2025-01-16

Clustering-Based Improved Ant Colony Optimization for the Multi-Trip Vehicle Routing Problem with Heterogeneous Fleet and Time Windows: An Industrial Case Study

Kim, Beom Sae Shawn

UNIVERSITY OF CALGARY

Clustering-Based Improved Ant Colony Optimization for the Multi-Trip Vehicle Routing Problem with Heterogeneous Fleet and Time Windows: An Industrial Case Study

by

Beom Sae Shawn Kim

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN GEOMATICS ENGINEERING

CALGARY, ALBERTA
JANUARY, 2025

Abstract

The growing complexity of logistics and transportation systems has led to significant interest in solving Vehicle Routing Problems (VRP) with realistic constraints. Real-world VRP extends beyond minimizing transportation costs to include balancing workloads among drivers, managing heterogeneous fleets, and adhering to strict time windows. Addressing these challenges requires advanced methodologies that ensure operational efficiency, fairness, and adaptability to practical constraints.

This thesis proposes a Clustering-Based Improved Ant Colony Optimization (CIACO) algorithm, integrating an improved Ant Colony Optimization (IACO) metaheuristic with advanced clustering techniques, including a modified density-based spatial clustering of applications with noise (DBSCAN-Plus) and a Micro-Cluster Fusion Scheme. The framework addresses the multi-trip VRP with heterogeneous fleet and time windows (MTVRPHFTW), focusing on minimizing total travel distance while handling constraints such as travel time, vehicle capacity, heterogeneous fleet configurations, customer-specific time windows, and multi-trip scheduling. Additionally, it ensures balanced workload distribution among vehicles while prioritizing the use of smaller, fuel-efficient vehicles to reduce CO_2 emissions, supporting both operational efficiency and sustainability goals.

This thesis also discusses the development of an interactive Geographic Information System (GIS) visualization system, implemented via custom Quantum Geographic Information System (QGIS) plugins. Designed specifically to enhance the interpretability and application of the CIACO algorithm, this system bridges optimization results with GIS functionality via custom QGIS plugins, providing logistics planners with dynamic visualizations, route overlays with toggling options, advanced filtering capabilities based on metrics such as CO_2 emissions, travel time, travel distance, and vehicle types, and an interactive dashboard for real-time analysis and decision-making support. These interactive features enhance the practicality of the proposed framework for real-world logistics applications, making the solutions more adaptable and actionable.

The proposed framework was validated using industrial data from a Canadian logistics company, demonstrating its effectiveness in addressing complex VRP. Experimental results show that CIACO outperforms existing methods in minimizing travel distance, achieving balanced workload distribution, and reducing environmental impact. The interactive GIS system amplifies the practicality of the approach by translating optimization outcomes into intuitive visualizations.

This thesis advances VRP research by integrating algorithmic optimization with GIS technologies, addressing modern logistical challenges, and offering scalable solutions for industrial applications.

Preface

The research presented in this thesis was conducted under the supervision of Dr. Xin Wang in the Intelligent Geospatial Data Mining (IGDM) Lab, Department of Geomatics Engineering, with additional guidance from Dr. Jeong Woo Kim, also from the Department of Geomatics Engineering.

Chapter 3 of this thesis has been published in:

Beom Sae (Shawn) Kim, Arash Mozhdehi, Yunli Wang, Sun Sun, and Xin Wang. 2024. *Clustering-Based Enhanced Ant Colony Optimization for Multi-Trip Vehicle Routing Problem with Heterogeneous Fleet and Time Windows: An Industrial Case Study.* In *17th ACM SIGSPATIAL International Workshop on Computational Transportation Science GenAI and Smart Mobility Session (IWCTS'24)*, October 29–November 1, 2024, Atlanta, GA, USA. ACM, New York, NY, USA, 10 pages.https://doi.org/10.1145/3681772.3698216

Acknowledgements

I would like to sincerely express my gratitude to my supervisor, Dr. Xin Wang, for her continuous support, invaluable guidance, and insightful feedback throughout my research journey. Her encouragement and expertise have been key in shaping this thesis.

I am also deeply thankful to my co-supervisor, Dr. Jeong Woo Kim, for his helpful advice and suggestions, which have contributed significantly to this work.

I want to extend my gratitude to the members of the Intelligent Geospatial Data Mining (IGDM) Lab for their support and collaboration during my research.

I am also grateful to the National Research Council of Canada's Artificial Intelligence for Logistics Program and the Natural Sciences and Engineering Research Council of Canada (NSERC) for their support. Special thanks to Dr. Sun Sun and Dr. Yunli Wang for their valuable suggestions and guidance on various aspects of this research.

Lastly, I wish to express my deepest gratitude to my parents, Bok Chul Kim and Chang Ha Kim, and my family members for their constant support and encouragement throughout my master's studies. Their belief in me and their sacrifices have been my greatest source of strength. I also want to thank all my friends in Calgary and South Korea for their encouragement and support, which have been a great comfort to me during this journey. Without the love and support of my family and friends, this accomplishment would not have been possible.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols and Abbreviations

Symbol	Definition
ACO	Ant Colony Optimization
CIACO	Clustering-Based Improved Ant Colony Optimization
CO_2	Carbon Dioxide
CRS	Coordinate Reference System
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
GeoJSON	Geographic JavaScript Object Notation
GIS	Geographic Information System
HF	Heterogeneous Fleet
IACO	Improved Ant Colony Optimization
MT	Multi-Trip
MTVRPHFTW	Multi-Trip Vehicle Routing Problem with Heterogeneous Fleet and Time Windows
OSM	OpenStreetMap
OSMnx	OpenStreetMap Network Extractor (Python Library)
QGIS	Quantum Geographic Information System
RL	Route Length
SD	Standard Deviation
TLD	Total Load Delivered
TSP	Traveling Salesman Problem
TTT	Total Travel Time
TW	Time Windows
VRP	Vehicle Routing Problem

Chapter 1

Introduction

1.1 Background

Efficient logistics and transportation planning are critical for addressing real-world challenges, such as reducing operational costs, minimizing environmental impacts, and meeting customer demands within tight time constraints. The Vehicle Routing Problem (VRP), first formalized by Dantzig and Ramser in 1959, is a fundamental combinatorial optimization problem [2]. It focuses on determining the most efficient routes for a fleet of vehicles to deliver goods to a set of customers [2, 3]. Over the years, numerous VRP variants have been developed, including the VRP with Time Windows (VRPTW), Multi-Trip VRP (MTVRP), and Heterogeneous Fleet VRP (HFVRP), to address practical challenges such as time restrictions, fleet heterogeneity, and multi-trip requirements. These adaptations highlight the increasing demand for advanced routing strategies as logistics operations grow more complex with the expansion of global supply chains.

Logistics costs account for a significant portion of operational expenses, with distribution costs comprising 15% to 40% of the total price of goods and, in some cases, exceeding 30% of overall logistics costs [4, 5, 6]. In Canada, vast geography and diverse markets further emphasize the need for effective logistics strategies to control costs and improve efficiency. Studies show that optimizing logistics operations can reduce transportation expenses by 10% to 30% [7], significantly enhancing delivery reliability and on-time performance [8]. These improvements not only lead to better financial outcomes but also strengthen a company's competitive position in the marketplace.

Transportation costs, which often constitute 44% to 50% of total logistics expenses, further emphasize the importance of advanced optimization methods [9, 10]. Addressing these expenses while maintaining high service quality is a pressing concern for industries worldwide. Traditional optimization techniques, while effective in certain contexts, often struggle with the scale and complexity of modern logistics. This has driven interest in metaheuristic algorithms, such as Ant Colony Optimization (ACO), which are particularly effective for solving VRP-related problems. However, many existing approaches fail to address critical issues such as load balancing, equitable workload distribution, and the incorporation of real-world constraints, especially in industrial applications.

In parallel, Geographic Information Systems (GIS) have become essential tools for managing and analyzing spatial data, supporting key tasks such as route optimization and operational planning. The transition from static GIS tools to interactive systems, particularly those powered by custom-developed plugins in platforms like Quantum Geographic Information System (QGIS), has significantly expanded their capabilities. QGIS, as an open-source platform, is especially attractive due to its adaptability, cost-effectiveness, and ability to support real-time analysis [11, 12]. By integrating GIS with advanced optimization methods, decision-makers can bridge the gap between theoretical models and practical applications. These systems not only enhance analytical capabilities but also provide actionable insights and interactive tools that empower decision-makers to explore and refine solutions. Such advancements ensure that GIS systems evolve from passive mapping tools into dynamic, decision-support systems designed to address complex logistical challenges.

As logistics operations become increasingly complex, the integration of advanced optimization techniques and interactive GIS systems has become essential. Together, they offer effective solutions for addressing modern logistical challenges, ensuring efficient, cost-effective, and sustainable operations.

1.1.1 Clustering-Based Improved Ant Colony Optimization

The Multi-Trip VRP with Real Heterogeneous Fleet and Time Windows (MTVRPHFTW) presents significant challenges in logistics due to its complex constraints, such as diverse vehicle capacities, time windows for customer service, and the requirement for multi-trip routing [1, 13]. Traditional approaches, while effective for simpler VRP variants, often struggle to address issues like load balancing, equitable workload distribution, and environmental impacts in such complex scenarios [1, 14].

Clustering techniques have emerged as a powerful tool to simplify problem complexity in large-scale logistics systems. By grouping customers into clusters based on spatial proximity, demand, and operational constraints, these methods transform a global problem into smaller, more manageable subproblems. This decomposition reduces computational effort and enables more targeted optimization strategies. Among clustering approaches, density-based algorithms like DBSCAN (Density-Based Spatial Clustering of Applications with Noise) are particularly effective, as they identify customer groups by considering spatial and demand-related features, making them well-suited for logistics optimization challenges.

The integration of clustering with metaheuristic algorithms such as ACO presents a compelling approach to enhancing optimization outcomes. ACO, inspired by the foraging behavior of ants, has proven effective in solving complex routing and scheduling problems [15]. Combining clustering for problem decomposition with ACO for route optimization enhances both scalability and solution quality, enabling logistics systems to address large-scale and industrial-level challenges effectively. Recent studies highlight how such integrations improve operational efficiency and address sustainability goals, making them highly relevant for modern logistics systems [16, 17, 18].

This broader framework highlights the potential of combining clustering techniques and ACO to tackle the unique demands of MTVRPHFTW. By focusing on scalability and solution quality as complementary goals, these approaches provide valuable tools for optimizing logistics operations in increasingly complex and competitive environments.

1.1.2 Interactive GIS Systems with Custom Developed QGIS plugins

The increasing complexity of logistics, driven by real-time demands and sustainability goals, has elevated the importance of GIS in providing effective tools for spatial data visualization and analysis. Traditional GIS systems have consistently supported tasks such as route planning, fleet management, and spatial trend analysis. However, their static nature often limits their ability to adapt to evolving logistical requirements, such as processing dynamic data or exploring alternative scenarios interactively [12, 19]. These limitations have led to the development of interactive GIS systems, which are better suited for addressing modern logistical challenges.

Interactive GIS systems distinguish themselves by enabling users to interact dynamically with spatial data. Unlike static GIS tools that primarily serve as repositories of pre-analyzed data, interactive systems provide real-time filtering, visualization, and scenario exploration capabilities. This evolution significantly enhances their applicability in logistical planning, where decision-makers must frequently evaluate trade-offs, assess operational efficiency, and respond to dynamic conditions. For instance, by visualizing optimized routes, customer locations, and resource allocations, these systems allow planners to gain actionable insights and make informed decisions more efficiently. When integrated with optimization outputs, interactive GIS systems further transform raw data—such as route distances, travel times, or CO_2 emissions—into visually interpretable insights that support effective decision-making.

This shift towards interactivity in GIS systems has driven the adoption of platforms like QGIS, which provide adaptable tools for creating interactive solutions across various applications, including logistics. QGIS, an open-source GIS platform, has emerged as a prominent tool for developing interactive systems across various domains, including urban planning, environmental management, and logistical operations. Its adaptability and wide range of functionalities make it suitable for addressing complex spatial challenges, such as optimizing resource allocation, analyzing geographical trends, or enhancing decision-making processes in diverse fields.

The extensibility of QGIS through custom Plugin development enables the creation of tailored tools that enhance functionalities such as integrating real-world road networks, visualizing spatial data dynamically, and filtering information based on scenarios. These capabilities allow QGIS to evolve from a static mapping platform into an interactive system, providing decision-makers with advanced tools to address complex logistical challenges [20, 21].

The integration of dynamic data and interactivity represents a significant advancement in GIS systems, bridging the gap between theoretical models and practical applications. This adaptability enables logistics planners to visualize, analyze, and explore alternative scenarios more effectively, supporting informed decision-making and efficient resource allocation. As GIS continues to evolve, its integration with optimization tools promises to transform logistical planning, setting the stage for innovative and sustainable decision-support systems.

1.2 Problem Statement and Research Objectives

1.2.1 Problem Statement

Industries today face increasing pressure to optimize logistics and transportation systems to reduce costs, minimize environmental impacts, and meet strict service requirements. The MTVRPHFTW reflects these challenges, but existing approaches frequently fail to address its operational complexities effectively. Traditional optimization methods struggle to accommodate the diverse constraints associated with MTVRPHFTW, such as heterogeneous fleets, multi-trip routing requirements, and time windows. These limitations restrict their ability to achieve load balancing, equitable workload distribution, and environmental sustainability in large-scale, real-world logistics problems.

GIS, widely used in logistical planning, presents additional challenges due to its lack of interactivity. While GIS tools have traditionally supported route planning and spatial analysis, static systems cannot integrate dynamically with outputs from advanced optimization models. This limitation constrains decision-makers from exploring alternative scenarios or visualizing key metrics, such as total traveling time, total traveled route length, route efficiency, and CO_2 emissions, in an interactive and accessible manner. Although recent advancements in GIS technology have introduced interactive capabilities, the seamless integration of GIS with optimization algorithms remains underdeveloped, creating a significant gap between theoretical solutions and practical applications.

To address these challenges, a comprehensive solution is needed that integrates advanced optimization techniques with interactive GIS systems. This solution should effectively address the operational complexities of MTVRPHFTW while providing dynamic visualization and actionable insights through user-friendly interfaces. Such a framework would not only optimize logistical operations but also equip decision-makers with the tools required to adapt to evolving demands efficiently and sustainably.

1.2.2 Research Objectives

This research aims to address the challenges of solving the MTVRPHFTW by proposing an integrated framework that combines advanced optimization techniques with an interactive GIS system developed through custom QGIS plugins. The primary objective is to develop a robust solution that enhances route optimization while addressing critical constraints, such as load balancing, and promotes the use of smaller, fuel-efficient vehicles, ultimately aiming to reduce both operational costs and CO_2 emissions.

A key objective of this study is to develop the CIACO algorithm, a novel approach tailored to address the intricate requirements of MTVRPHFTW. This algorithm integrates clustering techniques, including DBSCAN-Plus and a Micro-Cluster Fusion Scheme, alongside improved ACO methods. These techniques aim to effectively handle constraints such as time windows, vehicle heterogeneity, and multi-trip operations, with the goal of improving operational efficiency, workload equity, and route optimization.

Another important goal is to design and implement a user-friendly interactive GIS system within QGIS that integrates seamlessly with the outputs of CIACO. Developed through custom QGIS plugins, this system will offer dynamic visualization and filtering capabilities for key logistical metrics, including travel distance, travel time, vehicle utilization, and CO_2 emissions. It will also feature advanced filtering options, temporal analysis, and interactive dashboards displaying statistics for individual routes or aggregated totals. To support decision-making processes, the system will enable users to explore and refine optimization results through scenario-based exploration, allowing decision-makers to dynamically assess the impact of various constraints and metrics. By improving the accessibility and usability of optimization outcomes, this system will empower users to make informed, actionable decisions, ensuring practical applicability for logistics planning.

