity, and the total demand of the locations assigned to a vehicle cannot exceed this capacity. This is particularly relevant in the context of oil waste management, where each vehicle has a specific capacity for collecting waste.

The combination of MTSP and CVRP forms a complex optimization problem where the goal is to minimize the total travel distance while adhering to capacity constraints for each vehicle.

3.1.2 Objective Function

The primary objective of the MTSP-CVRP in oil waste management is to minimize the total travel distance for multiple vehicles, ensuring that each location is visited exactly once by one of the vehicles, starting and ending at a common depot, and respecting the capacity constraints of each vehicle.

Objective Function:

$$\text{Minimize } Z = \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot x_{ijk}$$

Where:

- Z is the total travel distance.
- m is the number of vehicles (salesmen).
- n is the number of locations (cities) to be visited.
- d_{ij} is the distance between location i and location j .
- x_{ijk} is a binary variable that equals 1 if vehicle k travels from location i to location j , and 0 otherwise.

3.1.3 Constraints

The optimization model must satisfy the following constraints:

1. **Single Visit Constraint:** Each location must be visited exactly once by exactly one vehicle.

$$\sum_{i=0}^n \sum_{k=1}^m x_{ijk} = 1, \quad \forall j = 1, 2, 3, \dots, n$$

2. **Flow Conservation Constraint:** Each vehicle must leave each location it visits exactly once.

$$\sum_{i=0}^n x_{ijk} = \sum_{i=0}^n x_{ikj} \quad \forall j = 1, 2, 3, \dots, n \text{ and } \forall k = 1, 2, 3, \dots, m$$

3. **Depot Constraint:** Each vehicle must start and end its route at the common depot.

Each vehicle starts its route from the depot:

$$\sum_{j=1}^n x_{0jk} = 1, \quad \forall k = 1, 2, 3, \dots, m$$

Each vehicle starts its route from the depot:

$$\sum_{i=1}^n x_{i0k} = 1, \quad \forall k = 1, 2, 3, \dots, m$$

4. **Capacity Constraint:** The total demand of the locations assigned to each vehicle must not exceed its capacity.

$$\sum_{i=1}^n d_i y_{ik} \leq C, \quad \forall k = 1, 2, 3, \dots, m$$

3.1.4 Variables

- **Binary Decision Variables:**

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ travels from location } i \text{ to location } j \\ 0, & \text{otherwise} \end{cases}$$

- **Continuous Variables:** U_i (position of location i in the tour)

(HERE)

3.1.5 Representation

For the whole system, we implemented permutation representation, where:

- Customers will be represented as the index they are listed within the dataset.
- Each vehicle route is represented as a 1D array of such customer indices, which indicates the order in which the customers will be visited.

- A solution would be a 2D array consisting of the number of 1D arrays equal to the number of vehicles used, each containing the list of customers

3.2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an iterative optimization algorithm inspired by the social behavior of birds flocking or fish schooling. The PSO algorithm works by having a population (swarm) of candidate solutions (particles) that move around in the search space to find the optimal solution. Each particle adjusts its position based on its own experience and the experience of neighboring particles.

3.2.1 Initialization

Typically, in PSO algorithm, the initial population is the only time a population is generated in the algorithm, unlike other algorithms like (GA). So, it is a crucial part of the algorithm and should be decided upon carefully. In most cases a random initial population is the safest bet, but we chose to implement the Greedy algorithm.

Greedy Algorithm: Each vehicle iteratively chooses the location that is closest to it, and hypothetically moves to that location and then repeats the process, until all customers are visited.

This of course disregards the capacity constraint, which is why this is only the initial population which will soon be modified upon by the algorithm to put the capacity constraint into consideration.

The upside though, is that this allows us to be closer to optimality as most of the distances are nearly optimal since each vehicle chooses the closest to it iteratively.

3.2.2 Velocity and Position Update

In traditional PSO, velocity updating of a particle is straightforward, you add the velocity vector (which depends on social and cognitive components) to the particle's current position, and you get a new particle, and see whether a new personal best or new global best was achieved. This could be applied to normal VRP problems, where only one vehicle is present.

However, in our case this velocity update would be impossible due to the permutation representation nature of the problem, this discrete representation contrasts with the continuous nature of traditional PSO, where positions are typically represented as vectors in a continuous space. Which brings us to what we implemented, Swap velocity.

Swap Operations: Instead of updating velocities with continuous values, swap operations are used to modify the sequence of locations in a particle's position. A swap operation involves exchanging two locations in the sequence, effectively altering the permutation.

The velocity of a particle in this context is represented as a series of swap operations that need to be applied to transform the current position towards a target position, which could be personal best or global best.

The position update involves applying a sequence of swaps to the current permutation to move towards a new permutation. The swaps are determined based on 2 probabilities, probability to move towards personal best, and a probability to move towards global best.

Create a swap sequence that combines the influence of the personal best and global best positions.

This sequence is analogous to the velocity in traditional PSO but in terms of swaps.

3.2.3 Fitness Evaluation

The fitness of a particle is the total distance equated from the objective function.

The lesser the distance, the fitter the particle.

3.2.4 Convergence Criteria

In PSO, the number of iterations is a huge factor in the time complexity of the algorithm, but of course a higher number of iterations leads to more close-to-optimal solutions.

But to save computational resources, we implemented a stopping criteria, where if after (n) iterations the solutions doesn't change, the algorithm terminates assuming no further improvement.

3.2.5 Advantages and Limitations of PSO

Advantages of PSO:

1. Simplicity:

PSO is relatively easy to implement and understand compared to other optimization algorithms. Its simple mathematical operations and straightforward structure make it accessible for various applications.

2. Efficiency:

PSO can quickly converge to a good solution with fewer iterations compared to some other optimization methods. This efficiency is beneficial for large-scale problems where computational resources are limited.

3. Flexibility:

PSO can be applied to a wide range of optimization problems, both continuous and discrete. It is versatile and can be easily adapted or hybridized with other algorithms to enhance performance.

Limitations of PSO:

1. Premature Convergence:

PSO can sometimes converge prematurely to a local optimum, especially if the diversity of the swarm is not maintained. This issue can be mitigated by introducing mechanisms to promote exploration.

2. Parameter Sensitivity:

The performance of PSO is highly dependent on the selection of its parameters, such as inertia weight, cognitive coefficient, and social coefficient. Finding the right balance of these parameters can be challenging and may require extensive experimentation.

3.3 Simulated Annealing (SA)

Simulated Annealing (SA) is a probabilistic optimization algorithm inspired by the annealing process in metallurgy, where a material is heated and then slowly cooled to remove defects and find a state of minimum energy. In optimization, SA is used to find a global optimum solution by exploring the solution space and allowing occasional uphill moves to escape local optima.

3.3.1 Initialization:

- Initial Solution: Start with an initial solution, typically chosen at random.
- Initial Temperature: Set an initial temperature T which controls the probability of accepting worse solutions.

3.3.2 Iteration Process:

- Neighborhood Exploration: Generate a neighboring solution by making a small change to the current solution. For the MTSP, this could involve swapping two locations in the tour.
- Acceptance Criterion: Evaluate the new solution using the objective function. If the new solution is better, it is accepted. If it is worse, it is accepted with a probability that decreases as the algorithm progresses.
- The probability is given by: $P(\Delta E) = \exp\left(\frac{-\Delta E}{T}\right)$

-Where, ΔE is the difference in the objective function values between the new and current solutions.

3.3.3 Temperature Cooling:

Gradually reduce the temperature T according to a cooling schedule, typically an exponential decay: $T \leftarrow \alpha T$, where α is the cooling rate, a parameter between 0 and 1.

3.3.4 Stopping Criteria:

The algorithm continues iterating until a stopping criterion is met, such as a minimum temperature, a maximum number of iterations, or a convergence threshold.

3.3.5 Advantages and Limitations of SA:

Advantages:

- Escaping Local Optima: SA's ability to accept worse solutions with a certain probability helps it escape local optima and explore the global solution space more effectively.
- Simplicity and Flexibility: SA is relatively simple to implement and can be applied to a wide range of optimization problems, both continuous and discrete.

Limitations of Simulated Annealing:

- Parameter Sensitivity: The performance of SA is sensitive to the choice of initial temperature, cooling rate, and stopping criteria. Tuning these parameters can be challenging.

3.4 HPSOSA

After discussing the 2 algorithms of interest, we will now discuss the main algorithm we are using for our problem, the Hybrid Particle swarm optimization and Simulated Annealing algorithm.

We will go into the details of the hybridization and discuss the benefit of hybridizing these 2 algorithms.

3.4.1 Algorithm Design

Generally, hybridization of algorithms can be done in many ways:

- Sequentially: running one algorithm and passing the output to the other.
- Parallel: running one algorithm simultaneously with the other algorithm.
- Concurrent: Utilizing the features of both, ex: updating velocity (PSO) and cooling temperature (SA), which can be beneficial for balancing exploration and exploitation.

Sequential vs. Parallel Hybridization: Which is Better for HPSOSA?

Table 1

Aspect	Sequential Hybridization	Parallel Hybridization
Advantages	<ul style="list-style-type: none">- Complementary Strengths: Leverages the strengths of both PSO and SA sequentially.- Control and Simplicity: Easier to manage and control- Resource Efficiency: Focuses computational effort on one algorithm at a time.- Balanced Exploration and Exploitation: Clear division between exploration (PSO) and exploitation (SA).	<ul style="list-style-type: none">- Speed and Efficiency: Runs PSO and SA simultaneously.- Improved Solution Quality: Simultaneous exploration and exploitation can lead to better solutions.- Diversity and Robustness: Maintains higher solution diversity.

Disadvantages	<p>- Time Consumption: More time-consuming due to sequential execution.</p> <p>-Dependence on Initial Solution Quality: Final solution quality depends heavily on the PSO phase's effectiveness.</p>	<p>- Complexity and Resource Intensive: More complex to implement and manage.</p> <p>-Increased Computational Demand: Demands more resources</p>
----------------------	--	--

Specific Considerations for Oil-Waste Collection

Aspect	Sequential Hybridization	Parallel Hybridization
Nature of the Problem	<p>- Complexity and Constraints.</p> <p>- Need for High-Quality Solutions</p>	<p>-Speed and Robustness: If computational resources are available.</p> <p>-Real-Time Adaptability: Beneficial if real-time adaptability is required.</p>
Specific Advantages	<p>- Simplicity and Control: Offers simplicity and control, allowing clear understanding of each phase's impact on the overall solution.</p> <p>- Balanced Approach: Ensures a systematic approach to exploration (PSO) and exploitation (SA).</p>	<p>-Speed and Robustness: Provides faster convergence and potentially better solutions.</p> <p>-Real-Time Adaptability: adjust routes based on dynamic conditions.</p>

Table 2

Conclusion: Which is Better?

For the oil-waste collection problem, **sequential hybridization** of HPSOSA is likely the better choice due to the following reasons:

- **Simplicity and Control:** The ability to clearly manage and understand each phase of the optimization process is crucial for dealing with the complexity of the problem.
- **Resource Efficiency:** Sequential hybridization is less resource-intensive, which is beneficial if computational resources are limited.
- **Balanced Approach:** It ensures a systematic approach to exploration and exploitation, aligning well with the need to first identify promising routes and then refine them.

Knowing this, we had to decide which algorithm should be run first, that would give better results, which in our case was PSO, because:

1. Its proficiency in exploring the solution space and identifying initial promising regions.
2. The sequential approach allows PSO to efficiently lay the groundwork for subsequent fine-tuning by Simulated Annealing (SA).
3. ensuring a methodical improvement of solutions towards optimal routes in the oil-waste collection problem.

3.5 Case Study: Oil Waste Collection

3.5.1 Introduction

Oil waste, generated from various industrial and commercial activities, poses significant environmental challenges worldwide. Managing this waste efficiently is crucial to mitigate environmental impact and comply with regulatory standards.

3.5.2 Environmental Impact

Oil waste can contaminate soil, water sources, and ecosystems if not handled properly. It contains toxic substances that can harm aquatic life and affect biodiversity. The improper disposal of oil waste contributes to soil degradation and can lead to long-term environmental damage.

3.5.3 Current Practices

Traditionally, oil waste collection involves specialized equipment and processes to safely extract, transport, and dispose of waste materials. Companies often rely on trained personnel and regulatory frameworks to ensure compliance with environmental laws.

3.5.4 Challenges

The main challenges in oil waste collection include logistical complexities, regulatory compliance, and the need for advanced technologies to handle diverse types of waste efficiently. Economic factors also play a role, as disposal costs and operational expenses can be significant.

CHAPTER 4

TESTING AND RESULTS

4.1 Performance Metrics

When evaluating the performance of the Hybrid Particle Swarm Optimization with Simulated Annealing (HPSOSA) algorithm, it's crucial to consider a variety of performance metrics. These metrics help to assess the algorithm's effectiveness, efficiency, and robustness in solving the optimization problem. Below are the key performance metrics commonly used:

4.1.1 Objective Function Value

This Measures the quality of the solution by evaluating the objective function (e.g., total travel distance in MTSP-CVRP).

Our Goal: Minimize the objective function value.

4.1.2 Convergence Rate

Indicates how quickly the algorithm converges to the optimal or near-optimal solution. Faster convergence rates are desirable as they imply that the algorithm can find good solutions in less time.

4.1.3. Computational Time

The total time taken by the algorithm to find the optimal solution. Minimize computational time to make the algorithm more practical for real-world applications.

4.1.4. Number of Iterations

The total number of iterations required to reach convergence. Fewer iterations to reach convergence indicates a more efficient algorithm.

4.1.5. Robustness

The ability of the algorithm to consistently find good solutions over multiple runs with different initial conditions. High robustness ensures reliability and reproducibility of results.

4.1.6. Solution Diversity

Measures the diversity of the solutions found by the algorithm across different runs. Maintain a balance between exploration and exploitation (convergence to the best solution).