Finally, the integrated framework will be subjected to thorough validation through benchmarking and real-world testing. The performance of CIACO and the GIS system will be evaluated against existing methods and industry benchmarks using real-world industrial data. Detailed case studies will demonstrate the effectiveness of the proposed approach in addressing the challenges of MTVRPHFTW.

By achieving these objectives, this research intends to deliver a comprehensive and practical solution to the complexities of MTVRPHFTW. It focuses on advancing optimization methodologies through the development of CIACO while enhancing logistical decision-making with an interactive GIS system. Ultimately, this work bridges the gap between theoretical models and practical applications, offering actionable tools to the field of logistics and transportation planning.

1.3 Research Contributions

This thesis presents a comprehensive solution to the MTVRPHFTW, combining advanced optimization algorithms with an interactive GIS system developed through custom QGIS plugins. The contributions of this thesis correspond directly to the identified research problems and objectives, offering targeted solutions to key logistical challenges. These contributions are summarized as follows:

1. **Development of CIACO Algorithm to Solve the MTVRPHFTW**

   To address the complexities of the MTVRPHFTW, this research developed the Clustering-Based Improved Ant Colony Optimization (CIACO) algorithm. CIACO enhances traditional ACO by incorporating advanced clustering techniques, such as DBSCAN-Plus and a Micro-Cluster Fusion Scheme, to effectively manage constraints, including time windows, heterogeneous fleets, and multi-trip requirements. These clustering methods simplify the problem representation by grouping customers based on spatial proximity and demand, enabling more efficient route optimization. Additionally, CIACO minimizes CO_2 emissions by prioritizing smaller, fuel-efficient vehicles during route planning. These enhancements make CIACO a scalable and sustainable solution for real-world logistics challenges.

2. **Integration of CIACO Outputs with an Interactive GIS System**

   A user-friendly interactive GIS system was developed through custom QGIS plugins to integrate CIACO's outputs for practical logistics planning. CIACO generates GeoJSON outputs containing key attributes such as travel time, travel distance, CO_2 emissions, vehicle types, and customer and depot locations. If the outputs exhibit spatial discrepancies, such as coordinate precision mismatches or CRS inconsistencies, these are recalibrated using Dijkstra's algorithm to ensure alignment with real-world road networks.

   The interactive GIS system provides advanced filtering options, allowing users to refine routes based on specific criteria such as vehicle type, CO_2 emissions, and route length and travel time (with user-defined range settings). Additional features include dynamic visualization of routes, toggling layers, and interactive dashboards. These functionalities enable decision-makers to effectively refine and analyze routes, offering both route-specific and aggregated metrics, including travel distance, emissions, and delivery efficiency.

3. **Validation and Benchmarking Using Real-World Industrial Data**

   The integrated framework was rigorously validated using real-world industrial data. CIACO was benchmarked against existing optimization methods and industry standards, with performance metrics such as computational efficiency, route optimization quality, and environmental impact being assessed. The interactive GIS system enabled practical implementation by aligning optimization results with real-world constraints. Detailed case studies demonstrated the framework's ability to address the MTVRPHFTW effectively, highlighting its robustness and applicability to large-scale logistical challenges.

This thesis delivers a practical and comprehensive solution to the MTVRPHFTW. The integration of CIACO with the interactive GIS system ensures operational efficiency, workload equity, and environmental sustainability. Ultimately, this research bridges the gap between theoretical optimization models and practical applications, offering actionable tools for decision-makers in dynamic logistics environments.

1.4 Thesis Outline

This thesis is structured into five chapters. Chapter 1 introduces the research, outlining the background, problem statement, objectives, and contributions, while emphasizing the need for advanced optimization techniques and interactive GIS systems to address the challenges of MTVRPHFTW. Chapter 2 provides a review of the literature on VRP variants, optimization methods like ACO, and the integration of GIS in logistics, including interactive GIS systems through QGIS plugins, identifying key gaps that this research aims to address.

Chapter 3 focuses on the development of the CIACO, focusing on methodologies such as DBSCAN-Plus clustering and Micro-Cluster Fusion, and presents experimental results that demonstrate its effectiveness in improving route optimization and operational efficiency. Chapter 4 discusses the creation of an interactive GIS system through custom QGIS plugins, highlighting its integration with CIACO outputs for dynamic visualization, filtering, and decision-making in logistics planning.

Finally, Chapter 5 summarizes the research contributions, discusses limitations, and proposes future directions to further enhance optimization and GIS integration, concluding with the significance of this work in solving complex logistical challenges.

Chapter 2

Related Works

2.1 Introduction

This chapter reviews the existing literature relevant to the development of the Clustering-Based Improved Ant Colony Optimization (CIACO) algorithm and its application to solving the Multi-Trip Vehicle Routing Problem with Heterogeneous Fleet and Time Windows (MTVRPHFTW). It further explores the integration of interactive Geographic Information Systems (GIS) with optimization algorithms, emphasizing the development and application of custom QGIS Plugins. The discussion focuses on their role in enhancing spatial data analysis, improving route optimization, and supporting decision-making processes in logistics and transportation planning.

Section 2.2 explores the evolution of the VRP and its numerous variants. Researchers adapted classical VRP to address real-world logistics challenges by incorporating constraints such as time windows, heterogeneous fleets, and multi-trip requirements. These adaptations expanded the problem's scope and complexity, necessitating the use of advanced optimization techniques. Among these, ACO emerged as a powerful approach due to its ability to tackle combinatorial optimization problems effectively. The review discusses ACO's strengths and limitations, particularly its challenges in addressing load balancing, equitable workload distribution, and environmental impacts. These insights formed the basis for CIACO's design, which aimed to enhance solution quality and applicability in industrial logistics settings.

Section 2.3 reviews advancements in GIS applications within logistics. Traditionally, GIS has played a critical role in managing and analyzing spatial data, supporting route optimization, and improving operational efficiency. However, the transition from static GIS tools to interactive systems has enabled greater user engagement and dynamic analysis capabilities. The review highlights the role of custom-developed QGIS Plugins in enabling real-time data visualization, filtering, and spatial analysis. By integrating GIS with optimization outputs, these tools bridged the gap between theoretical methods and their practical application in logistics.

This review establishes the foundation for this thesis by presenting relevant advancements in optimization methods and GIS integration. It highlights the gaps in existing literature, particularly the need for advanced algorithms like CIACO and interactive GIS systems, to address the challenges of solving the MTVRPHFTW in realistic and complex logistical environments.

2.2 Advancements in VRP and Optimization Methods

2.2.1 Variants of VRP and Challenges

The Vehicle Routing Problem (VRP) originates from the Traveling Salesman Problem (TSP). In their seminal 1959 work, Dantzig and Ramser introduced the "Vehicle Routing Problem" in their study, The Truck Dispatching Problem, which focused on optimizing fuel delivery [2].

The basic VRP involves determining a set of routes that begin and end at a depot, covering a group of customers. Each customer has specific demands, and the capacity of vehicles limits the number of customers serviced on a single route. The goal is to minimize total travel distance, the number of vehicles, or both. Over time, researchers have expanded VRP to address real-world logistics complexities, leading to numerous variants.

The VRP consists of a wide range of variants that address different logistical constraints and operational complexities. These include Capacitated VRP (CVRP), VRP with Time Windows (VRPTW), VRP with Simultaneous Pickup and Delivery (VRPSPD or PD or Backhauls (-B)), Multi-Depot VRP (MDVRP), Multi-Trip VRP (MTVRP), Time-Dependent VRP (TDVRP), Heterogeneous Fleet VRP (HVRP), and Dynamic VRP (DVRP).

One foundational variant, the Capacitated VRP (CVRP), focuses on ensuring that vehicle loads do not exceed their carrying capacity [13]. CVRP often includes additional constraints like route length limitations [13]. Another widely studied variant, the VRP with Time Windows (VRPTW), requires each customer to be serviced within specific time intervals [22].

The VRP with Simultaneous Pickup and Delivery (VRPSPD) allows vehicles to both deliver and collect goods from customers during the same visit, addressing bidirectional logistics needs such as grocery delivery and pickup [23]. This variant mandates that customers be visited only once, further complicating route optimization [23].

The Multi-Depot VRP (MDVRP) extends the problem to scenarios with multiple depots. Researchers have proposed approaches like clustering-based assignments (e.g., K-means) and urgency-based assignments to efficiently allocate customers to depots before solving VRPs for each one [24, 25]. This variant introduces additional computational complexity in balancing customer-to-depot allocations and overall route optimization.

The Multi-Trip VRP (MTVRP) allows drivers to perform multiple trips per day, driven by constraints like perishable goods and maximum driving hours [26, 27]. This variant is particularly relevant in city logistics, where smaller vehicles handle deliveries within a defined time horizon [26, 27]. As logistical demands have increased, VRP variants have begun addressing the need for repeated vehicle usage, leading to the development of the Multi-Trip VRP (MTVRP).

The Time-Dependent VRP (TDVRP) incorporates time-varying travel times, particularly in congested urban settings [26]. Ichoua et al. proposed modeling travel speed as a stepwise function, leading to dynamic time-dependent travel time functions [28].

The Heterogeneous Fleet VRP (HVRP) introduces another layer of complexity by using vehicles with varied capacities and operational characteristics [29]. Decisions regarding fleet composition and vehicle assignment directly impact operational costs and carbon emissions [29].

Additionally, the Dynamic VRP (DVRP) handles route adaptations in response to real-time changes, such as new customer requests or unexpected delays. Pillac et al. highlighted its ability to accommodate evolving information during route execution, making it critical in dynamic and uncertain environments [30].

Building on these foundational variants, the Multi-Trip Vehicle Routing Problem with Time Windows and Heterogeneous Fleet (MTVRPTWHF) introduces additional complexities, including time windows for each customer, varying vehicle capacities, and the number of skids [1, 13, 14, 31, 32]. This variant addresses not only the scheduling of deliveries within specific time constraints but also the operational challenges of managing a heterogeneous fleet capable of performing multiple trips [29]. A critical limitation is the overall time horizon for completing all routes, reflecting the practical challenge of fulfilling customer demands within finite operational hours.

To address these complexities, solution approaches such as Local Search and Simulated Annealing have been employed, optimizing both route planning and fleet utilization [31]. These methodologies have been evaluated using well-established VRP benchmarks, demonstrating their effectiveness in addressing the unique challenges of MTVRPTWHF [31, 33].

2.2.2 ACO in VRP

ACO algorithms, initially proposed by Dorigo et al. (1996), were widely applied to various combinatorial optimization problems, including VRPs [15]. The algorithm's ability to find optimal paths through probabilistic decision-making and pheromone trails made it particularly suitable for dynamic and complex problems like MTVRP [34]. The specific application of ACO in heterogeneous fleet scenarios became a topic of increasing interest. When ACO was tailored to account for different vehicle types and capacities, it led to more cost-effective and practical routing solutions [1, 15, 22, 34]. This research highlighted the adaptability of ACO to diverse logistical requirements.

Beginning with the foundational work by Blum and Roli [34], which offered a broad overview of metaheuristics, this section set the groundwork for understanding the significant role of ACO in combinatorial optimization. Their analysis highlighted the critical balance between intensification and diversification strategies, essential for effective solution space exploration and exploitation in complex optimization problems, including those in vehicle routing.

Phuc and Thao [35] applied ACO to the complex Multi Pickup and Multiple Delivery Vehicle Routing Problem with Time Window and Heterogeneous Fleets (MPMDVRPTWHF), illustrating ACO's ability to adapt to logistical constraints such as varying vehicle capacities and strict delivery schedules, aiming to minimize total travel costs. This study demonstrated ACO's potential in optimizing routes to accommodate a wide range of operational constraints and fleet diversities, indicating its suitability for complex logistical challenges. Furthering the application of ACO, Yu et al. [36] introduced an improved variant for the VRP, incorporating strategies like the "ant-weight strategy" and mutation operations to boost the algorithm's performance. Their approach confirmed ACO's capacity to produce solutions that were competitive with traditional methods, highlighting its adaptability and efficiency in routing optimization across different problem settings.

Mazzeo and Loiseau [37] concentrated on the Capacitated Vehicle Routing Problem (CVRP), exploring various aspects of the ACO algorithm, such as route building, transition rules, pheromone updates, and the implementation of improvement heuristics. Their findings pointed to ACO's competitive advantage over other metaheuristics, especially in handling problems up to 50 nodes and showing promising applicability for larger issues.

Gupta and Saini [38] enhanced the traditional ACO framework for the Vehicle Routing Problem with Time Windows (VRPTW) by introducing a new pheromone reset and update function alongside a 2-opt method for path improvement. Their version of the enhanced ACO demonstrated substantial improvements in routing optimization, emphasizing the algorithm's effectiveness in managing time-constrained routing challenges.

Wu et al. [39] proposed a Hybrid Ant Colony Optimization (HACO) for the VRPTW, incorporating a unique blend of strategies, including a novel pheromone update method, adaptive parameters, and mutation operations. This approach aimed to overcome the limitations of traditional ACO by enhancing solution diversity and avoiding premature convergence to local optima. Based on Solomon's instances, their experimental results demonstrated HACO's effectiveness in optimizing routes within specified time windows, highlighting its practical implications for complex routing problems. This study further solidified the adaptability and potential of ACO-based methods in addressing the intricacies of VRPTW, contributing to the ongoing development of efficient logistical and transportation solutions.

2.2.3 Gaps in Literature and Motivation for CIACO

Despite extensive research on traditional VRPs, studies specifically addressing MTVRPHFTW were limited, particularly in terms of investigating this problem under realistic conditions and benchmarking performance using real industrial data. This gap highlighted the need for more research focused on practical solutions to MTVRPHFTW.

ACO demonstrated significant efficiency in addressing various variants of vehicle routing problems, including those with heterogeneous fleets and multi-trip requirements [1, 15, 22, 34]. By continuously evolving through improvements and adaptations, ACO emerged as a powerful tool in logistics and transportation, significantly reducing the number of vehicles, travel distances, and operational costs. However, most studies focused primarily on minimizing travel distance or time, overlooking critical aspects like load imbalance. Ensuring equitable workload distribution among vehicles and drivers remained a challenge, as it was crucial for operational equity and long-term sustainability.

Addressing load imbalance and environmental impacts was vital for creating a sustainable and equitable logistics system. Load imbalance in VRP referred to the unequal distribution of workloads among drivers, often leading to stress and dissatisfaction. Matl et al. [40] argued that equitable workload allocation could be achieved without significantly compromising cost minimization objectives, emphasizing its importance in maintaining driver morale and operational consistency. Similarly, Lehuédé et al. [41] highlighted the role of workload balancing in multi-objective VRP settings, where metrics like route length and duration quantified workload disparities. Bender et al. [42] further noted that balanced and stable delivery districts enhanced long-term service reliability by ensuring predictable workloads.

Environmental considerations in VRP were increasingly recognized as essential in modern logistics. Abdullahi et al. [43] pointed out that sustainable VRP research focused on minimizing greenhouse gas emissions, aligning with global climate goals and improving logistical efficiency. Bektaş and Laporte [44] introduced the Pollution-Routing Problem, which explicitly incorporated environmental factors into routing decisions, demonstrating the trade-offs between cost and ecological impact.

CIACO addressed these gaps by integrating clustering and improved ACO to handle real-world constraints such as heterogeneous fleets, time windows, and multi-trip requirements. While focusing on reducing total travel distance as the primary objective, CIACO also handled challenges like balancing workload distribution, minimizing travel costs, and reducing environmental impacts by treating these aspects as constraints and systematically improving their outcomes. This approach provided a comprehensive and practical solution to the challenges of MTVRPHFTW.

2.3 GIS in Logistics: From Applications to Interactive Solutions and Challenges

2.3.1 The Role of GIS in Modern Logistics and Spatial Analysis

GIS has become a powerful tool for managing, analyzing, and visualizing spatial data. Advances in geospatial technologies and the increasing availability of data significantly broadened GIS applications across diverse fields, including urban planning, environmental management, public health, disaster response, and logistics [45, 46]. Within logistics, GIS revolutionized route optimization by utilizing spatial data to improve efficiency, reduce costs, and support data-driven decision-making [47].

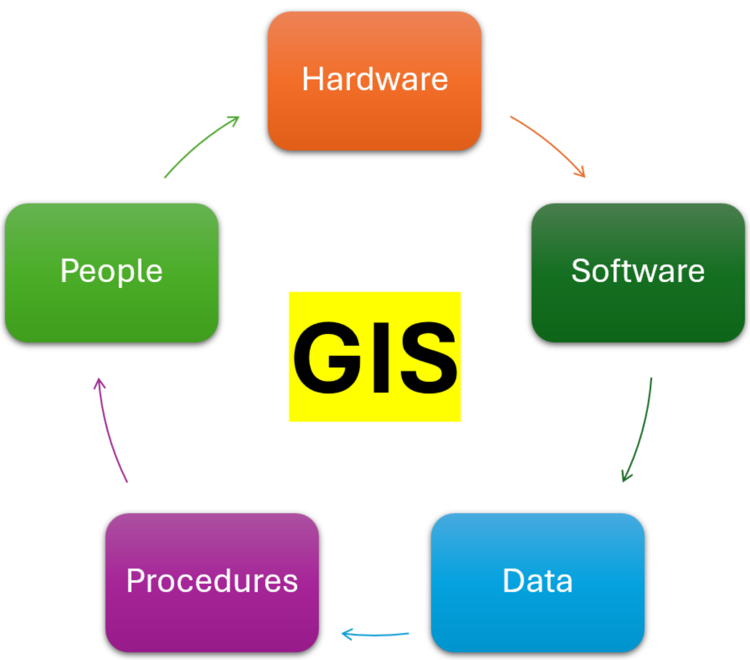

Figure 2.1: Core components of GIS

The core components of GIS—hardware, software, data, procedures, and people—served as the foundation for its effectiveness (Figure 2.1) [48]. Hardware included devices such as computers, servers, and GPS tools that enabled the collection and storage of spatial data [49]. Software provided analytical and visualization capabilities essential for interpreting spatial relationships and generating maps [49][50]. Data, comprising vector and raster formats sourced from GPS, surveys, or remote sensing, formed the foundation of GIS analyses [51][52]. Procedures established the methods for processing, analyzing, and disseminating spatial information, ensuring consistency and accuracy [52]. Lastly, GIS professionals and end-users applied these outputs to solve complex spatial problems [49][50].

In logistics, these components collectively enabled applications such as supply chain analysis, operational improvements, and route optimization. Recent studies demonstrated GIS's ability to address logistical challenges, particularly by integrating real-time data from road networks to create more resilient and efficient routing strategies [11]. Additionally, GIS was applied to support multi-objective optimization tasks, such as minimizing environmental impacts while enhancing operational performance [53].

Integrating GIS with advanced optimization algorithms, such as the CIACO, further enhanced its utility in logistics. This thesis explored such integration to address complex problems, including MTVRPHFTW. Combining GIS's spatial analysis capabilities with its interactive functions and optimization algorithms allowed for precise route planning, incorporating real-world constraints such as load balancing, time windows, and road network alignment. GIS's interactive functions enhanced user engagement by enabling dynamic data visualization, real-time analysis, and scenario exploration, making complex spatial relationships more accessible. By employing GIS to process, analyze, and interact with spatial data, logistics systems achieved more practical, accurate, and user-friendly solutions to routing challenges [19, 11].

2.3.2 Developing Interactive GIS Systems for Logistics Applications

The evolution of GIS technology led to the development of interactive systems, which allowed users to engage directly with spatial data for real-time analysis and manipulation. This interactivity significantly improved user experience, supported better decision-making, and enhanced data exploration [12, 54]. By providing immediate feedback, interactive GIS enabled users to visualize changes and comprehend complex spatial relationships [19].

Interactive GIS systems relied on platforms such as QGIS and ArcGIS, which supported dynamic map updates, spatial queries, and layer management [11]. Among these, QGIS emerged as a preferred platform for developing interactive GIS systems due to its flexibility as an open-source tool. Developers created custom Plugins tailored to specific needs, such as geospatial analysis or route optimization [21]. For example, Plugins designed for real-time data visualization and spatial analysis enhanced user engagement by enabling live updates and providing customizable outputs [55, 56].

The development of custom Plugins in QGIS typically involved using Python scripting and the QGIS Plugin Builder to address specific requirements [20]. These Plugins expanded QGIS's capabilities by incorporating interactive interfaces, real-time data updates, and advanced analytical tools. In logistics, custom Plugins enabled stakeholders to explore optimized routes, apply dynamic filters, and visualize spatial data more effectively. For instance, integrating optimization results with real-world road networks through QGIS ensured that route planning aligned with practical constraints while providing actionable insights.

Real-world examples illustrated the effectiveness of interactive GIS systems in logistics and disaster management. In transportation, GIS-based systems were employed to model traffic patterns and design adaptive routing strategies. For instance, Supriadi et al. [11] demonstrated how GIS improved congestion management by integrating road network data. Similarly, GIS systems were used to assess and visualize environmental impacts, supporting multi-objective decision-making in logistics planning [53].

This thesis built on such advancements by developing an interactive GIS system using QGIS Plugins, specifically designed to visualize and analyze the results of CIACO for the MTVRPHFTW problem. By enabling dynamic visualization, filtering options, and spatial analysis, the system enhanced the accessibility and usability of CIACO optimization outputs for logistics decision-makers.

2.3.3 Challenges and Future Directions in GIS Integration for Logistics

Despite its growing capabilities, GIS still faces challenges that limit its potential in logistics applications. Data quality and accuracy remain critical issues, as the effectiveness of GIS outputs depends heavily on the reliability of input data. Inconsistent or outdated datasets can lead to inaccurate analyses, particularly in dynamic fields such as logistics and transportation [57]. These issues are especially problematic when visualizing optimized routes or analyzing logistical metrics, where inaccuracies in customer or depot locations may compromise planning. Integrating reliable road network data and ensuring validation against real-world constraints have been identified as key strategies to address these concerns.

Interoperability between GIS platforms and optimization algorithms remains another significant challenge. The lack of standardized data formats often creates compatibility issues, particularly when transitioning between optimization systems and GIS tools [58]. Standardized formats such as GeoJSON have been recognized for their potential to improve compatibility by enabling seamless data exchange between platforms. However, their application in logistics-focused GIS systems, especially those integrating outputs from optimization models, remains underexplored.

Open-source platforms like QGIS offer significant flexibility for customizations, making them popular for logistical planning. However, they still require substantial technical expertise and support for effective use [59]. Custom plugins development in QGIS, while highly extensible through Python scripting, demands advanced programming skills that can pose a barrier for non-specialist users. Balancing user-friendly interfaces with advanced analytical features is an ongoing focus for researchers seeking to expand the accessibility of GIS systems [12].

While advancements in GIS have addressed many of these challenges, opportunities for further research and development remain. Automating data integration processes, particularly when dealing with diverse sources such as GPS and real-time traffic data, could enhance the scalability and practicality of GIS systems. Expanding these systems to incorporate real-time environmental and traffic information would significantly improve their utility for dynamic logistical environments. Additionally, simplifying the development process for QGIS Plugins through more intuitive tools or interfaces could make interactive GIS systems more accessible to non-programmers. These directions represent promising avenues for future work, offering the potential to bridge existing gaps and further enhance the capabilities of GIS in logistical applications.

2.4 Conclusion

This chapter reviewed the current state of research on the VRP, including its various extensions and adaptations, optimization methods specifically related to ACO, and the integration of interactive GIS with custom-developed QGIS Plugins in logistics applications. The limitations identified in the literature include challenges in addressing load balancing, equitable workload distribution, environmental impacts, and the scalability of GIS systems for handling real-world logistics problems. Additionally, the lack of seamless integration between advanced optimization algorithms and interactive spatial analysis tools remains a significant gap.

By utilizing CIACO's clustering-based optimization capabilities and the interactive GIS developed with custom QGIS Plugins, it is possible to enhance route optimization, improve decision-making, and address complex logistical constraints more effectively.The contributions of this thesis aim to bridge these gaps, offering practical tools to solve the MTVRPHFTW while enabling interactive visualization and analysis of results.

The next chapter will elaborate on the methodology and experimental setup for the proposed CIACO approach, focusing on its design, implementation, and integration with interactive GIS systems.

Chapter 3

Clustering-Based Improved Ant Colony Optimization for Multi-Trip Vehicle Routing Problem with Heterogeneous Fleet and Time Windows

3.1 Introduction

The VRP is a significant optimization challenge in logistics and transportation, focused on determining the most efficient routes for a fleet of vehicles to deliver goods to customers. While traditional VRP models primarily aim to minimize transportation costs, real-world applications present additional complexities, such as balancing workloads among drivers, managing heterogeneous fleets, and adhering to strict time windows. Addressing these complexities requires advanced approaches that ensure both operational efficiency and fairness.

VRP-solving approaches are generally categorized into exact algorithms and heuristics, with heuristics further divided into constructive and improvement methods [1]. Given that the MTVRPHFTW is an NP-Hard problem, exact algorithms are computationally infeasible for large-scale, real-world applications [13]. Improvement heuristics can be effective in practice, but they typically require an initial feasible solution, which is often challenging to obtain for most VRP variants [13]. These challenges require the development of advanced algorithms capable of efficiently managing the complexity and scale of real-world VRPs.

The main difference between the traditional VRP and the MTVRPHFTW lies in the latter's ability to handle a heterogeneous fleet with varying capacities. Unlike many studies that focus solely on vehicle weight constraints, our approach also considers the number of skids as a crucial capacity factor, offering a more comprehensive solution. Additionally, the MTVRPHFTW allows each vehicle to perform multiple trips within a specified planning horizon while ensuring that customer-specific time windows are respected [1].

Despite extensive research, gaps remain in addressing the complexities of multi-trip operations, heterogeneous fleet constraints, and balanced workloads. Traditional methods often fail to handle such diverse constraints efficiently. ACO offers distinct advantages for solving VRPs by mimicking natural optimization processes, enabling robust exploration of large solution spaces, and iteratively refining solutions through pheromone-based learning [15]. These characteristics make ACO particularly suitable for integrating constraints such as time windows and diverse vehicle capacities. By effectively integrating these strengths, the CIACO framework addresses the identified research gaps, providing a scalable solution for complex logistical challenges in industrial settings.

To address these challenges, we propose CIACO, designed specifically for solving the MTVRPHFTW. CIACO integrates clustering techniques with an enhanced ACO method. It begins by employing a modified DBSCAN algorithm (DBSCAN-Plus) to group customers into micro-clusters based on spatial proximity, demand and time windows while considering vehicle capacity constraints. These micro-clusters are then fused to ensure balanced workloads across all vehicles. The enhanced ACO algorithm optimizes vehicle routes by incorporating a return mechanism to handle multi-trip scenarios and by considering vehicle types to maximize the use of smaller vehicles.

The main objectives are to minimize the total traveling distance and time, reduce load imbalance to ensure fair workload distribution among vehicles and maximize the use of smaller vehicles. Using real-world data from a Canadian logistics company, the proposed framework demonstrates significant improvements in reducing transportation costs and balancing workloads compared to traditional methods. The approach enhances the use of smaller vehicles, contributing to overall efficiency and sustainability.

The following sections present the problem definition and discuss the foundational assumptions underlying the development of the CIACO framework. This chapter introduces the proposed methodology, detailing the integration of clustering and optimization techniques, such as DBSCAN-Plus and enhanced ACO, that form the foundation of the approach. It also addresses the handling of critical constraints like time windows, multi-trip operations, and heterogeneous fleet management while presenting improvement mechanisms designed to enhance solution quality. Finally, the chapter evaluates the proposed framework through experiments conducted on real-world logistics data, analyzing performance metrics such as travel distance, load balancing, and CO_2 emissions, and concludes with a discussion of challenges and potential areas for future work.

3.2 Problem Definition and Assumption

In this section, we define the problem of solving an MTVRPHFTW and outline the core assumptions and notations used in the formulation. This problem is a practical extension of the VRP designed to accommodate multiple trips, vehicles of varying sizes with different capacities, strict service time windows, and customers' specific preferences.

The MTVRPHFTW is modeled as a complete graph $G = (N, E)$, where:

- $N = \{n_0, n_1, n_2, \ldots, n_n\}$ represent the set of geographically distributed nodes, with n_0 denoting the depot and the remaining nodes representing customer locations.

- $E = \{(i,j)|i,j \in N\}$ is the set of edges defining the possible routes connecting these nodes.

Each customer i has a demand d_i consisting of both weight (mass of goods to be delivered) and the number of skids (pallets used for transport). Customers must be serviced within specific time windows $[t_i^{\text{start}}, t_i^{\text{end}}]$, where t_i^{start} and t_i^{end} represent the earliest and latest allowable service times for customer i, respectively. The service time at each node n_i, denoted by s_i, refers to the time required to complete the service, with s_0 specifically indicating the loading time at the depot. The travel time between nodes i and j is indicated by t_{ij}. We define the set $H^f = \{0, 1, 2, \ldots, |H^f|\}$ to represent the heterogeneous fleet types based on their capacities. In our industrial case, the fleet consists of three different vehicle types, denoted by f, where the largest truck is labelled as 1, the mid-sized truck is labelled as 2, and the smallest truck is labelled as 3. The capacity of a vehicle $v \in H$, in terms of weight and number of skids, is denoted by C_v. Additionally, each customer i may have a preference for a specific vehicle type, indicated by p_i. If customer i has no preference, p_i is set to 0. The maximum allowable working hours per day for each vehicle is represented by $T_{\max}$. Figure 3.1, adapted from [1], describes an example of the MTVRPHFTW, where vehicles of different types perform multiple trips to serve customers within their respective time windows, starting and returning to the depot after each trip.

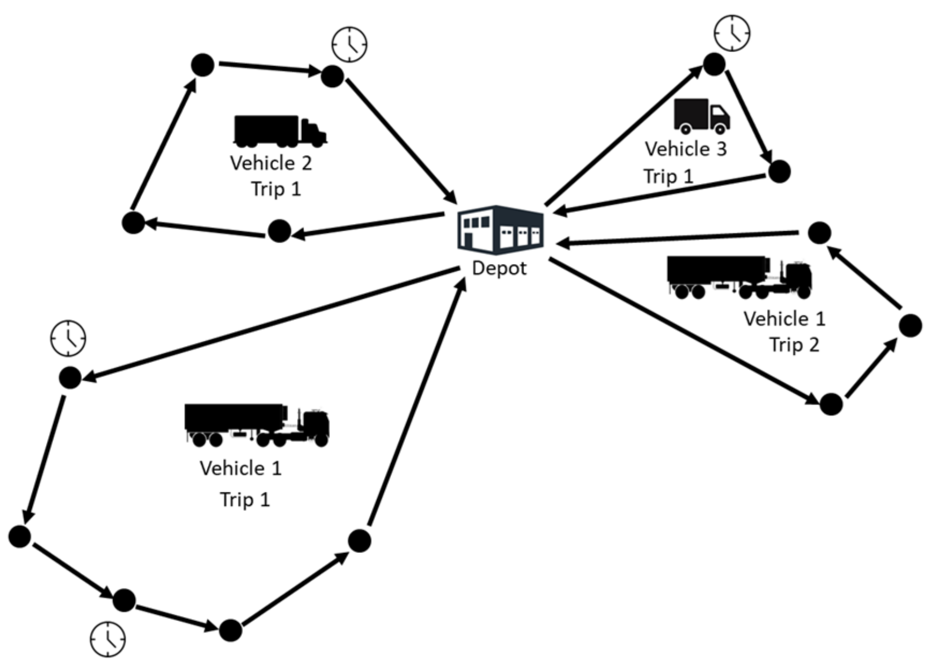

Figure 3.1: Example of MTVRPHFTW (adapted from [1])

3.3 Methodology

3.3.1 Overview

In this section, we propose the use of the CIACO algorithm for practical application in the industry for MTVRPHFTW. The approach integrates clustering techniques and enhanced ACO to achieve balanced workloads, minimize travel costs, optimize the use of smaller vehicles to reduce CO_2 emissions and adhere to time window constraints. The framework is structured into four phases: customer clustering (DBSCAN-Plus), micro-cluster fusion, and route optimization, accommodating practical logistics constraints such as multiple trips per vehicle, varying vehicle sizes and capacities, and customer-specific service time windows.