4.1.7. Exploration vs. Exploitation Balance

Assesses how well the algorithm balances exploration of the search space and exploitation of the best solutions found. Achieve an optimal balance to avoid premature convergence and ensure thorough search.

4.1.8. Scalability

The algorithm's ability to handle increasing problem sizes (e.g., more cities in MTSP). Ensure that the algorithm performs well even as the problem size grows.

4.1.9. Adaptability

The ability of the algorithm to adapt to different types of problems or changes in problem parameters. High adaptability ensures the algorithm's applicability to a wide range of optimization problems.

4.3 Experimental Design

This section outlines the setup and methodology used to evaluate the performance of the Hybrid Particle Swarm Optimization with Simulated Annealing (HPSOSA) algorithm. The design encompasses the datasets used, algorithm configuration, experimental environment, and the procedures followed to conduct the experiments.

4.3.1 Datasets

In the field of VRP, there is a library dedicated to providing datasets with their optimal solutions for different types of VRP problems. In our case. It's a CVRP problem, so (Set A) from the library will be used, as it covers all the performance criteria. Here is a sample out of one of the datasets in the set:

X	Y	DEMAND
40	50	0
25	85	20
22	75	30
22	85	10
20	80	40
20	85	20

Table 3

4.3.2 Algorithm Configuration

Number of trucks: 10

Truck capacity: 100

Particle Swarm Optimization (PSO) Parameters:

- Number of iterations: 1000
- Population size: 50
- Personal best probability: 0.7
- Global best probability: 0.4

Simulated Annealing (SA) Parameters:

- **Initial Temperature (T0):** 10000.
- **Cooling rate:** 0.995.
- **Stopping Criterion:** max iteration (10000)

4.3.3 Experimental Environment

Hardware and Software Environment:

- **Hardware:** Experiments were conducted on a workstation with the following specifications:
 - **Processor:** Intel Core i7-9700K, 3.6 GHz
 - **RAM:** 16 GB
 - **Storage:** 512 GB SSD
- **Software:**
 - **Operating System:** Windows 10
 - **Development Environment:** PyCharm for algorithm implementation and experimentation.
 - **Libraries:** NumPy, SciPy, and Matplotlib for numerical computations and plotting.

4.3.4 Experimental Procedure

We're going to conduct the experiment on a small-scale problem for simplicity, which is set (A-n32-k5), which is 32 locations, 5 vehicles, vehicle capacity = 100.

1. **Initial Solution:** As you recall, we start by running the PSO algorithm, which generates an initial population using Greedy algorithm, and we calculate the total distance:

1530.4741901598134

2. **Algorithm Execution:** After running both algorithms, we get the following Final solution:

922.2600868354622

4.3.5 Data Collection

In this section we will discuss the recorded the objective function values, convergence rates, computational times, and number of iterations for each run and collect data on solution diversity and robustness by running the algorithm multiple times with different initial conditions.

Objective function values:

Run	1	2	3	4	5	6	7	8	9	10
Result	1043.083	1235.166	1023.619	1116.003	1062.711	1067.254	1069.71	1066.8850	1038.50	1103.38

Table 4

- The graph shows significant variability in the objective function values across different runs. This indicates that the performance of the algorithm can vary depending on the initial conditions or the random seed for each run.

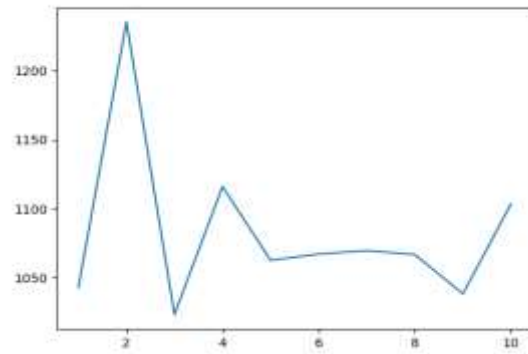


Figure 1

- The significant drop from the initial cost to the cost after PSO shows that PSO is effective in finding a better solution than the initial one.
- The further reduction in cost after applying both SA and PSO suggests that combining these two algorithms results in a more optimized solution than using PSO alone.

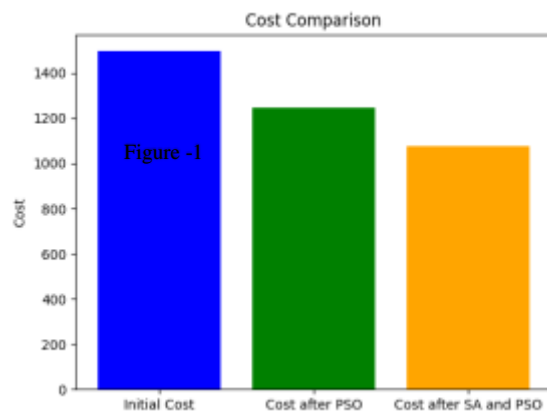


Figure 2

Mean = 1082.631
Standard Deviation = 27

These values indicate the consistency of the algorithm

Final solution visualized:

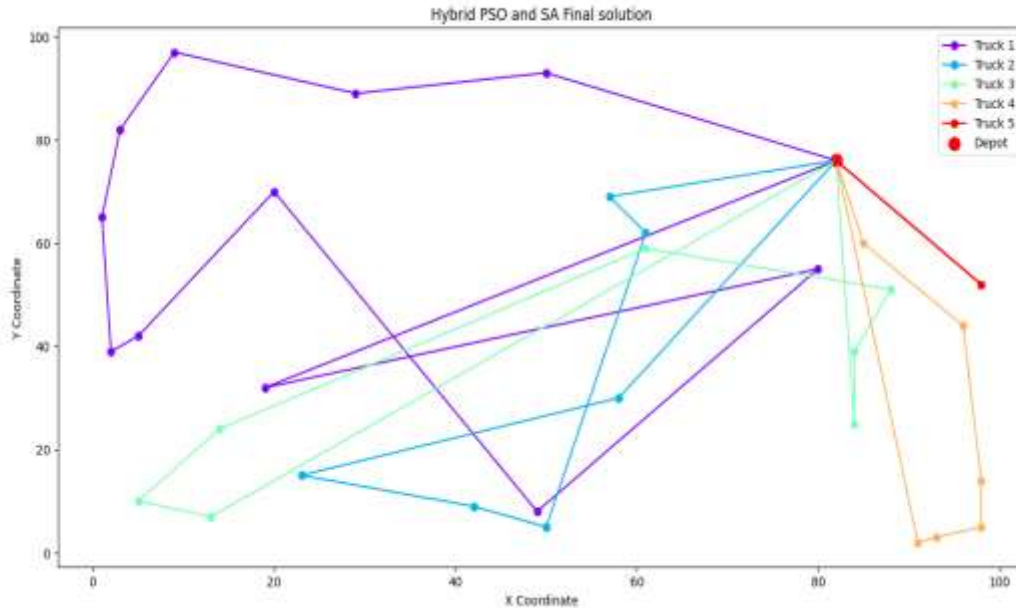


Figure 3

4.4 Baseline Comparisons

The benefit of using CVRPLIB is that each dataset has a benchmark that we can determine how close to true optimality we are.

We will look at the performance of our algorithm in various situations and compare it to the benchmark:

Table 5

HPSOSA	A-n32-k5
922.26	784

HPSOSA	A-n80-k10
1829.477	1763

HPSOSA	M-n200-k17
2346	1275

Of course, it's evident that we aren't optimal, but the solutions reached are close to optimality consistently. But close to optimal is enough to prove algorithm is efficient.

Now, how can we for sure make sure that combining the 2 algorithms is more efficient than just using them standalone? By using the same datasets and comparing the results, and also using the same parameters we used in the previous test:

Set A-n32-k5

Table 6

	Initial Sol.	SA	PSO	HPSOSA
A-n32-k5	1536.289	1203.02	1077.737	922.26

Set A-n80-k10

Table 7

	Initial Sol.	SA	PSO	HPSOSA
A-n80-k10	3575.199	2507.37	2327.089	1829.477

Set M-n200-k17

Table 8

	Initial Sol.	SA	PSO	HPSOSA
M-n200-k17	6201.417	3050.468	2667.36	2346

4.5 Sensitivity Analysis

4.5.1 Reasons for Performing Sensitivity Analysis:

- **Understanding Parameter Influence:** Sensitivity analysis helps in understanding how different parameters affect the performance of your HPSOSA algorithm. Knowing which parameters have the most significant impact can guide you in fine-tuning the algorithm.
- **Improving Robustness:** By analyzing the sensitivity of your algorithm to various parameters, you can make it more robust. This means ensuring that the algorithm performs well across a range of parameter values, rather than being highly sensitive to small changes in any single parameter.
- **Optimizing Performance:** Sensitivity analysis can help identify the optimal values for your parameters. By systematically varying each parameter and observing the outcomes, you can find the best configuration for your specific problem.
- **Model Validation:** It provides a form of validation for your model. If small changes in parameters lead to wildly different outcomes, it may indicate that your model is unstable or overly sensitive, which could be problematic in real-world applications.

4.5.2 Steps to Perform Sensitivity Analysis:

- Select Parameters for Analysis:
 - Initial population size of PSO
 - Number of iterations of PSO
 - Global best probability of PSO
 - Personal best probability of PSO
 - Temperature of SA
 - Cooling rate of SA
 - Maximum iterations of SA

- Define the Range:
 - Initial population size: 10, 20, 30
 - Number of iterations: 50, 100, 150
 - Global best probability: 0.5,0.6,0.7
 - Personal best probability: 0.3,0.4, 0.5
 - Temperature of SA: 500, 5000, 10000, 15000
 - Cooling rate of SA: 0.990,0.995,0.999
 - Maximum iterations of SA: 5000,10000,15000
- **Tornado Diagram:**
- Vary one parameter at a time while keeping others constant. Record the performance of the HPSOSA algorithm for each set of parameter values then subtract the highest from lowest value you get Range to plot in the diagram. **It can be done manually:**

Table 9

Parameter	diff_min	diff_max	min_distance	base_distance	max_distance	difference
p_best_prob	-42	26	1493	1535	1561	68
max_iterations	-6	84	1491	1497	1581	90
pso_population	-110	9	1347	1457	1466	119
initial_temp	-47	125	1389	1436	1561	172
g_best_prob	-171	79	1321	1492	1571	250
cooling_rate	-194	60	1346	1540	1600	254
pso_iterations	-196	128	1289	1485	1613	324

- But that will be impractical and sometimes wrong because of the randomness in the code of the objective function.
- So, we will need to do so many tornado diagrams to find the two variables that always happen to have the biggest ranges in the tornado diagrams so we will need to make too many diagrams.

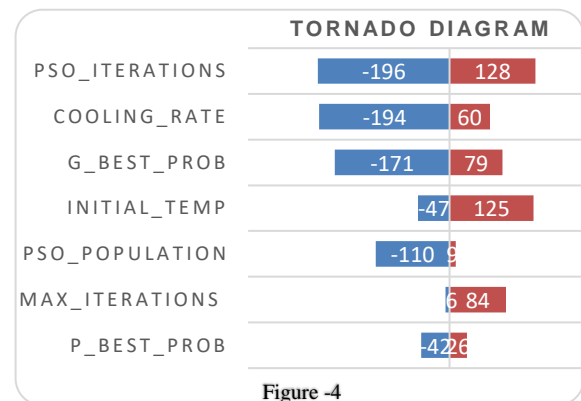


Figure 4

- One way to solve this problem is Automating the process of making tornado diagrams from the code to excel by using “openpyxl” library and the tornado diagrams that got repeated the most was this one:

Table 10

Parameter	diff_min	diff_max	min_distance	base_distance	max_distance	difference
p_best_prob	-45	14	1419	1464	1478	59
max_iterations	-32	69	1595	1627	1696	101
initial_temp	-19	86	1448	1467	1553	105
cooling_rate	-46	134	1536	1582	1716	180
g_best_prob	-12	172	1430	1442	1614	184
pso_iterations	-26	163	1372	1398	1561	189
pso_population	-182	38	1323	1505	1543	220

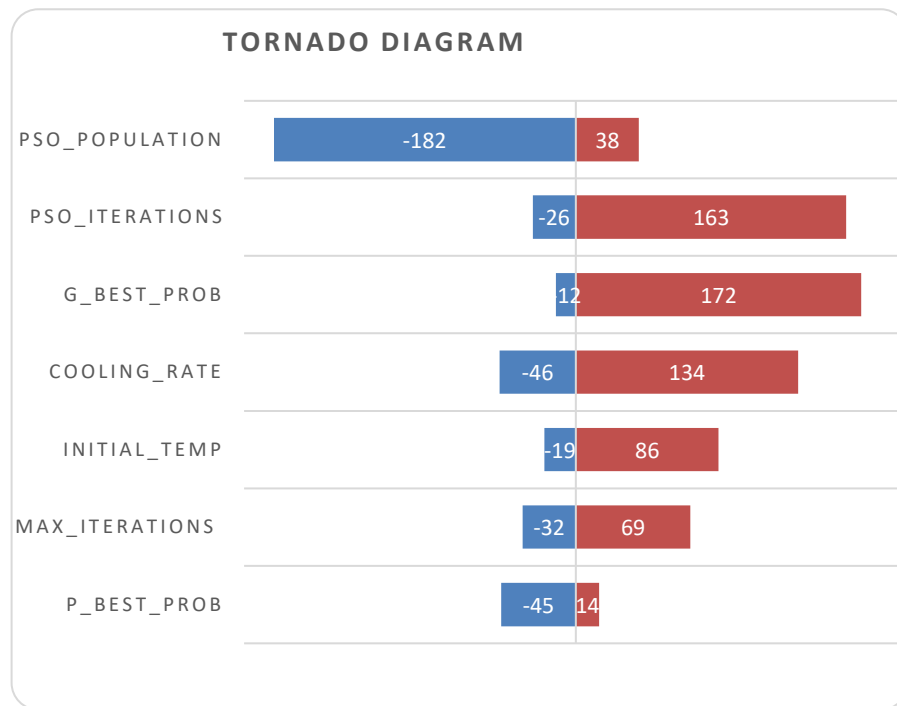


Figure 5

This makes the result way more accurate

- And there is also another way by simply making a code that generates the two variables that affect the output the most and count how many each variable gets generated and two variables with the biggest count is the two variables that affect the output the most we tried it and that was the result:

“With 30 generations”

```
Pso iteration count: 8
Pso population count: 6
Personal best count: 2
Global best count: 8
Initial temperature count: 3
Cooling rate count: 3
Max iteration count: 0
```

“With 60 generations”

```
Pso iteration count: 16
Pso population count: 13
Personal best count: 10
Global best count: 10
Initial temperature count: 2
Cooling rate count: 9
Max iteration count: 0
```

Figure 6

The result indicates the PSO iterations and PSO population size are the variables that affect the output the most.