To ensure that the optimization reflects real-world travel conditions, CIACO integrates OSMnx to calculate actual travel distances rather than relying on Euclidean approximations. By using OSMnx and NetworkX's shortest-path algorithms, the framework accounts for urban constraints such as road layouts and intersections. This enhancement ensures that all phases of the optimization process, including clustering and route refinement, align with industrial logistics requirements.

In contrast to traditional ACO, which often simplifies assumptions such as uniform distances and homogeneous fleets, CIACO addresses real-world complexities. By incorporating DBSCAN-Plus for clustering, managing multi-trip operations, and handling heterogeneous fleet constraints, CIACO enhances the optimization process with structured inputs and realistic road networks, making it reliable and well-suited for industrial scenarios.

Figure 3.2 illustrates the structured phases of the CIACO framework, which systematically optimizes vehicle routing by clustering customers, merging micro-clusters, and refining routes to meet practical logistics constraints. Each phase will be discussed in more detail in Sections 3.3.2, 3.3.3, 3.3.4, and 3.3.5.

3.3.2 Clustering and Micro-Cluster Fusion

DBSCAN-Plus Clustering

To reduce computational complexity and ensure balanced workload distribution in the context of MTVRPHFTW, we developed the DBSCAN-Plus algorithm. This approach builds upon the traditional DBSCAN clustering method, previously used in various vehicle routing optimization frameworks, such as the work by Li, Fang, and Tang [60]. However, our adaptation—DBSCAN-Plus—incorporates several key modifications to better handle the specific challenges posed by MTVRPHFTW. The main procedure of DBSCAN-Plus is outlined in Algorithm 3.1.

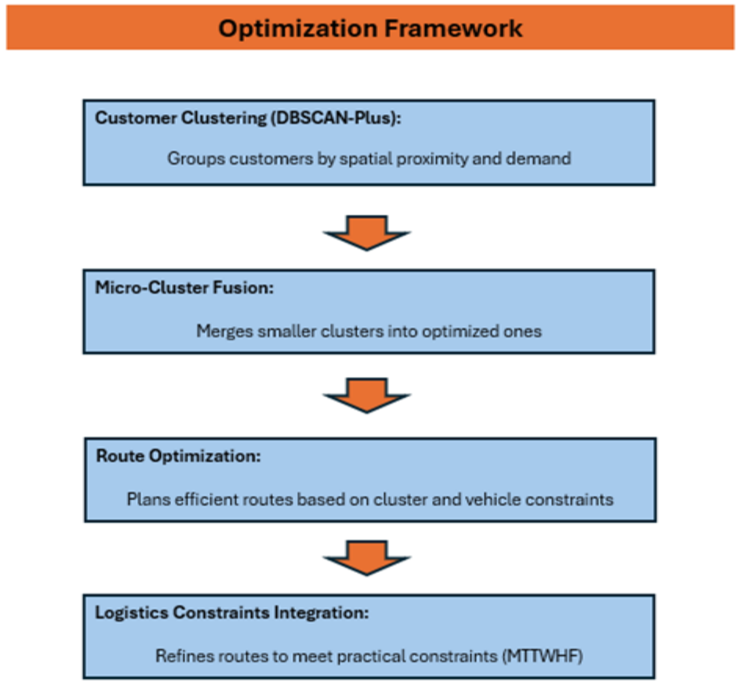

Figure 3.2: CIACO optimization framework

The DBSCAN-Plus algorithm begins by normalizing customer demands relative to vehicle capacities, as shown in Line 3 of Algorithm 1. This normalization is expressed as $D_i^{\text{normalized}} = \frac{D_i}{C}$ for each customer i, ensuring that the clustering process accurately reflects the varying demands of different customers, similar to the technique used by Li et al. [60]. Following this, DBSCAN is applied to these normalized demands *(Lines 4-5 in Algorithm 3.1)*, producing initial clusters. The total demand for each cluster k is then calculated as $\text{Total Demand}_k = \sum_{i \in \text{Cluster}_k} D_i$ *(Lines 9-10 in Algorithm 3.1)*.

While the foundational clustering process of DBSCAN-Plus is similar to that described in [60], our approach introduces specific modifications to address the complexities of multi-trip vehicle routing with a heterogeneous fleet and time windows. Unlike traditional clustering methods such as k-means, DBSCAN-Plus explicitly incorporates vehicle capacity constraints by normalizing customer demands and ensuring that each cluster remains within manageable limits. This adjustment is crucial for real-world applications like vehicle routing, where overloading vehicles would lead to infeasible solutions. Additionally, as a density-based algorithm, DBSCAN-Plus naturally forms clusters of varying sizes, accommodating unevenly distributed customer locations. These features make it particularly effective in handling the multi-trip, heterogeneous fleet scenario, where additional constraints such as strict time windows and varying vehicle capacities add significant complexity to the clustering and optimization processes.

First, Enhanced Demand Normalization *(Lines 3-5 in Algorithm 3.1)* adjusts for the presence of multiple vehicle types with differing capacities for both weight and skids. This adjustment allows the clustering process to more accurately reflect real-world constraints. Second, during the Cluster Fusion stage *(Lines 11-21 in Algorithm 3.1)*, the algorithm checks if any cluster's total demand exceeds the vehicle's capacity C. If it does, the cluster is split into smaller sub-clusters to ensure that each sub-cluster's demand remains within manageable limits.

This sub-clustering approach is specifically tailored to address the MTVRPHFTW scenario, where multiple vehicle types and constraints must be considered simultaneously. Finally, the Integration with ACO *Algorithm 3.2* uses these Micro-Clusters as input for an ACO algorithm. The structured initial conditions provided by the clustering phase help in achieving faster convergence and better optimization outcomes, enabling our approach to handle the multi-trip, heterogeneous fleet scenario more effectively.

In summary, the DBSCAN-Plus algorithm effectively breaks the problem into smaller, more manageable clusters, leading to faster convergence and consistent initial conditions for the ACO algorithm. By improving the utilization of smaller vehicles, this approach reduces the need for larger vehicles and lowers CO_2 emissions, as vehicle weight is a critical factor in emission calculations [61]. Compared to [60] and traditional DBSCAN, which may not fully account for demand (weight and skid) and time windows or efficiently merge clusters in multi-trip scenarios, Cluster-ACO offers a more robust and adaptable solution. It is specifically designed to address the complexities of the multi-trip, heterogeneous fleet scenario in ways that traditional approaches do not, making it a more comprehensive and effective method.

Micro-Cluster Fusion

Following the initial clustering phase, the Micro-Cluster Fusion process is applied to optimize the cluster configuration further. This step involves merging smaller Micro-Clusters into larger ones while ensuring that the total demand within each fused cluster does not exceed the vehicle's capacity. The fusion process not only enhances the efficiency of the delivery routes by reducing the number of trips required but also helps in minimizing the overall travel distance. By carefully managing the fusion of Micro-Clusters, the algorithm achieves a more cohesive and balanced set of customer clusters, ultimately contributing to a more efficient and cost-effective delivery operation. The main procedure of Micro-Cluster is given in *Algorithm 3.2*.

To enhance flexibility, the Micro-Cluster Fusion algorithm incorporates a mechanism to split clusters that exceed the vehicle's capacity. If a cluster's total demand surpasses this limit during the fusion phase, it is divided into smaller sub-clusters that individually satisfy the capacity constraint. This mechanism ensures feasibility for routing and vehicle assignment while addressing scenarios where initial clustering or fusion might produce overcapacity clusters.

In the Micro-Cluster Fusion process, the clusters are merged based on their total demand. For each pair of

Algorithm 3.1 DBSCAN-Plus Clustering

```
Input:
    Set of customer locations (coordinates x, y): X
    Demands: D, Vehicle capacity: C
    DBSCAN parameters: ϵ (eps) and min_samples
Output: Clusters satisfying capacity constraints: fused_clusters
Normalize demands by vehicle capacity: D_normalized = D / C
Perform DBSCAN clustering on X, weighted by D_normalized
labels ← DBSCAN(X, ϵ, min_samples, sample_weight = D_normalized)
Extract initial clusters: initial_clusters ← {label : indices of points with label in labels}
Adjust cluster indices: adjusted_clusters ← Adjust indices of initial_clusters by adding 1
Initialize empty list: fused_clusters ← []
for each cluster in adjusted_clusters do
    Compute total demand for the current cluster: total_demand ← Σ_{i∈cluster} D[i]
    if total_demand exceeds C then
        Sort cluster by demand: sorted_cluster ← Sort(cluster)
        Initialize sub-cluster: sub_cluster ← [], sub_cluster_demand ← 0
        for each customer i in sorted_cluster do
            if i can be added without exceeding capacity then
                Add i to sub_cluster: sub_cluster ← sub_cluster ∪ {i}
                Update demand: sub_cluster_demand ← sub_cluster_demand + D[i]
            else
                Add sub-cluster to fused_clusters: fused_clusters ← fused_clusters ∪ sub_cluster
                Start new sub-cluster with i: sub_cluster ← {i}, sub_cluster_demand ← D[i]
            end if
        end for
        if sub_cluster is not empty then
            Add remaining sub-cluster: fused_clusters ← fused_clusters ∪ sub_cluster
        end if
    else
        Add entire cluster: fused_clusters ← fused_clusters ∪ cluster
    end if
end for
Return fused_clusters
```

clusters considered for merging, the combined demand is calculated as Combined Demand $= \text{Total\_Demand}_{\text{current}} + \text{Total\_Demand}_{\text{candidate}}$ *(Lines 10-11 in Algorithm 3.2)*. This merger occurs only if the combined demand does not exceed the vehicle's capacity (Combined Demand $\leq C_{\max}$) *(Line 11 in Algorithm 3.2)*. After merging, the demand for the new cluster is updated to $\text{Total\_Demand}_{\text{new cluster}} = \text{Total\_Demand}_{\text{current}} + \text{Total\_Demand}_{\text{candidate}}$ *(Lines 12-16 in Algorithm 3.2)*, ensuring that the resulting clusters are both efficient and within capacity limits.

3.3.3 Route Optimization

IACO Environment Setup and Update

The Environment Setup and Update algorithm is a critical component of the ACO process, ensuring that

Algorithm 3.2 Microcluster-Fusion

```
Input:
    List of fused clusters fused_clusters
    Demands of each cluster demands
    Maximum vehicle capacity C_max
Output: Final list of fused microclusters final_fused_clusters

Initialize an empty list for fused microclusters: final_fused_clusters ← []
Sort fused_clusters in descending order based on their total demand
while there are fused clusters left to process do
    Select the largest fused cluster current_cluster from fused_clusters
    Remove current_cluster from fused_clusters
    Initialize a new cluster new_cluster with current_cluster
    Set current_demand as the total demand of current_cluster
    for each candidate_cluster in fused_clusters do
        Calculate candidate_demand as the total demand of candidate_cluster
        if current_demand + candidate_demand ≤ C_max then
            Merge candidate_cluster into new_cluster
            Update current_demand ← current_demand + candidate_demand
            Remove candidate_cluster from fused_clusters
        end if
    end for
    Add new_cluster to final_fused_clusters
end while
Return final_fused_clusters
```

the ants' environment is properly initialized and maintained throughout the optimization. This algorithm aims to set up the initial pheromone levels and heuristic information, which guide the ants in constructing their solutions. Additionally, the algorithm includes mechanisms for updating the pheromone trails based on the quality of solutions found and recalculating transition probabilities to reflect changes in the environment. This dynamic adjustment helps in balancing the exploration and exploitation aspects of the search process [15]. The detailed steps of the Environment Setup and Update are presented in *Algorithm 3.3*.

The *Algorithm 3.3* begins by setting up the pheromone matrix τ with initial values of 1, where the diagonal elements are set to 0 to prevent self-loops *(Lines 2-3 in Algorithm 3.3)*. The heuristic matrix is computed as the inverse of the distance matrix, which uses OSMnx-based road network data to reflect actual travel distances and is normalized to a [0,1] range *(Lines 4-5 in Algorithm 3.3)*. Additionally, time window constraints and load-balancing factors are initialized, along with the time window restriction matrix *(Lines 6-7 in Algorithm 3.3)*.

The *Update-Pheromone* function *(Lines 9-13 in Algorithm 3.3)* updates the pheromone values, applying pheromone evaporation and reinforcing paths taken by high-quality solutions. The *Update-Probability* function (Lines 15-22 in *Algorithm 3.3)* recomputes the probabilities of transitioning between customers by first computing the normalization factor Z_i as the sum of pheromone and heuristic products *(Lines 17-18* *in Algorithm 3.3)*. These probabilities guide the ants in constructing their routes based on the updated pheromone and heuristic information.

Algorithm 3.3 Environment Setup and Update

Initialization:
Initialize pheromone matrix τ with ones
Set diagonal of τ to zero
Compute heuristic matrix η as inverse of distance matrix
Normalize η to [0, 1] scale using min-max normalization
Initialize time window constraints and load balancing factors
Compute time window restriction matrix
End Initialization
function UPDATEPHEROMONE($\mathcal{C}$)
    Decay existing pheromones in τ using evaporation rate
    for each solution in $\mathcal{C}$ **do**
        Update pheromones on paths taken by solution based on fitness
    end for
end function
function UPDATEPROBABILITY
    for each customer i **do**
        Compute normalization factor Z_i as:

$$Z_i = \sum_{j=1}^{n} \tau[i,j] \cdot \eta[i,j]$$

        for each customer j **do**
            Calculate probability of moving from i to j using:

$$P[i,j] \leftarrow \frac{\tau[i,j] \cdot \eta[i,j]}{Z_i}$$

        end for
    end for
end function

IACO for MTVRPHFTW

The IACO for MTVRPHFTW with Time Window Penalty algorithm is designed to iteratively optimize solutions for the MTVRPHFTW. This algorithm applies the principles of ACO to find the most efficient routing solutions that minimize travel costs while meeting operational constraints, such as vehicle capacities and customer time windows. To handle violations of time constraints, a time window penalty is applied to the fitness of each solution, guiding the search toward feasible and optimal routes. The main steps of this process are outlined in *Algorithm 3.4*.

The ACO algorithm simulates the behavior of an ant colony in searching for food sources by depositing pheromones along paths, which in turn influences the decisions of other ants [15]. In this context, the set of ants is denoted by A, where each ant $\alpha \in A$ constructs a solution by iteratively selecting the next node to visit, forming a sequence of nodes $i \in N$ that begins and ends at the depot.

After all ants have constructed their solutions, the pheromone levels on the paths are updated through both evaporation and deposition processes. As shown in equation 3.1, the pheromone level $\tau_t(i,j)$ on the path from the node i to node j at iteration t decreases by an evaporation rate ρ (where $0 < \rho < 1$). It is also increased by the amount of pheromone deposited by each ant $a \in A$, denoted as $\Delta\tau_t^a(i,j)$, which is calculated as the inverse of the length of the solution L_a ($\tau_t = \frac{1}{L_a}$) [1, 15].

$$\tau_t(i,j) = (1-\rho) \cdot \tau_{t-1}(i,j) + \sum_{a \in A} \Delta\tau_t^a(i,j) \tag{3.1}$$

The ants determine the next node to visit based on a transition probability that considers both the pheromone level and the proximity (or visibility) of the nodes [1, 15]. The visibility from node i to j is represented as h_{ij}^b, where h_{ij} denotes the visibility between nodes i and j, and β is a constant reflecting the importance of visibility. The set of feasible neighbor nodes that can be visited from node i is denoted by N_i. The probability of transitioning from node i to node j at iteration t is calculated as shown in equation 3.2 [1]:

$$P_t(i,j) = \frac{\tau_t(i,j)^\alpha h_{ij}^\beta}{\sum_{j \in N_i} \tau_t(i,j)^\alpha h_{ij}^\beta} \tag{3.2}$$

Algorithm 3.4 begins by initializing the ACO environment (env) and an empty colony to store the solutions *(Lines 3-4)*. Each $\alpha \in A$ is assigned an initial solution using the 'InitializeSol' function, and the best solution is identified and recorded *(Lines 5-8)*.