Heatmap:

By identifying the two most influential variables from the sensitivity analysis results we used them to make the heatmap, by making code that generates heatmap by using “matplotlib” library we get this:

That indicates that the best value for PSO iterations and PSO population size that will get us the minimum distance is 1000 for PSO iterations and 100 for PSO population size.

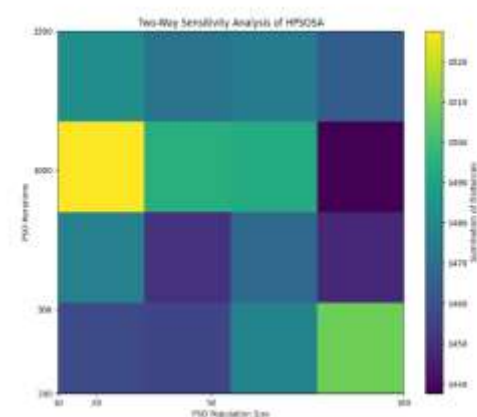


Figure 7

4.5.3 Benefits of Sensitivity Analysis:

- **Enhanced Understanding:** Gain a deeper understanding of the inner workings of your HPSOSA algorithm. Knowing how each parameter influences performance can help in troubleshooting and improving the algorithm.
- **Informed Decision Making:** Sensitivity analysis provides a data-driven basis for choosing parameter values. This leads to more informed and confident decision-making.
- **Algorithm Robustness:** By identifying and mitigating sensitive parameters, you can make your algorithm more robust and reliable. This is particularly important in dynamic or uncertain environments where parameter values may fluctuate.
- **Efficiency Gains:** Optimize the performance of your algorithm by focusing on the most influential parameters. This can lead to significant efficiency gains and better outcomes.

4.6 Case Study Results

Here is the part which matters the most, because it is the purpose of developing this algorithm.

We ran a survey on a small scale, in Rehab city, New Cairo, collecting data about actual customers to see the number of people who would be willing to sign up and donate oil waste:

And more than 60% of were willing to and were already getting ready of oil waste in their own way, though a large proportion was throwing it away which is harmful for the nature.

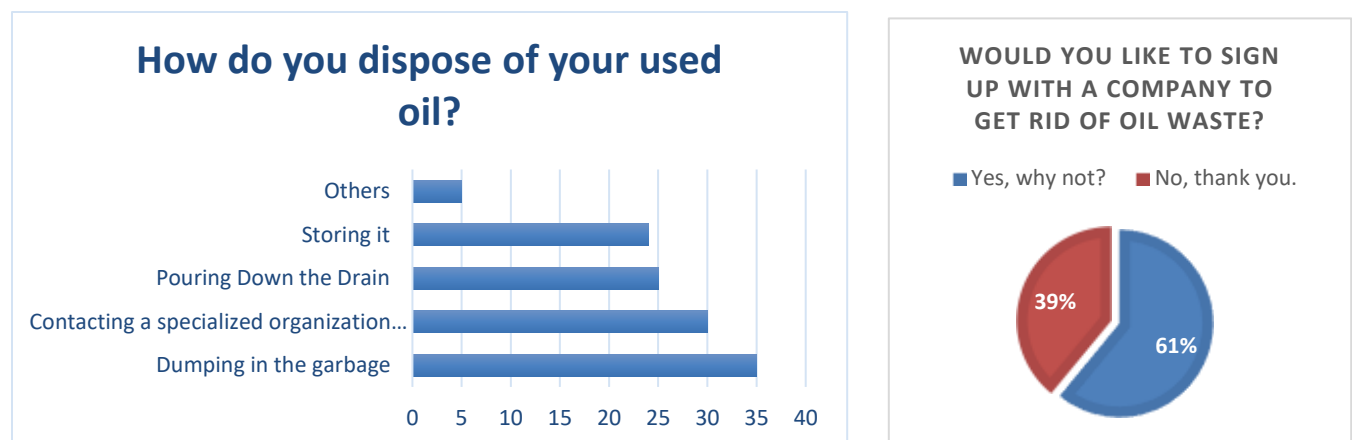


Figure 8

We also asked for the quantity of oil they disposed of on average. This is to determine the average demand and maximum demand to help determine truck capacity:

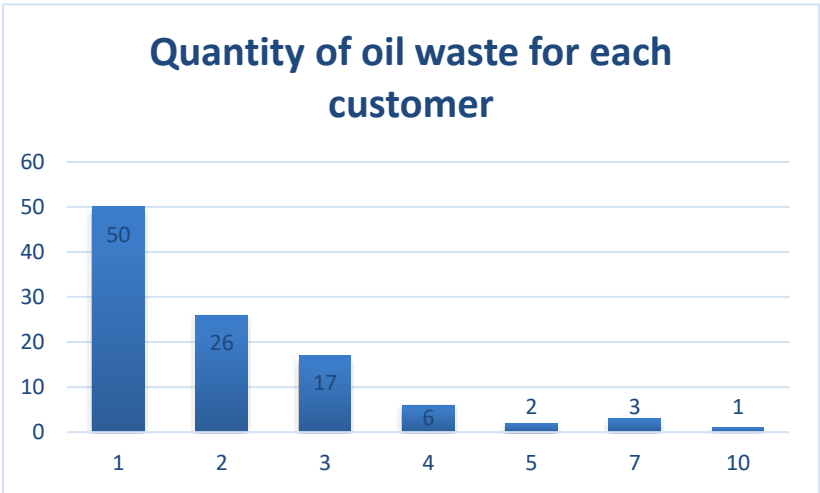


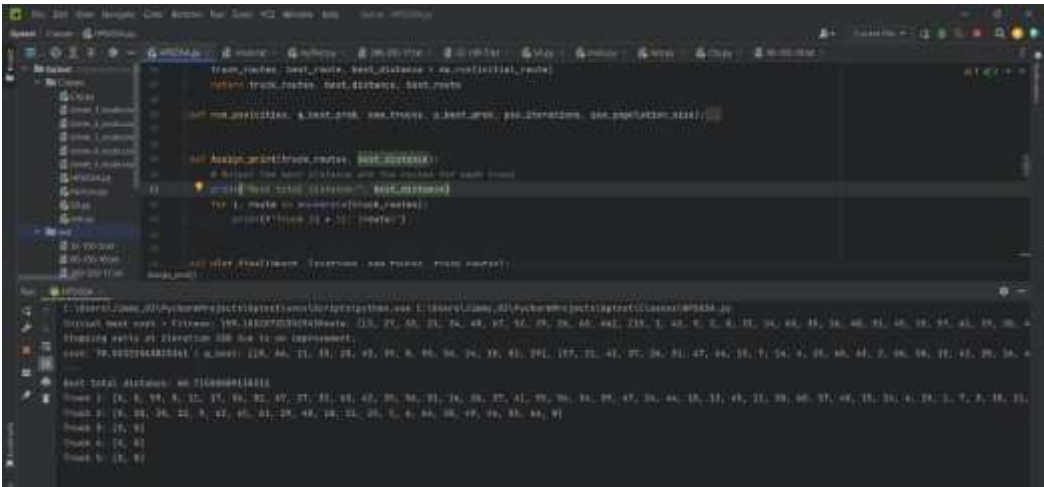
Figure 9

Now, we have real locations of customers, latitudes and longitudes of customers, to run on our algorithm. Here are a few rows from it:

Latitude	Longitude	Demand (Liters)
30.06007	31.49262	2
30.07291	31.50928	1
30.07435	31.48576	1
30.06376	31.49434	2
30.0592	31.51271	1

Table 11

Now let’s run this data on our algorithm and check the results:



- As observed, we start with an initial solution of 159.1 km.
- Then after the PSO is run the solution is improved to be 70.5 km.
- And we end up with a total cost (distance) of 60.75 km.

The graph clearly illustrates the effectiveness of the HPSOSA algorithm in reducing the cost through each optimization stage, showcasing the substantial improvements from the initial cost to the final optimized solution.

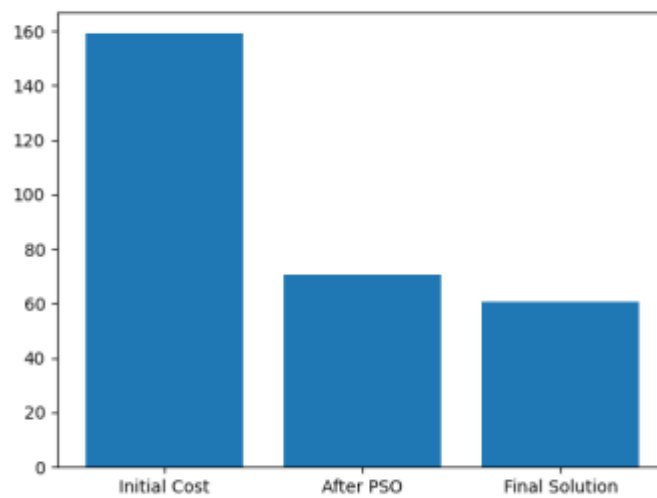


Figure 11

4.7 Mobile Application:

4.1 Introduction

The mobile application developed as part of this project aims to streamline the oil waste collection process by providing truck drivers with optimized routes, real-time updates, and an intuitive interface to manage their tasks. The app leverages the Hybrid Particle Swarm Optimization and Simulated Annealing (HPSOSA) algorithm to solve the Multiple Traveling Salesman Problem (MTSP), ensuring efficient and effective route planning. Additionally, the app includes a customer interface that allows customers to request deliveries and manage loyalty points.

4.2 Objectives

The primary objective of the mobile application is to provide truck drivers with optimized routes for oil waste collection and offer a platform for customers to request deliveries and manage loyalty points.

4.3 Features

The mobile application includes the following features:

Driver Features:

1. User Authentication:

- Secure login and authentication for drivers and administrators.

2. Route Optimization:

- Displays optimized routes calculated using the HPSOSA algorithm.
- Allows drivers to view and follow their assigned routes.

3. Interactive Maps:

- Integrates with mapping services to display routes and collection points.
- Allows drivers to view detailed maps and navigate to collection points.

4. Notifications:

- Sends notifications to drivers about route changes, new assignments, and important updates.

5. Data Management:

- Stores route data, driver performance, and collection metrics for analysis and reporting.

Customer Features:

1. **First determine the role of user screen:**
2. **User Authentication**
3. **Request Deliveries:** Customers can request oil waste collection services through the app.
4. **Loyalty Points:**
5. Customers earn loyalty points for each service request.
 - Points can be redeemed for rewards within the app.

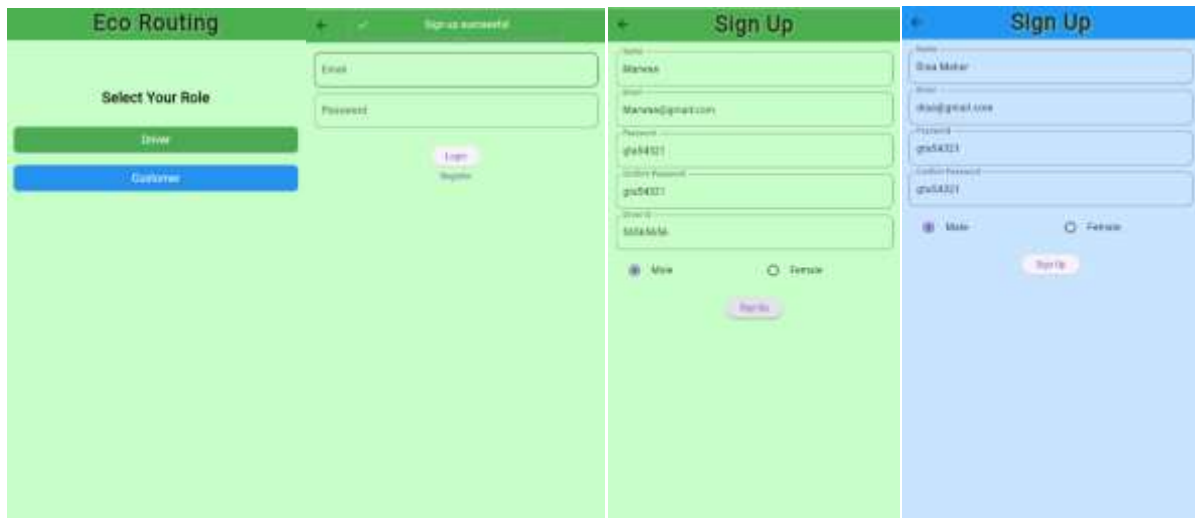
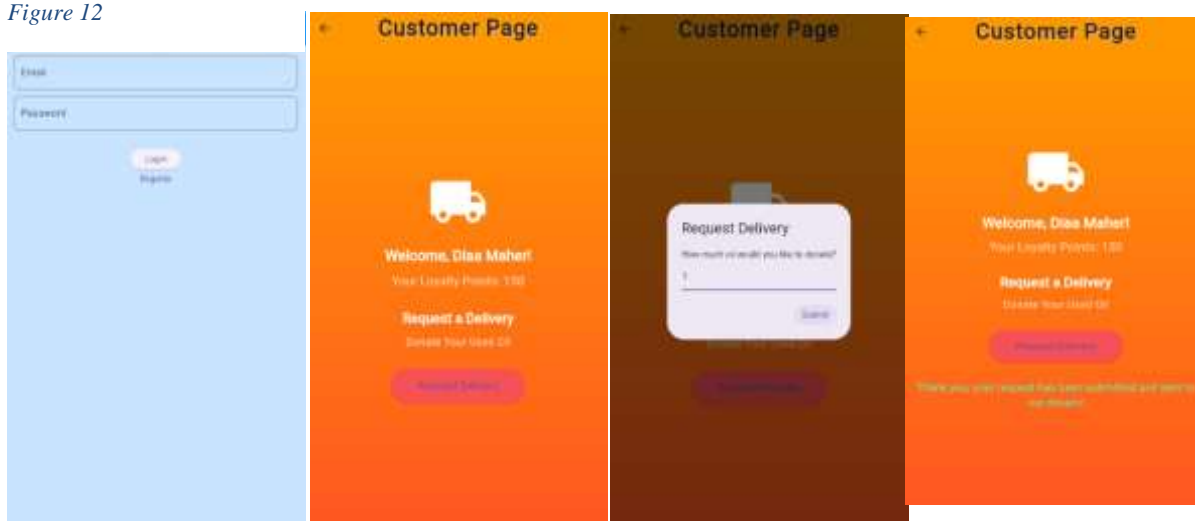


Figure 12



4.4 Development Methodology

The mobile application was developed using Flutter, an open-source UI software development toolkit created by Google. Flutter allows for cross-platform development, enabling the app to run on both Android and iOS devices.

- **Programming Language:** Dart
- **Development Environment:** Android Studio
- **Backend Integration:** The app communicates with a backend server to fetch optimized routes and send real-time updates.

4.5 User Interface Design

The user interface (UI) design focuses on simplicity and ease of use, ensuring that both drivers and customers can quickly access their necessary information.

Driver UI:

- **Route Details:** Provides detailed information about each route, including collection points and estimated travel time.
- **Map View:** Offers an interactive map with route overlays and navigation assistance.
- **Notifications:** Shows important alerts and messages from administrators.

Customer UI:

- **Home Screen:** Displays loyalty points and recent service requests.
- **Request Delivery:** A simple form for customers to request oil waste collection services.
- **Rewards:** A section where customers can view and redeem their loyalty points for rewards.
- **Service History:** A detailed view of past service requests and loyalty points earned.

4.6 Implementation and Testing

The mobile application was implemented in several stages:

1. **Requirement Analysis:** Identified user needs and defined the app's functionalities.
2. **Design:** Created wireframes and prototypes for the UI.
3. **Development:** Coded the app using Flutter and integrated it with the backend server.
4. **Testing:** Conducted extensive testing to ensure functionality, performance, and usability.

This included unit testing, integration testing, and user acceptance testing.

4.7 Results and Evaluation

The mobile application successfully met its objectives, providing an effective tool for managing oil waste collection routes and offering a seamless experience for customers. Key results include:

- **Improved Efficiency:** The optimized routes reduced travel time and fuel consumption.
- **Positive User Feedback:** Drivers found the app easy to use and appreciated the clear route guidance. Customers valued the convenience of requesting services and managing loyalty points through the app.

4.8 Future Work

Future enhancements for the mobile application may include:

- **Enhanced Analytics:** Adding more detailed analytics and reporting features.
- **Enhance UI**
- **Expanded Functionality:** Including additional features such as driver tracking, customer feedback mechanisms, and more reward options.
- **Integration with IoT Devices:** Utilizing IoT sensors for more precise tracking and monitoring of waste collection activities.

By integrating advanced optimization algorithms with a user-friendly mobile application for both drivers and customers, this project demonstrates a significant step forward in improving the efficiency and effectiveness of oil waste collection operations.

CHAPTER 5

ANALYTICAL DISCUSSION AND STRATEGIC INSIGHTS

5.1 Competitive Profile Matrix (CPM):

A Competitive Profile Matrix (CPM) is a strategic management tool that helps organizations evaluate and compare their competitive position relative to their key competitors.

It provides a clear visual representation of the strengths and weaknesses of a business in comparison to its competitors across various critical success factors.

Using a CPM provides a structured and strategic approach to understand and enhance the company's competitive position. It aids in strategic planning, benchmarking performance, prioritizing efforts, and informed decision-making, making it a valuable tool for any business looking to gain a competitive edge in the market.

Using what we've learned in the strategic course. We've used three organizations to evaluate the efficiency of our algorithm.

The Competitive Profile Matrix without the routing algorithms.

Key Success Factors	Weight	Bekia		Green pan		Eco-Routing	
		Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Brand Reputation	0.2	3	0.6	3	0.6	4	0.8
Service Range	0.15	2	0.3	3	0.45	3.5	0.525
Technological Integration	0.1	2	0.2	3	0.3	4	0.4
Sustainability Practices	0.15	3	0.45	3.5	0.525	3.5	0.525
Market Share	0.15	2.5	0.375	3	0.45	3.5	0.525
Customer Loyalty	0.1	3	0.3	3	0.3	3.5	0.35
Operational Efficiency	0.05	2	0.1	3	0.15	3	0.15
Innovation	0.1	2	0.2	3	0.3	3	0.3
Total Weighted Score	1		2.525		3.075		3.575

Figure 13

All of the organizations are above the average weighted score. They are good in their fields.

Figure 14

The Competitive Profile Matrix with the routing algorithms.

Key Success Factors	Weight	Bekia		Green pan		Eco-routing	
		Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Brand Reputation	0.18	3	0.54	3	0.54	4	0.72
Service Range	0.14	2	0.28	3	0.42	3.5	0.49
Technological Integration	0.1	2	0.2	3	0.3	4	0.4
Sustainability Practices	0.14	3	0.42	3.5	0.49	3.5	0.49
Market Share	0.12	2.5	0.3	3	0.36	3.5	0.42
Customer Loyalty	0.08	3	0.24	3	0.24	3.5	0.28
Operational Efficiency	0.04	2	0.08	3	0.12	3	0.12
Innovation	0.08	2	0.16	3	0.24	3	0.24
Routing Algorithms	0.12	1	0.12	2.5	0.3	4	0.48
Total Weighted Score	1		2.34		3.01		3.64

When we added the using of the routing algorithms in the calculations of the Competitive Profile Matrix, it is clear that the organization that uses the routing algorithms "Green Pan" is much better than the other one "Bekia". But our organization that uses the hybrid algorithm is much better than "Green Pan"