In the main loop (Lines 10-25), the algorithm iteratively constructs solutions for each ant by generating routes using the 'AntMovement' function, as detailed in *Algorithm 3.5*. The fitness of each solution is evaluated, incorporating a time window penalty to account for any violations *(Lines 15 in Algorithm 3.4)*. The colony is updated by replacing the worst solution with a better one if a superior solution is found *(Lines 16-18 in Algorithm 3.4)*. Pheromone levels are adjusted based on the quality of the solutions, influencing decisions in subsequent iterations *(Lines 20-21 in Algorithm 3.4)*. If a new best solution is discovered, it replaces the current best solution *(Lines 22-23 in Algorithm 3.4)*. *Algorithm 3.5* is critical in this process, as it determines the route construction for each ant. It considers the current environment, including pheromone levels, heuristic information, and the constraints on load and time. The algorithm ensures that the ants construct feasible routes that respect the vehicle capacity and time window constraint while also allowing for the return to the depot when necessary. The process continues until the termination criteria are met, at which point the algorithm returns the best solution found *(Line 26 in Algorithm 3.4)*.

Algorithm 3.4 IACO for MTVRPHFTW with Time Window Penalty

```
Data: MTVRPHFTW instance
Result: Best solution
Initialize env    // (ACO environment) Initialize pheromone matrix, heuristic matrix, and other parameters
colony ← []    // Collection of solutions and their fitness
for each α ∈ 𝒜 do
    sol ← InitializeSol()
    colony.append(sol)
end for
best ← colony[argmin(colony.fitness)] // Track the best solution
for each iteration do
    for each α ∈ 𝒜 do
        sol ← InitializeSol()    // Reinitialize solution for each ant
        sol.route ← AntMovement(env) // Construct solution routes
        sol.fitness ← EvaluateSol(sol.route)
        Calculate time window penalty for sol.route
        sol.fitness ← sol.fitness + time_window_penalty // Apply the penalty
        Sort colony by fitness    // Sort colony to maintain the best solutions
        colony[worst] ← sol    // Replace the worst solution if the new solution is better
    end for
    Update env.pheromone    // Update pheromone levels based on current solutions
    Update env.probability    // Update transition probabilities for the next iteration
    if min(colony.fitness) < best.fitness then
        best ← colony[argmin(colony.fitness)] // Update the best solution found
    end if
end for
return best // Return the best solution found
```

3.3.4 Constraints Handling

The MTVRPHFTW involves complex operational constraints that must be carefully managed to produce feasible and optimized solutions. This section outlines how these constraints are handled within the proposed framework, mainly focusing on the AntMovement function *(Algorithm 3.5)* and the initial clustering phases with DBSCAN-Plus and Micro Cluster-Fusion.

Multi-Trip Constraint

The multi-trip nature of the MTVRPHFTW problem requires efficient vehicle utilization, where each vehicle may perform multiple trips. The 'AntMovement' function in Algorithm 3.5 manages this by tracking the current vehicle index (`curr_vidx`), cumulating load (`L`) and the cumulative time (T_{total}) of the vehicle as it serves customers. After each trip, the algorithm evaluates whether to continue serving more customers or return to the depot to start a new trip based on these variables *(Lines 18-22 in Algorithm 3.5)*. This approach ensures that vehicles are optimally deployed across multiple trips to meet customer demands effectively.

Time Window Constraint

Managing time windows is critical to ensuring deliveries occur within the specified time frames. The algorithm handles this by continuously updating the cumulative time (T_{total}) for each trip and checking if the addition of a customer would violate their time window. If adding the customer would result in exceeding the maximum time allowed (max_time), the vehicle returns to the depot to start a new trip *(Lines 12-18 in Algorithm 3.5)*. This mechanism ensures that all deliveries respect the time constraints.

Heterogeneous Fleet and Capacity Constraints

The heterogeneous fleet constraint involves varying vehicle capacities and capabilities, which is addressed by selecting the most appropriate vehicle based on the cumulative load (L). As the route is constructed, the algorithm continuously checks whether the cumulative load (L) remains within the vehicle's capacity ($C_{\max}$). If a customer's demand exceeds the vehicle's capacity, the vehicle returns to the depot to begin a new trip with an updated load (L_{updated}) and time (T_{total}) *(Lines 14-18 in Algorithm 3.5)*.

The algorithm also uses the variable `curr_vtype` to ensure that only suitable customers are included in the route. If a customer does not specify a fleet type (indicated by a zero label), the algorithm defaults to the largest fleet type to prevent capacity violations *(Lines 18-19 in Algorithm 3.5)*.

This approach ensures that each vehicle is optimally utilized, matching the vehicle's capacity with the demands of the assigned route while preventing overloads. By handling the heterogeneous fleet and capacity constraints in a unified process, the algorithm maintains feasible routes that respect each vehicle's operational limits.

3.3.5 Improvement Mechanisms

Utilization of Smaller Vehicle Sizes

This strategy optimizes fleet utilization by assigning smaller vehicles to routes where feasible, potentially reducing CO_2 emissions. During the DBSCAN-Plus clustering phase, customer demands are normalized relative to vehicle capacities, enabling the identification of clusters suitable for smaller vehicles *(Lines 3-5 in Algorithm 3.1)*. The Micro-Cluster Fusion step further refines these clusters by merging smaller clusters while ensuring the total demand remains within the capacity limits of the smallest feasible vehicle *(Lines 6-13 in Algorithm 3.2)*. In the 'AntMovement' function *(Algorithm 3.5)*, the algorithm dynamically assigns the most appropriate vehicle based on the cumulative load (L) and customer requirements, prioritizing smaller vehicles when possible *(Lines 12-14 in Algorithm 3.5)*. To quantify the reduction in CO_2 emissions, the following equation 3.3 is used [61]:

$$\text{CO}_2 \text{ Emissions (Kg)} = \frac{\theta(v)}{1000} \times SL \tag{3.3}$$

Where SL is the route segment length (km), v is the average speed of the vehicle (km/h), and $\theta(v)$ is the CO_2 emission factor (g/km), as expressed in equation 3.4:

$$\theta(v) = k + av + bv^2 + cv^3 + d\frac{1}{v} + e\frac{1}{v^2} + f\frac{1}{v^3} \tag{3.4}$$

The coefficients $k, a, b, c, d, e,$ and f are derived based on the vehicle type and weight. The algorithm, therefore, optimizes route efficiency and aims to minimize environmental impact by selecting vehicles that produce lower emissions over the route segment length.

Time Window Penalty

The algorithm continuously checks for potential time window violations during route construction in the 'AntMovement' function *(Algorithm 3.5).* If adding a customer would breach their time window, a penalty is applied to the solution's fitness *(Lines 13-16 in Algorithm 3.4).* This discourages the selection of infeasible routes in future iterations, guiding the algorithm toward solutions that respect time constraints.

Load Balancing

Reducing load imbalance across vehicles is a critical improvement mechanism that the algorithm addresses through multiple stages. Initially, in the DBSCAN-Plus clustering phase, clusters are formed with careful consideration of demand distribution relative to vehicle capacities, ensuring that no vehicle is initially assigned an excessive load *(Lines 3-5 in Algorithm 3.1).* During the Microcluster-Fusion process, these clusters are further refined to maintain a balanced distribution of total demand across all vehicles, minimizing the risk of overloading any single vehicle *(Lines 5-15 in Algorithm 3.2).*

In the 'AntMovement' function, load balancing is dynamically managed by reassigning vehicles to routes based on the current cumulative load (L) and cumulative time (T_{total}). The algorithm also tracks the number of trips and the cumulative working time for each vehicle to ensure fairness among drivers. This ensures that each vehicle carries a load as close as possible to its capacity without exceeding it and that the workload (in terms of both load and time) is equitably distributed among all vehicles and drivers *(Lines 14-18 in Algorithm 3.5).* By integrating these multiple load balancing factors, the algorithm prevents overloads and promotes fairness across the fleet.

Algorithm 3.5 AntMovement

Input:
Environment variables: pheromone matrix τ, heuristic matrix η, probability matrix $\mathbf{P}$
Set of customers $\mathbf{C}$, demands $\mathbf{D}$, capacity $\mathbf{C}_{\max}$
Time matrix $\mathbf{T}$, service times $\mathbf{S}$
Maximum time window max_time
Output: Constructed route for each ant
Initialize route $\mathbf{R} \leftarrow [0]$ // Start at depot (node 0)
Initialize cumulative load $\mathbf{L} \leftarrow \mathbf{0}$
Initialize cumulative time $\mathbf{T_{total}} \leftarrow 0$
Initialize vehicle index curr_vidx $\leftarrow 0$
Initialize vehicle type curr_vtype $\leftarrow 0$
Initialize set of unvisited customers $\mathbf{U} \leftarrow \mathbf{C}$
while $\mathbf{U} \neq \emptyset$ **do**
  Calculate probabilities $\mathbf{P}(i, j)$ for each unvisited customer $j \in \mathbf{U}$
  Select next customer j based on $\mathbf{P}(i, j)$
  if $\mathbf{L} + \mathbf{D}_j \leq \mathbf{C}_{\max}$ and $\mathbf{T_{total}} + T(i, j) + S_j \leq$ max_time **then**
    Add customer j to route $\mathbf{R} \leftarrow \mathbf{R} \cup \{j\}$
    Update cumulative load $\mathbf{L} \leftarrow \mathbf{L} + \mathbf{D}_j$
    Update cumulative time $\mathbf{T_{total}} \leftarrow \mathbf{T_{total}} + T(i, j) + S_j$
    Remove j from $\mathbf{U}$
  else
    Return to depot and start a new trip
    $\mathbf{R} \leftarrow \mathbf{R} \cup \{0\}$
    Reinitialize $\mathbf{L}$ and $\mathbf{T_{total}}$
    Update vehicle index curr_vidx $\leftarrow$ curr_vidx $+ 1$
    Set curr_vtype $\leftarrow 0$ //Reset vehicle type
  end if
end while
Return route $\mathbf{R}$, vehicle indices **curr_vidx**, vehicle types **curr_vtype**

3.4 Experiments

3.4.1 Experimental Setup

In this study, experiments were conducted using 31 real-world instances provided by a logistics company in Canada, each representing a typical day of operations. The company's fleet is heterogeneous, composed of three types of vehicles with varying capacities (weight and skid), labeled as 1 (largest), 2 (mid-sized), and 3 (smallest). Additionally, some customers have specific fleet requirements, while others may have no restrictions, denoted by 0.

The vehicles operate within a fixed time window, starting at 6:00 am and ending at the depot by 6:00 pm. The loading time at the depot is set to 60 minutes, with 20 minutes allocated for servicing each customer. The trucks are assumed to travel at an average speed of 0.7 km/m, equivalent to 42 km/h.

The objective function is to minimize the total travel distance. Additionally, it aims to ensure a fair distribution of workloads among the drivers and vehicles, as measured by load imbalance, and to maximize the use of smaller vehicles, which can potentially reduce both operational costs and CO_2 emissions.

To demonstrate CIACO's effectiveness under industrial conditions, route distances were recalculated using OSMnx-based road networks rather than Euclidean distances. This step ensures that the optimization results reflect real-world travel constraints, such as urban road layouts and intersections.

Several performance metrics were used to evaluate the algorithm's practical applicability:

- Total traveling distance (km)
- Total traveling time (mins)
- Total CO2 emissions
- Number of trips per vehicle
- Number of total trips
- Number of vehicles
- Load imbalance: Measured by the standard deviation (SD) of (1) the total load delivered (TLD), (2) total travel time (TTT), and (3) route length (RL).
- Fleet composition (usage distribution across vehicle sizes)

The ACO algorithm was configured to achieve these objectives with 100 ants, 300 iterations, and a pheromone evaporation rate of 0.1. Although computation time was not the primary focus of this study, it is influenced by key algorithmic parameters such as the number of iterations, the number of ants, and the complexity of the problem instance. Increasing the number of ants or iterations proportionally increases the runtime, as more solutions are constructed and evaluated during each iteration. Similarly, larger problem instances or stricter constraints, such as tighter time windows, require additional computational effort to explore and optimize the solution space. Due to the ACO process's inherent randomness, each instance is run ten times, and the average results are reported to ensure consistency and reliability.

To validate the effectiveness of the proposed improvement mechanisms within the ACO framework, the experiments first compare these enhanced solutions with baseline solutions. Next, the algorithm's performance is compared with the actual industry solutions currently used by the company, providing a practical benchmark for evaluation.

3.4.2 Experimental Results

The experimental results, presented in Tables 3.1, 3.2, and 3.3 provide a comprehensive analysis of the performance of the CIACO, IACO, and Industry-standard approaches. Improvement computation was calculated as $\frac{\text{New Method}-\text{Baseline}}{\text{Baseline}} \times 100\%$. The main objective of the experiments was to assess improvements in total traveling distance, total traveling time, CO_2 emissions, and load balancing metrics across the different methods.

Table 3.1: Comparative Experiment Results with Improvements from Baseline (ACO, IACO, and Industry)

	ACO[b]	CIACO[a]	IACO[c]	Industry[d]	Impv.[ab]	Impv.[ac]	Impv.[ad]
Total traveling distance (km)	1429.76	1371.51	1351.23	1924.89	-4.08%	1.50%	-28.75%
Total traveling time (mins)	1753.26	1682.15	1641.23	2191.58	-4.05%	2.49%	-23.24%
Total CO2 emissions	1658.79	1584.79	1558.66	1942.63	-4.46%	1.68%	-18.42%
No. of trips per vehicle	3.55	3.22	3.45	3.51	-9.30%	-6.67%	-8.26%
No. of total trips	33.62	31	31.63	38.84	-7.79%	-1.99%	-20.19%
No. of vehicles	9.43	9.69	10.13	14.83	2.76%	-4.34%	-34.68%

Table 3.1 provides a comparative analysis between CIACO, ACO, IACO, and Industry-standard approaches across key performance metrics. CIACO demonstrates notable improvements over ACO in several areas. Specifically, CIACO achieves a reduction in total traveling distance by approximately 4.08% relative to the ACO baseline. This decrease is accompanied by a corresponding 4.05% reduction in total travel time and a 4.46% reduction in CO_2 emissions, indicating that CIACO effectively optimizes route efficiency while minimizing environmental impact.

Additionally, CIACO lowers the number of trips per vehicle and the overall number of trips by 9.30% and 7.79%, respectively, highlighting an efficient use of fleet resources. However, CIACO employs a slightly higher number of vehicles than ACO, with an increase of 2.76%. This is due to the strategy of balancing the load more evenly across the fleet, thereby preventing any single vehicle from being overburdened.

When comparing CIACO to IACO and Industry-standard approaches, further improvements are evident. While CIACO results in a 1.50% increase in total traveling distance compared to IACO, it still provides a substantial reduction of 28.75% compared to industry standards. Regarding total travel time, CIACO is 2.49% longer than IACO but remains 23.24% shorter than Industry standards, highlighting efficiency despite the slight increase compared to IACO. Similarly, CIACO records CO_2 emissions that are 1.68% higher than IACO but demonstrate a significant 18.42% reduction compared to industry-standard practices. The number of trips per vehicle decreases by 6.67% compared to IACO and by 8.26% relative to industry standards, while the total number of trips drops by 1.99% compared to IACO and 20.19% compared to industry. Moreover, CIACO reduces the number of vehicles by 4.34% compared to IACO and by 34.68% compared to industry practices.