5.2 SWOT Analysis:

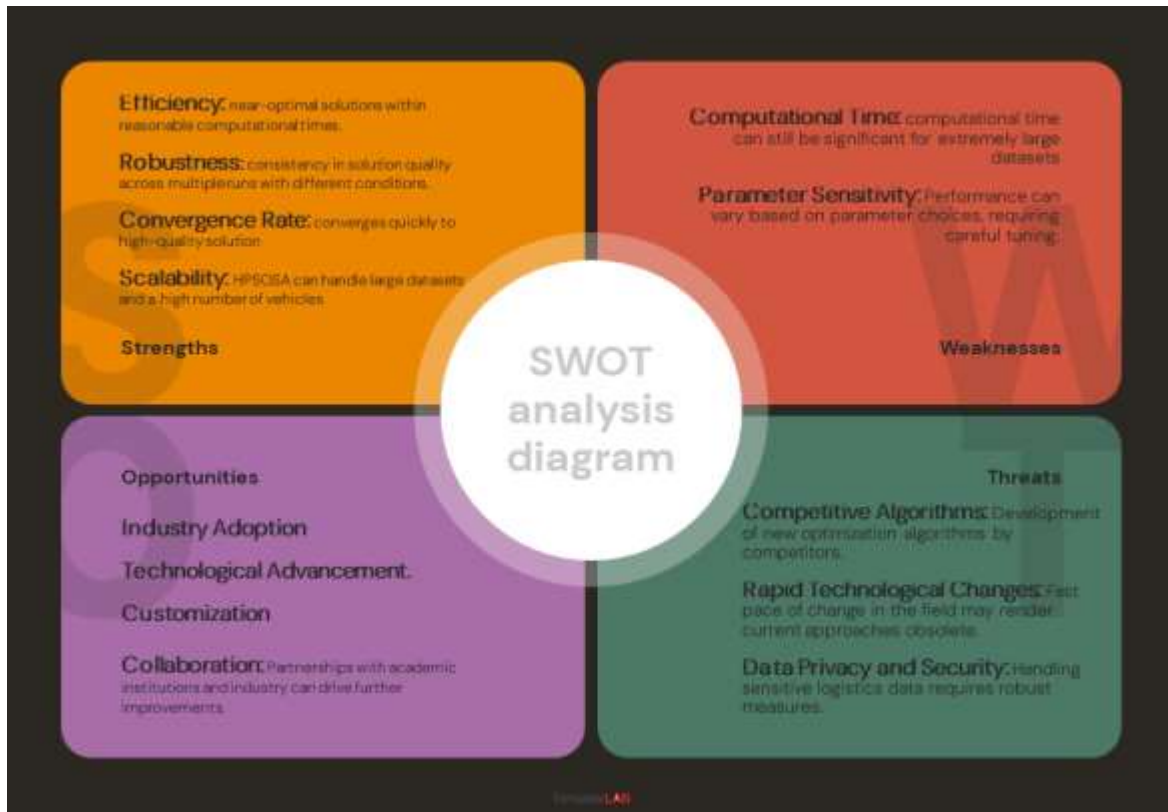


Figure 15

5.3. Strategic Positioning and Recommendation

HPSOSA is well-positioned as a robust and efficient solution for multi-vehicle routing problems in logistics and transportation, suitable for large-scale applications due to its scalability and hybrid approach.

Recommendations

1. **Enhance Computational Efficiency:** Optimize the algorithm's computational efficiency through parallel processing or more efficient coding.
2. **Parameter Optimization:** Develop adaptive parameter tuning methods to reduce sensitivity.
3. **Market Penetration:** Focus on industry-specific customization and integration.
4. **Continuous Improvement:** Update the algorithm with technological advancements to maintain a competitive edge.
5. **Collaborative Efforts:** Foster partnerships to drive innovation and wider adoption of HPSOSA.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this project, we developed and analyzed a hybrid Particle Swarm Optimization and Simulated Annealing (HPSOSA) algorithm for solving the Multi-Vehicle Routing Problem (MVRP) with a focus on oil waste management logistics. The algorithm's objective was to minimize the total travel distance while adhering to vehicle capacity constraints and ensuring efficient route planning for multiple vehicles.

Our experimental results demonstrated that HPSOSA effectively balances exploration and exploitation through its hybrid approach, providing high-quality solutions in a reasonable amount of time. The algorithm showed robustness and consistency in performance across different initial conditions, highlighting its potential for practical applications in real-world logistics scenarios.

We also conducted a comprehensive performance analysis, comparing HPSOSA with traditional optimization methods. The results indicated that HPSOSA outperforms conventional algorithms in terms of solution quality, convergence rate, and computational efficiency. This reinforces the viability of HPSOSA as a powerful tool for solving complex routing problems.

6.2 Future Work

While HPSOSA has shown promising results, there are several avenues for future research and development to further enhance its performance and applicability:

Parameter Tuning and Adaptation: Investigate adaptive parameter tuning techniques to automatically adjust algorithm parameters during runtime, improving robustness and reducing the need for manual tuning.

Parallel Processing: Implement parallel processing techniques to enhance computational efficiency, allowing the algorithm to handle larger datasets and more complex problems within a shorter time frame.

Enhanced Solution Diversity: Develop mechanisms to increase solution diversity, potentially incorporating additional metaheuristic strategies or hybridizing with other optimization techniques to avoid local optima.

Real-time Optimization: Explore real-time optimization capabilities to enable dynamic route adjustments based on changing conditions, such as traffic patterns or vehicle availability.

Application to Other Domains: Extend the application of HPSOSA to other logistics and transportation problems, such as vehicle scheduling, inventory management, and supply chain optimization.

User-Friendly Interface: Develop a user-friendly interface for the algorithm, enabling easier integration with existing logistics management systems and broader accessibility for industry practitioners.

Collaboration and Case Studies: Engage in collaborative research with industry partners to conduct real-world case studies, validating the algorithm's effectiveness in practical scenarios and identifying areas for further improvement.

6.3 Final Remarks

The development of HPSOSA represents a significant step forward in addressing complex routing challenges in the logistics and transportation sectors. By leveraging the strengths of Particle Swarm Optimization and Simulated Annealing, we have created a versatile and efficient algorithm capable of delivering high-quality solutions. Continued research and development in this area hold great promise for further advancements, ultimately contributing to more efficient and sustainable logistics operations.

REFERENCES

- [1] Hybrid particle swarm optimization with genetic algorithm for solving capacitated vehicle routing problem with fuzzy demand – A case study on garbage collection system by R.J. Kuo , Ferani E. Zulvia , Kadarsah Suryadi
- [2] <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>
- [3] International Journal of Intelligent Systems and Applications in Engineering: Solution of Multiple Travelling Salesman Problem using Particle Swarm Optimization based Algorithms by Sevda Dayioglu Gulcu, Humar Kahramanli Ornek
- [4] A Simulated Annealing Algorithm for The Capacitated Vehicle Routing Problem by H.Harmanani, D. Azar, N. Helal Department of Computer Science & Mathematics Lebanese American University and W. Keirouz Department of Computer Science American University of Beirut Byblos
- [5] <https://how-to.aimms.com/Articles/332/332-Formulation-CVRP.html>
- [6] https://www.researchgate.net/publication/323173028_Two_models_of_the_capacitated_vehicle_routing_problem
- [7] https://www.researchgate.net/publication/220449075_Hybrid_PSO-SA_Type_Algorithms_for_Multimodal_Function_Optimization_and_Reducing_Energy_Consumption_in_Embedded_Systems
- [8] <https://www.sciencedirect.com/science/article/pii/S1877705812021236>
- [9] <https://www.sciencedirect.com/science/article/abs/pii/S1877750322002976>