These results emphasize CIACO's effectiveness in achieving operational and environmental benefits while closely matching the efficiency of IACO and significantly outperforming industry-standard practices. Importantly, CIACO also demonstrates a notable improvement in load balancing across routes, which is further detailed in Table 3.3.

Table 3.2 provides insights into vehicle utilization, particularly focusing on the distribution of vehicle types across CIACO, IACO, and ACO. CIACO significantly outperforms in utilizing smaller vehicles (Vehicle Type 3) with a 3.87% higher utilization compared to IACO and an 8.63% increase relative to ACO. This result is consistent with the algorithm's design, which aims to maximize the use of smaller, more fuel-efficient vehicles where feasible, potentially reducing CO2 emissions. Conversely, CIACO shows a lower utilization of larger vehicles (Vehicle Type 1), with a decrease of 30.53% compared to IACO and 38.91% compared to ACO. This reallocation of vehicle types reflects CIACO's strategy to optimize fleet usage by deploying vehicles that match the demand requirements of each route rather than over-relying on larger, less efficient trucks.

Table 3.2: Average percentage differences in vehicle utilization

Vehicle Type	CIACO[a]	IACO[b]	ACO[c]	Impv.[ab]	Impv.[ac]
1	3.14	4.52	5.14	-30.53%	-38.91%
2	12.71	13.67	14.23	-7.02%	-10.68%
3	20.14	19.39	18.54	3.87%	8.63%

Table 3.3: Comparative Experiment Results: Load Imbalance

	CIACO[a]	IACO[b]	Indus.[c]	Impv.[ab]	Impv.[ac]
TLD SD diff.	13188.82 kg	15863.16 kg	12283.22 kg	-16.86%	7.37%
TTT SD diff.	33.69 mins	60.82 mins	57.58 mins	-44.60%	-41.49%
RL SD diff.	23.36 km	50.68 km	58.76 km	-53.88%	-60.25%

Load imbalance is a critical factor that impacts driver fairness and overall fleet efficiency. Table 3.3 analyzes three specific metrics: Total Load Delivered std diff, Total Travel Time std diff, and Route length std diff. These metrics were computed based on the standard deviation of load weights, travel times, and route lengths across different routes in the CIACO algorithm. The standard deviations provide an understanding of how balanced the workload is across the fleet.

CIACO significantly reduces load imbalances compared to both IACO and Industry standards. Specifically, CIACO achieves a 16.86% improvement in the Total Load Delivered std diff compared to IACO, although it is 7.37% less balanced than the Industry. This result highlights CIACO's ability to distribute load more evenly across vehicles, reducing the likelihood of any vehicle overloading. In terms of Total Travel Time std diff, CIACO shows a substantial 44.60% reduction compared to IACO and a 41.49% reduction relative to Industry. Finally, the Route length std diff is where CIACO's performance is particularly striking, with a 53.88% improvement over IACO and a 60.25% improvement over Industry. These results confirm that CIACO improves efficiency and contributes to a more balanced workload distribution, which is crucial for driver satisfaction and fleet management.

The experimental results show that CIACO offers considerable improvements over traditional ACO, IACO, and Industry-standard methods in total traveling distance, travel time, CO_2 emissions, and load balancing. CIACO optimizes vehicle utilization and distributes loads more evenly across the fleet while reducing environmental impact, making it a highly effective solution for modern logistics. These findings highlight CIACO's potential to enhance both operational efficiency and sustainability in real-world applications.

3.5 Discussion and Challenges

The CIACO framework has demonstrated significant potential in addressing the complexities of the MTVR-PHFTW. However, certain limitations and challenges remain that could inform future research and development efforts. This section explores key challenges related to CIACO, followed by potential directions for further improvement.

3.5.1 Clustering and Micro-Cluster Fusion

The integration of DBSCAN-Plus for customer clustering and Micro-Cluster Fusion has proven effective in creating balanced initial conditions for the optimization process. However, the clustering phase introduces two key challenges:

- **Parameter Sensitivity:** The performance of DBSCAN-Plus is sensitive to parameters such as ϵ (neighborhood radius) and *min_samples* (minimum cluster size) [62, 63]. Selecting inappropriate values can result in suboptimal clusters, leading to poor workload distribution or infeasible solutions. Such clustering errors can propagate inefficiencies into subsequent ACO optimization stages, affecting the quality of route allocations and overall solution performance [62, 63].

- **Complex Constraints:** Incorporating multiple capacity constraints (e.g., weight, skids, and time windows) into clustering increases computational complexity. The need to split and merge clusters dynamically adds further overhead, particularly when managing large and heterogeneous customer datasets. Poorly formed clusters may necessitate additional iterations in the optimization phase, increasing computation times and degrading convergence efficiency.

Future research could explore adaptive parameter tuning techniques or hybrid clustering methods to improve the robustness and efficiency of this phase. However, the computational cost of implementing hybrid methods or adaptive approaches, particularly for large datasets, warrants careful consideration to maintain scalability.

3.5.2 Ant Colony Optimization Challenges

While the enhanced ACO algorithm addresses the multi-trip and heterogeneous fleet requirements, several challenges persist:

- **Convergence Speed:** ACO's iterative nature requires multiple iterations to converge on optimal or near-optimal solutions. For large instances, this can lead to extended computation times.
- **Complexity of Pheromone Updates:** Maintaining and updating pheromone trails for all edges in the solution space is computationally expensive. This becomes more pronounced as the number of customers and vehicles increases, potentially affecting the algorithm's scalability.
- **Multi-Trip Management:** Incorporating multi-trip constraints into the route optimization process adds complexity, as it requires efficient allocation of vehicles across multiple trips while adhering to time windows and capacity constraints. These constraints may conflict with other objectives, such as minimizing travel distance or balancing workloads, requiring trade-offs.

Parallel computing techniques, discussed in Section 3.5.4 could address these challenges by distributing the computational workload. However, implementing parallelism might require significant modifications to the algorithm's structure and increase resource requirements, particularly for systems with limited hardware capabilities.

3.5.3 Load Balancing and Solution Quality

CIACO demonstrates significant improvements in workload balance across vehicles; however, achieving an ideal level of load balance is constrained by practical limitations. Key challenges include:

- **Residual Imbalances in Workload Distribution:** Despite notable improvements, minor variations in metrics such as weight, travel time, and route length persist. These imbalances, while expected, highlight the inherent trade-offs in optimizing workload distribution within real-world constraints, which can still impact driver satisfaction and fleet utilization.
- **Managing Competing Constraints:** While the primary objective of CIACO is to minimize travel distance, additional considerations such as balancing workloads, maximizing the use of smaller vehicles, and reducing CO_2 emissions present significant challenges. Effectively addressing these constraints while maintaining optimization performance requires careful trade-offs. Time windows are particularly critical, as they restrict scheduling flexibility and often demand tighter route optimization to ensure timely deliveries. This issue becomes more evident when customer locations are widely dispersed or

when many customers have overlapping time frames. Similarly, vehicle capacity, including both weight and skid limits, directly affects route feasibility. Incorrectly estimating these constraints can lead to infeasible solutions and require additional iterations for resolution. Together, these constraints demand precise balancing to achieve high-quality, practical solutions for real-world logistics.

Future work could incorporate multi-objective optimization techniques to better navigate these trade-offs, though such approaches often entail higher computational complexity and require careful balancing to prevent excessive compromise across objectives.

3.5.4 Parallel Computing for Scalability

The computational challenges identified in CIACO highlight the need for scalable solutions. Parallel computing presents a promising avenue for enhancing the efficiency and applicability of the framework [64, 65]:

- **Parallelizing ACO:** Each ant's solution construction can be executed independently, making this phase naturally parallelizable. Distributing ants across multiple processors or threads can significantly reduce computation time [64, 65].

- **Clustering and Distance Calculations:** Distance matrix calculations and DBSCAN-Plus clustering can benefit from parallelization, particularly when processing large spatial datasets. However, parallelizing clustering algorithms introduces synchronization and communication overhead, which must be carefully managed to ensure net efficiency gains.

- **Road Network Integration:** Calculating shortest paths and route lengths using OSMnx can be computationally expensive. Employing parallel computing or GPU acceleration for these tasks can expedite the process but requires compatibility with existing road network libraries and infrastructure.

Implementing parallelism using frameworks such as `multiprocessing`, `Dask`, or GPU-based libraries like `CuPy` could enable CIACO to handle larger datasets and solve problems in real-time settings.

3.6 Conclusion

This chapter addresses the complexities of the MTVRPHFTW by introducing the CIACO framework, which integrates enhanced ACO with clustering techniques such as DBSCAN-Plus and Micro-Cluster Fusion. The proposed framework effectively minimizes travel distances while addressing critical operational constraints, including workload balance, capacity limitations, CO_2 emissions, and traveling time.

To enhance solution quality, the framework incorporates additional mechanisms such as dynamic vehicle allocation and improved load balancing strategies, ensuring practical applicability for last-mile logistics operations. The clustering phase provides a robust foundation by generating initial solutions that respect capacity constraints, while the optimization phase efficiently refines these solutions to achieve the primary objective of minimizing travel distance.

The results, derived from experiments on real industrial data, demonstrate the efficacy of CIACO in outperforming existing methods across key performance metrics, such as travel distance, load balance, and environmental sustainability. While the framework demonstrates strong potential, challenges remain, including the sensitivity of clustering parameters, computational overhead, and residual workload imbalances. These limitations highlight opportunities for further refinement, such as adaptive clustering methods and parallel computing.

In summary, this chapter presents CIACO as a practical and innovative solution for the MTVRPHFTW. By balancing efficiency, practicality, and sustainability, it provides a valuable foundation for advancing vehicle routing optimization in complex, real-world logistics scenarios.

Chapter 4

An Interactive GIS System for Clustering-Based Improved Ant Colony Optimization in Logistics through Custom-Developed QGIS Plugins

4.1 Introduction

Geographic Information Systems (GIS) have become powerful tools for managing, analyzing, and visualizing spatial data. Recent advancements in data availability, computing power, and geospatial technologies have significantly enhanced the role of GIS in solving spatial challenges and supporting data-driven decision-making [46]. GIS applications extend across various fields, such as urban planning, environmental management, public health, and disaster response [45]. In logistics, GIS is often used to optimize route planning by analyzing spatial data, leading to greater efficiency and reduced costs [47].

In this thesis, GIS was integrated with the CIACO algorithm to address the MTVRPHFTW. By combining optimization techniques with geospatial analysis, this approach aimed to improve decision-making in complex logistics environments. To achieve this, a custom interactive GIS system was developed using QGIS, an open-source GIS platform. This system was further enhanced by creating a set of custom plugins that allowed users to visualize optimized routes, interact with spatial data, and perform detailed analyses.

The plugins developed in this study extend traditional GIS functionalities, enabling users to explore optimized routes based on real-world road networks interactively. The system utilizes Dijkstra's algorithm for pathfinding, allowing it to generate routes based on the actual road network rather than relying on simplified Euclidean distances. The addition of custom plugins in QGIS made it possible to dynamically visualize results, filter data, and gain insights into various logistical scenarios, such as identifying routes with the shortest travel time or lowest CO_2 emissions.

Developing this interactive GIS system through custom QGIS plugins addresses several challenges in the field of logistics, where efficient route planning and real-time decision-making are critical. By visualizing complex optimization outputs in an accessible manner, the system supports users in making data-informed decisions. Additionally, the interactive features enable users to adjust filters, explore different routing scenarios, and quickly assess the impact of various strategies.

The following sections will define the problem and outline key assumptions made in developing the interactive GIS system. The chapter will cover the system's architecture, integration of optimization algorithms, and the details of implementing custom plugins in QGIS. This chapter also includes an evaluation of the system through testing, user feedback analysis, and assessment of its impact on logistics planning, concluding with a discussion of the challenges encountered during development and a summary of key findings.

4.2 Problem Definition and Assumption

4.2.1 Problem Definition

The development of an interactive GIS system through custom QGIS plugins overcomes key obstacles in integrating the optimized routes generated by the CIACO algorithm for the MTVRPHFTW. While CIACO effectively minimizes CO_2 emissions, reduces travel times, improves load balance, optimizes vehicle utilization, and reduces total travel distance, its outputs require further refinement to ensure practical usability and accurate visualization in real-world logistics applications.

A key challenge lies in the differences in how road networks are represented in the Python-based OSMnx environment versus GIS platforms like QGIS. These differences can result in misalignments, missing components, or inaccurate route displays when transitioning outputs from CIACO to QGIS. For example, routes generated using simplified Euclidean distances often fail to align with the actual road geometries required for visualization, resulting in spatial inaccuracies that complicate decision-making, as shown in Figure 4.1. In contrast, real-world road networks provide a more accurate basis for route planning by accounting for road connectivity and practical constraints. Achieving accurate alignment of CIACO-generated routes with real-world road networks is, therefore, essential for effective and reliable logistics planning.

The road network is represented as a graph $G = (N, E)$, where:

- $N = \{n_0, n_1, n_2, \ldots, n_k\}$: Represents the set of nodes, where n_0 denotes the depot and $n_1, n_2, \ldots, n_k$ represent customer locations.
- $E = \{(i, j) | i, j \in N\}$: Defines the set of edges representing road segments between nodes.

The CIACO algorithm generates optimized routes as sequences of customer nodes:

$$R_m = [n_0, n_{c_1}, n_{c_2}, \ldots, n_0]$$

where R_m represents the sequence of nodes visited by vehicle m, starting and ending at the depot n_0. These

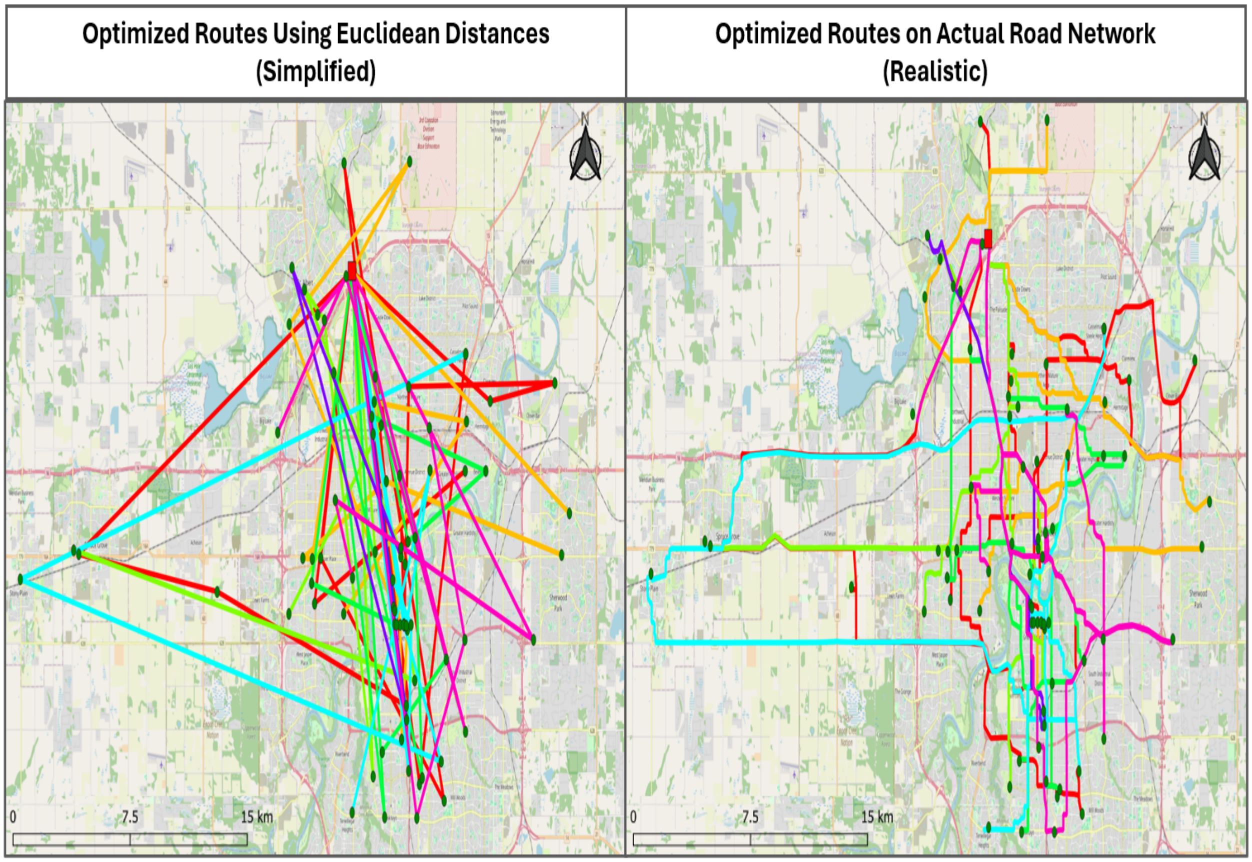

Figure 4.1: Route optimization: Euclidean distances vs. real road network alignment.

routes are initially generated using the OSMnx graph, with edge weights calculated based on travel distances and times. However, discrepancies often arise when visualizing these routes in QGIS due to differences in how spatial data is handled, necessitating recalibration for real-world applications. For instance, misalignments may occur due to variations in coordinate systems or discrepancies between road network representations in Python-based OSMnx and the QGIS platform, as shown in Figure 4.2.

Another major challenge involves transforming CIACO's outputs into GIS-compatible formats, such as GeoJSON, while addressing issues like CRS inconsistencies and format mismatches. These interoperability issues complicate the seamless integration of optimization results into GIS platforms, further limiting the utility of the outputs.

Additionally, the lack of interactivity in many existing GIS tools for VRP poses a significant barrier to their practical application. Static visualizations fail to support dynamic scenario analysis or the exploration of key logistical metrics, such as CO_2 emissions, travel times, and vehicle utilization. For logistics planners, this lack of interactivity impedes the ability to refine and adapt routes based on changing conditions or specific operational constraints.

Figure 4.2: Example of route misalignment due to environment differences between OSMnx and QGIS.

To resolve these issues, the proposed interactive GIS system, developed through custom QGIS plugins, recalibrates CIACO-generated routes using Dijkstra's algorithm. This ensures alignment with real-world road networks and validates the spatial accuracy of the outputs. The system also transforms CIACO's outputs into GIS-compatible formats, resolving interoperability challenges such as CRS inconsistencies and data formatting. Through these processes, the GIS system provides accurate visualizations of optimized routes that are practical for real-world applications.

By addressing these challenges, the interactive GIS system, implemented with tailored QGIS plugin functionality, bridges the gap between CIACO's optimization outputs and their practical implementation. It enhances usability by incorporating dynamic filtering, temporal analysis, and detailed visualization of logistical metrics, enabling planners to refine and analyze optimization results interactively. These features ensure that the system is accessible even to non-specialist users, offering a robust solution for modern logistics planning.

4.2.2 Assumptions

In developing the interactive GIS system through custom QGIS plugins, several assumptions were made to simplify and enhance the integration of the CIACO optimization algorithm with geospatial visualization:

1. **Road Network Data**: The system assumes that road network data retrieved from OSM is accurate and up-to-date. This is essential for generating realistic paths using Dijkstra's algorithm in QGIS. Any discrepancies in the OSM data could affect the accuracy of route optimization and visualization, leading to potential misalignments with actual road conditions.

2. **OSM Data Completeness**: It is assumed that the OSM data provides complete coverage of the road networks in the region of interest. The accuracy of route calculations and visualizations heavily depends on having comprehensive OSM data. Missing or incomplete road segments in the OSM database may lead to gaps in the optimized routes displayed in the GIS system.

3. **Vehicle Speed and Travel Times**: The system assumes that all vehicles travel at a consistent average speed of 42 km/h on the road networks in Edmonton. This simplification is used to calculate travel times between customer locations on the QGIS map. Factors such as traffic congestion, road conditions, or weather are not accounted for, as these are beyond the scope of the current system.

4. **Service and Loading Times**: The model incorporates fixed service times (20 minutes per customer) and loading times (60 minutes at the depot) based on historical data provided by the logistics company. These fixed times are used to estimate the total route duration visualized in QGIS, even though actual times may vary in real-world conditions.

5. **Spatial Data Compatibility**: The GIS system is designed to work primarily with GeoJSON data formats for seamless integration with QGIS. It assumes that users will provide GeoJSON files that are formatted correctly, with all required fields included. Any discrepancies in data formatting or missing attributes may result in errors during data loading and visualization.

6. **Coordinate Reference System (CRS) Consistency**: The system assumes that all spatial data (including routes and customer locations) use a consistent Coordinate Reference System (CRS). Aligning the CRS between the CIACO output (from Python) and QGIS is critical to avoid misalignments when visualizing routes on the map. This ensures that all data layers are correctly aligned, preventing route discrepancies due to projection errors.

4.3 Methodology

4.3.1 Overview

The development of the interactive GIS system in this chapter aims to utilize QGIS to visualize and analyze optimized routes generated by the CIACO algorithm for the MTVRPHFTW. This system bridges the gap between Python-based optimization outputs and geospatial visualization in QGIS, thereby enhancing the decision-making process for logistics planning.

To achieve this, a custom QGIS plugins was developed using Python and the PyQGIS library. The plugins integrates the optimized route data from the CIACO algorithm, processes it, and displays it on a real-world road network using OSM data. This method provides a more accurate representation of routes compared to simplified Euclidean distances and offers users an interactive interface to explore, analyze, and refine the generated routes.

There are five key stages in the interactive GIS system development process, including Optimization Algorithm and GIS Integration, System Architecture and Implementation, Route Correction and Pathfinding, Visualization and User Interaction, and Data Validation and Testing, as shown in Figure 4.3.

Each stage will be discussed in detail in Sections 4.3.2 through 4.3.7. The outlined stages form a cohesive framework to ensure that the developed system is both robust and efficient, enabling effective visualization and analysis of logistics data.

4.3.2 Optimization Algorithm and GIS Integration

The integration of the CIACO algorithm with QGIS enables seamless visualization of optimized routes for the MTVRPHFTW. The main objective is to transform the CIACO algorithm's output, which is initially computed in a Python environment using OSMnx, into a geospatial format that can be effectively visualized and analyzed within QGIS.

Data Preparation and Export

The CIACO algorithm generates optimized routes based on customer sequences, demands, vehicle capacities, and time window constraints. These routes are computed as lists that include customer IDs, coordinates, assigned vehicle indices, and vehicle types, representing the optimal sequence in which customers should be visited. Additional metrics—such as total route length, travel time, and CO_2 emissions—are calculated separately and integrated back into the route data. Initially stored as numerical values, these metrics are formatted into Python dictionaries and included in the final GeoJSON export. The resulting file contains

Figure 4.3: Overall workflow for interactive GIS system development in QGIS

detailed information, including customer locations, vehicle assignments, total route distance, travel time, and CO_2 emissions.

Loading Data into QGIS

Once the GeoJSON file is generated, it is imported into the QGIS environment using a custom plugins developed with PyQGIS. The plugins reads the GeoJSON data and ensures alignment with the appropriate Coordinate Reference System (CRS). Maintaining consistency in the CRS is essential to prevent misalignment between the Python-generated data and existing road network layers within QGIS. This step guarantees that customer locations, depots, and routes are accurately mapped onto geospatial layers, allowing for precise visualization on the map.

Integration with OpenStreetMap (OSM) Data

To enhance the visualization of optimized routes, the QGIS plugins integrates OSM data layers, ensuring that the routes generated by the CIACO algorithm align with actual road segments rather than relying on simplified Euclidean paths. This integration allows the routes to better reflect real-world routing scenarios.

To further refine the alignment, Dijkstra's algorithm is applied within QGIS to adjust route segments according to the actual road network [66]. This recalibration ensures that the CIACO-generated routes align accurately with real-world roads, enhancing the reliability of visualizations used in logistics planning.

However, since the CIACO algorithm solves the MTVRPHFTW problem using data from a logistics company, the recalibration process using Dijkstra's algorithm must respect the customer sequences established by CIACO. While Dijkstra's algorithm optimizes the path between each pair of consecutive nodes, it must adhere to the pre-determined sequence of customer visits. Therefore, the recalibration adjusts the actual travel paths to minimize costs while strictly preserving the optimized customer delivery order set by CIACO.

A more detailed explanation of the route correction process using Dijkstra's algorithm is provided in Section 4.3.5.

4.3.3 System Architecture

The interactive GIS system developed in this thesis is structured with a comprehensive architecture designed to optimize the visualization of routes generated by the CIACO algorithm for solving the MTVRPHFTW. This system integrates data processing, geospatial visualization, and user interaction in a cohesive and efficient manner. The architecture consists of three main components: the Data Layer, the Custom QGIS plugins for CIACO Integration, and the User Interface, as illustrated in Figure 4.4.

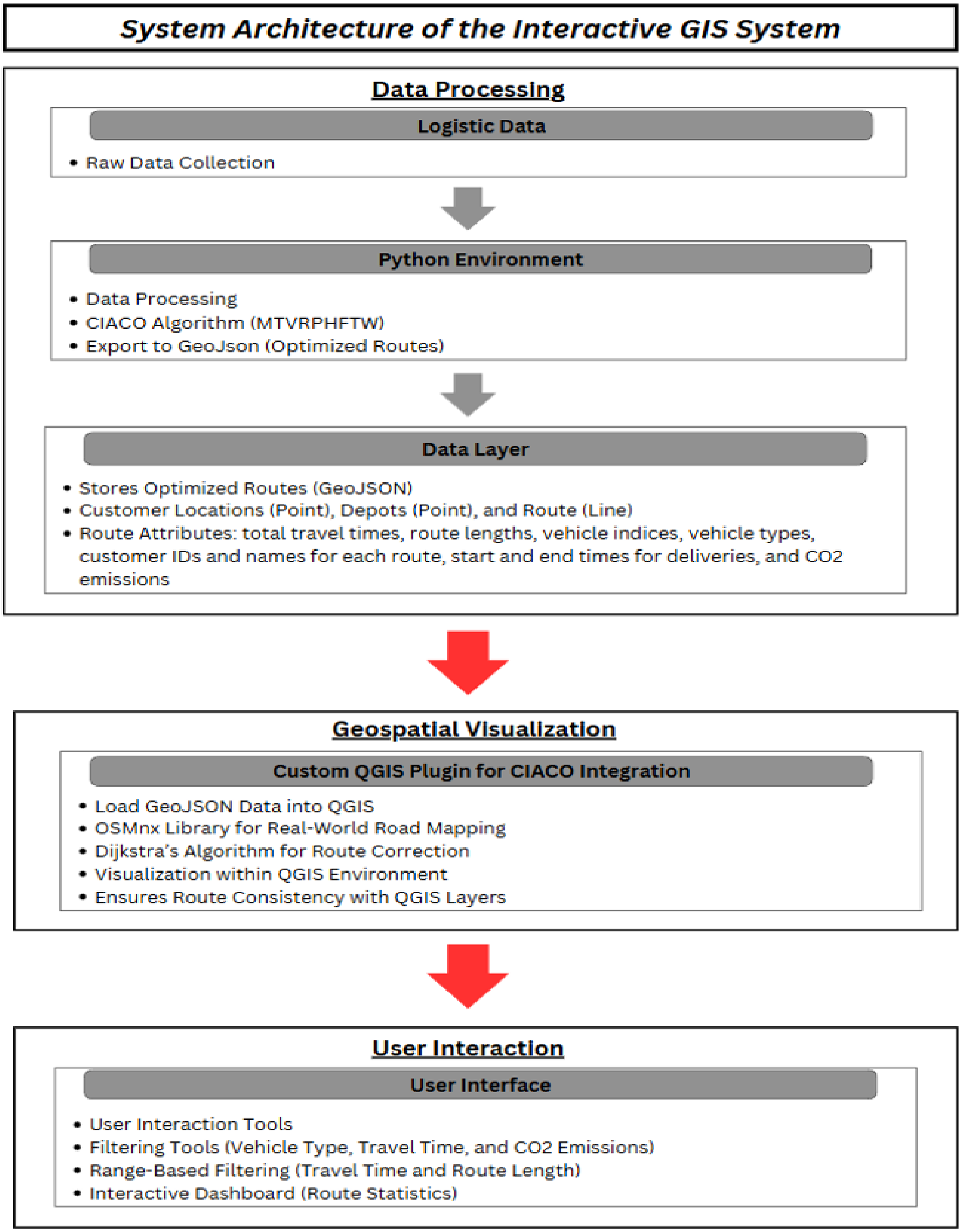

Figure 4.4: System Architecture for the Interactive GIS System Using CIACO Optimization

Data Layer

The Data Layer forms the core of the system by storing essential geospatial data, including customer locations, depots, and vehicle routes. The CIACO algorithm optimizes routes based on a combination of factors such as the sequence of customer orders, spatial proximity, customer demands, vehicle capacities, and time windows. The optimized routes are represented using three layers within QGIS: customer locations and depot points (latitude, longitude, customer IDs, and names) and route lines. The route layer includes attributes like total travel times, route lengths, vehicle indices, vehicle types, and CO_2 emissions. Additionally, it provides details on customer IDs and names for each route, along with start and end times for deliveries. This layered data structure ensures that all necessary information is available for in-depth spatial analysis and decision-making.

Custom QGIS plugins for CIACO Integration

The custom QGIS plugins for CIACO Integration component connects the optimization outputs to the geospatial system. The optimized routes, initially generated in the Python environment, are exported as GeoJSON files with the appropriate CRS. A custom QGIS plugins was specifically developed to load and visualize these routes within QGIS. The system uses the OSMnx library to map the routes to real-world road networks. However, if some routes do not align perfectly due to discrepancies between the Python-generated data and QGIS layers, Dijkstra's algorithm is employed to adjust these routes. This recalibration ensures that the routes match actual road conditions, focusing on connectivity and realistic path mapping. While speed limits are not explicitly considered, the process ensures that routes align with existing roads. The recalibration also verifies that the attributes such as total route length, travel time, and order sequence generated by CIACO remain consistent within QGIS.

User Interface

While the custom QGIS plugins for CIACO integration focuses on aligning the optimized routes with real-world road networks, the user interface component of the custom QGIS plugins provides an intuitive platform for exploring and analyzing these routes interactively.The User Interface enhances the user experience by providing interactive tools for exploring and analyzing optimized routes. The custom-developed QGIS plugins allows users to load GeoJSON files and apply filters based on criteria such as vehicle type, travel time, route length, and CO_2 emissions. Users can also specify ranges for travel times and route lengths to filter routes that fall within the desired parameters. The system's interface supports filtering by allowing users to type specific ranges, enabling a more precise selection of routes for analysis. Additionally, the interactive dashboard displays detailed statistics for each route, including total route length, total travel time, CO_2

emissions, and the number of deliveries completed within each route.

This level of interactivity helps logistics planners quickly analyze route performance, make adjustments, and optimize logistics strategies based on the system's dynamic analysis capabilities. Although the data used to generate the routes is derived from historical logistics records rather than real-time inputs, the system enables users to simulate real-time decision-making by adjusting parameters and exploring various scenarios interactively. This flexibility provides logistics planners with a practical tool for exploring alternative routing strategies and optimizing resource allocation.

This integrated system architecture effectively combines route optimization with geospatial visualization, improving the efficiency of logistics planning by providing a clear and interactive visualization of optimized routes. By using custom plugins and integrating advanced algorithms with QGIS, the system ensures that users can dynamically visualize, analyze, and adjust routes to suit their operational needs.

4.3.4 Implementation Details

The implementation of the GIS system involved integrating various tools and technologies to efficiently visualize the optimized routes generated by the CIACO algorithm. The system was primarily developed using QGIS, Python, and custom plugins, ensuring a seamless connection between the optimization outputs and the geospatial visualization environment. The following details outline the key components and steps taken during the implementation:

QGIS Environment

QGIS serves as the primary platform for spatial visualization and analysis. The software was chosen due to its open-source nature and extensive support for geospatial data manipulation. To display the optimized routes on actual road networks, layers for customer locations, depots, and routes are loaded into QGIS. The system ensures that all spatial data align correctly by standardizing the CRS throughout the project.

Python Scripting and CIACO Integration

The Python environment generates routes based on a combination of factors such as the sequence of customer orders, spatial proximity, customer demands, vehicle capacities, and time windows. The routes are optimized for each vehicle by considering these factors *(Line 4 in Algorithm 4.6)* After the optimization is completed, Python scripts are used to export the results in GeoJSON format, which is compatible with QGIS *(Lines 6-8 in Algorithm 4.6).*

Algorithm 4.6 Route Generation and Export to GeoJSON

Input: Set of customers C, vehicle fleet V, time windows, and capacities
Initialize parameters for CIACO algorithm
for each vehicle $v \in V$ **do**
    Optimize route R_v based on customer sequence, proximity, demand, and constraints
    Calculate attributes for R_v:
    - Total travel time t_v
    - Route length $\sum d(i,j)$ for all segments (i,j) in R_v
    - CO_2 emissions e_v

    Store optimized route R_v with calculated attributes
end for
Export all routes R to GeoJSON format
Output: GeoJSON file containing optimized routes and attributes

Development of Custom QGIS plugins

Custom plugins were developed using the PyQGIS library to enhance interactivity and data analysis within QGIS. These plugins are responsible for loading GeoJSON files and mapping the optimized routes *(Lines 1-3 in Algorithm 4.7)*. Once loaded, customer locations, depots, and routes are visualized on the map *(Lines 4-6 in Algorithm 4.7)*. Also, it allows users to interact with the data *(Line 7 in Algorithm 4.7)*.

Algorithm 4.7 Loading GeoJSON Data and Visualization via QGIS plugins

1: **Input**: GeoJSON file with routes R
2: Load layers for customer locations, depots, and routes in QGIS
3: Standardize all layers to CRS CRS_{target}
4: **for** each route R_v **do**
5:     Visualize customer points and route lines on the map
6: **end for**
7: Enable interactive filtering tools for user analysis
8: **Output**: QGIS map with loaded and visualized layers

If discrepancies are found between the optimized routes and the QGIS layers (e.g., misalignments due to environment differences), the plugins employ Dijkstra's algorithm to correct the routes *(Lines 2-13 in Algorithm 4.8)*.

Dynamic Filtering and Visualization

The custom plugins enhance the user interface by providing filtering tools to explore optimized routes interactively. The system allows users to filter routes based on criteria such as vehicle type, travel time, route length, and CO_2 emissions *(Lines 2-4 in Algorithm 4.9)*. Additionally, an interactive dashboard displays detailed statistics for each route *(Line 5 in Algorithm 4.9)*.

In summary, this comprehensive implementation ensures that the system effectively integrates route optimization with geospatial visualization, providing logistics planners with a powerful tool for dynamic

Algorithm 4.8 Route Correction Using Dijkstra's Algorithm

Input: GeoJSON layer with routes R, road network graph $G = (N, E)$, CIACO-defined sequence of customer nodes
Initialize $d(n_0) = 0$ for the depot and $d(n_j) = \infty$ for all other nodes $n_j \in N$
for each route R_v in the GeoJSON layer **do**
    for each consecutive pair of nodes (i, j) in R_v **do**
        Find the nearest nodes $n_i, n_j \in N$ on the road network G
        Apply Dijkstra's algorithm to compute the shortest path $P(i, j)$ and its distance $D(i, j)$
        if $P(i, j)$ is valid **then**
            Update the route segment (i, j) in R_v with $P(i, j)$
        else
            Report error or skip segment (i, j)
        end if
    end for
    Recalculate cumulative distance $d(n_j) = \sum_{(i,j)\in R_v} D(i, j)$
    Update timestamps for each segment based on travel times and service times
end for
Output: Corrected route layer with updated distances and timestamps

Algorithm 4.9 Dynamic Filtering and Visualization

Input: GeoJSON layer with route attributes R
Load filtering interface with options:
- Vehicle type
- Travel time t_v
- Route length $\sum d(i, j)$
- CO_2 emissions e_v

Allow user to specify range for t_v and $\sum d(i, j)$
Update map visualization based on selected filters
Display statistics on dashboard for total $\sum d(i, j)$, t_v, and e_v
Output: Filtered visualization and route statistics

analysis and decision-making.

4.3.5 Route Correction and Pathfinding

The optimized routes generated by the CIACO algorithm may require adjustments to align accurately with real-world road networks. Although CIACO utilizes OSMnx in the Python environment to incorporate actual road attributes, discrepancies can arise when visualizing these routes in QGIS. These differences may result from mismatches between the Python-based OSMnx outputs and the QGIS environment, incorrect CRS alignment, or errors during network graph generation.

Handling Discrepancies between Python and QGIS Environments

The CIACO algorithm solves the MTVRPHFTW problem by generating an optimized sequence of customer deliveries based on factors like spatial proximity, customer demands, vehicle capacities, and time windows. However, despite using real-world road data with OSMnx, the routes may not perfectly align with road networks in QGIS. For example, discrepancies can occur due to differences in spatial data handling between the two platforms, leading to route misalignments, missing road segments, or incorrectly placed customer points, which can be seen in Figure 4.2.

To address these issues, Dijkstra's algorithm is employed within a custom-developed QGIS plugins to adjust the route segments so they match the actual road network more accurately. The primary goal of this recalibration is to adjust the optimized paths generated by CIACO without altering the customer delivery sequence. This ensures that the routes remain efficient while adhering to real-world road constraints. For the details of this correction process, refer to Algorithm 4.8 in Section 4.3.4.

Application of Dijkstra's Algorithm for Route Correction

Dijkstra's algorithm was chosen for its ability to find the shortest path on weighted graphs, which is crucial for recalibrating routes based on real-world road conditions [66, 67]. The algorithm systematically explores all paths without relying on heuristics, making it particularly suitable for aligning optimized routes with road networks where weights represent distances. This ensures that the recalibration optimizes paths between consecutive customer nodes while preserving the original order determined by CIACO.

The algorithm operates by initializing the shortest known distance to the starting node (depot) as $d(n_0) = 0$, with all other nodes set to an infinite distance $d(n_j) = \infty$ *(Line 2 in Algorithm 4.8)*. A priority queue is used to process nodes, prioritized by their shortest known distance. The iterative process involves selecting the node with the minimum distance, exploring its neighbors, and updating distances if shorter paths are found *(Lines 3-12 in Algorithm 4.8)*[66].

The weight of each edge is defined as the distance between nodes:

$$w(i, j) = D(i, j) \tag{4.1}$$

where $D(i, j)$ represents the length of the edge (distance) between nodes i and j on the road network *(Line 6 in Algorithm 4.8)*.

For each neighbor n_j connected to the current node n_i, the distance $D(i, j)$ is calculated. The shortest known distance to n_j is updated as:

$$d(n_j) = \min\{d(n_j), d(n_i) + D(i, j)\} \tag{4.2}$$

(Line 5 in Algorithm 4.8). If $d(n_j)$ is updated, n_j is re-prioritized in the queue *(Lines 5-8 in Algorithm 4.8)*.

The total distance for a path is expressed as *(Line 13 in Algorithm 4.8)*:

$$d(n_j) = \min_{n_i \in \text{path}} \{d(n_i) + D(i,j)\} \tag{4.3}$$

Alternatively, it can be represented as the cumulative sum of distances along the path *(Line 13 in Algorithm 4.8)*:

$$d(n_j) = \sum_{(i,j) \in \text{path}} D(i,j) \tag{4.4}$$

This iterative process continues until all nodes have been processed, resulting in the shortest paths between consecutive nodes along the real-world road network. *(Lines 2-15 in Algorithm 4.8)*. These paths are recalibrated using the CIACO-defined sequence of customer nodes. For a detailed breakdown of the algorithm, refer to *Algorithm 4.8*.

Maintaining Delivery Sequences during Recalibration

While Dijkstra's algorithm is effective in finding the shortest path between nodes, it operates under the constraints set by the CIACO optimization. This means that the recalibration focuses solely on refining the travel paths between consecutive customers without altering the pre-optimized delivery sequence. The purpose here is not to further minimize travel times or CO_2 emissions, as these metrics have already been optimized by CIACO. Instead, the recalibration step ensures that the optimized routes align more accurately with the spatial data in QGIS.

The recalibration addresses any discrepancies that may arise due to differences between the OSMnx environment used for initial optimization and the QGIS platform used for visualization. By refining the route geometry to fit the actual road network data in QGIS, the system ensures that the optimized routes are both accurate and feasible for implementation without changing the original delivery sequence determined by CIACO.

Benefits of Route Recalibration

The recalibration process ensures that the optimized routes align correctly with the road network data in QGIS. This step focuses on correcting any discrepancies between the OSMnx-generated routes in Python and their visualization in QGIS, without altering the optimized travel times, CO_2 emissions, or costs established by CIACO. By refining the route geometry to match the spatial data in QGIS, the system ensures that the planned routes are practical for real-world execution while maintaining the efficiency metrics already optimized by CIACO. This enhances the reliability of the logistics planning process, ensuring that the visualized routes are both accurate and feasible for actual implementation.

4.3.6 Visualization and User Interaction

The QGIS plugins developed for this system provides an interactive interface that enhances the visualization and analysis of optimized routes. By integrating custom features, the plugins allows logistics planners to explore, analyze, and refine routes based on various criteria directly within the QGIS environment. This section details how the interactive elements, such as the GeoJSON loader, dynamic filtering tools, and dashboards were implemented within the custom-developed QGIS plugins to enhance visualization and improve decision-making.

Dynamic Filtering Options

The system enables users to filter routes based on specific criteria such as vehicle type, travel time, route length, and CO_2 emissions. Users can enter custom ranges for travel times or route lengths to narrow down the routes displayed on the map. This filtering capability provides flexibility, allowing users to focus on routes that meet specific operational needs, such as minimizing travel time or reducing environmental impact.

Interactive Dashboard

A key feature of the system is its interactive dashboard, which is fully integrated into the custom QGIS plugins. The dashboard displays detailed analytics for both individual routes and aggregated totals. It provides metrics such as total travel time, route length, CO_2 emissions, and the number of deliveries completed. Users can toggle between viewing data for a selected route or viewing aggregated metrics for all routes. This feature allows logistics planners to quickly assess route performance and compare different routing strategies based on key performance indicators.

The dashboard's interactivity is closely integrated with the dynamic filtering feature. Once a filter is applied, the dashboard updates in real time, displaying statistics pertinent to the selected routes based on the filter criteria. This ensures that users have immediate access to detailed information tailored to their specific needs, aiding in route optimization and more effective operational planning.

Visualization of Depot, Customer Points and Routes

The plugins enhances the visualization of customer locations, depots, and optimized routes. Customer points are displayed using distinctive markers, while routes are visualized as lines on the map. This clear differentiation helps users easily identify the sequence of deliveries, vehicle assignments, and overall route structure.

In summary, the custom-developed QGIS plugins successfully integrates both visualization and user interaction functionalities, providing logistics planners with an efficient tool for real-time analysis and decision-making. The combination of filtering, interactive dashboards, and clear visual representation of routes ensures that users can dynamically explore different routing scenarios and adjust strategies as needed.

4.3.7 Data Validation and Testing

The final phase of developing the interactive GIS system involved validating the accuracy and consistency of data outputs generated by the CIACO algorithm when visualized in QGIS. While most validation checks were performed during the development process, additional tests were conducted after all layers were loaded in QGIS to ensure the final outputs aligned correctly.

Ensuring Consistency between CIACO Outputs and QGIS

A key component of validation was ensuring that the data exported from CIACO in the Python environment was accurately visualized in QGIS. This process included checking that both platforms maintained consistent spatial data interpretation. Special attention was given to the following:

- **CRS Validation**: Ensuring accurate visualization required confirming that the CRS settings in the CIACO, which used OSMnx in Python, matched those in QGIS. Misaligned CRS settings could result in incorrect positioning of customer locations, depot points, and routes.
- **Road Network Validation**: The road network graph, generated using OSMnx in Python based on CIACO outputs, was thoroughly verified to ensure its accuracy and completeness. This included checking that edges and nodes aligned with actual roads and intersections when visualized in QGIS. Additionally, the verification process ensured that the road network graph was correctly generated without any missing components. These steps confirmed that the optimized routes adhered to real roads rather than simplified Euclidean paths, accurately reflecting real-world driving conditions.

Validation of Attribute Consistency

Another critical validation step involved ensuring that attributes such as total travel time, route length, CO_2 emissions, customer locations, customer names, the sequence of customers included in each route, vehicle indices, and vehicle types were correctly retained in the exported GeoJSON files. Debugging techniques were employed during development, including printing intermediate outputs and using status messages to confirm that these attributes were properly transferred from Python to QGIS. Additionally, it was necessary to verify that the custom-developed QGIS plugins successfully read and stored all these attributes from the GeoJSON files, ensuring that the metrics and details optimized by CIACO were accurately represented in the QGIS environment without any data loss or inconsistencies.

Comparing Euclidean-based vs. Road Network-based Routes

To evaluate the effectiveness of using actual road network data, a comparison was made between initial routes generated using Euclidean distances and those adjusted to follow real roads. By visualizing both sets of routes in QGIS, the system could compare travel times, route lengths, and overall efficiency. The corrected routes that adhered to real-world road conditions were shown to be more practical for logistics planning.

Final Testing of Interactive Features

The last stage of validation involved testing the interactive features within QGIS, such as dynamic filtering and dashboard updates. After loading all data layers, tests were performed to confirm that the filtering tools and dashboard analytics responded correctly to user inputs. This final check ensured that the system provided accurate and actionable insights for logistics planners, supporting effective decision-making.

4.4 Visualization and Evaluation of the Interactive GIS System

4.4.1 Visualization of Optimized Routes and Interactive Features

This section presents the visualizations of the optimized routes generated by the CIACO algorithm and displayed within the custom-developed QGIS plugins. These visualizations illustrate how the system maps customer locations, depots, and vehicle routes onto real-world road networks using geospatial data derived from the CIACO algorithm.

Each map highlights key route attributes, including customer sequences, total travel time, route length, and CO_2 emissions, as calculated during the optimization process. By integrating these elements into a clear and interactive visual format, the system supports more effective analysis and decision-making for logistics planners.

The following figures highlight the functionalities and insights provided by the custom QGIS plugins developed for integrating and analyzing CIACO outputs. These visualizations demonstrate the system's capability to address various operational scenarios while ensuring that the optimized routes align precisely with the real-world road network. Each figure is accompanied by an explanation of its key components and insights.

To begin, Figure 4.5 introduces the core functionalities of the custom QGIS plugins developed for CIACO visualization and analysis. These plugins enable users to load GeoJSON files, apply filters, toggle layers, and interact with dynamic dashboards for enhanced route exploration and analysis.

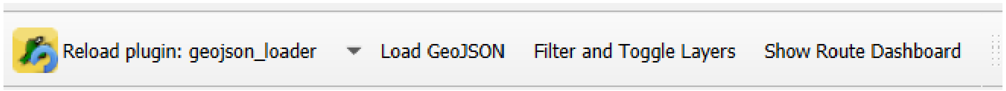

Figure 4.5: Custom QGIS plugins for CIACO visualization

Building on this foundation, Figure 4.6 demonstrates the significant improvements achieved through the custom QGIS plugins by comparing the system's capabilities before and after their implementation. The top section illustrates the limitations of routing based on simplified Euclidean distances, which fail to account for actual road connectivity and constraints, resulting in unrealistic and impractical routes. In contrast, the bottom section presents the refined and realistic route alignments achieved by integrating CIACO's outputs with real-world road network data.

The improvements evident in the bottom section include routes that accurately follow real-world road geometries, making them more applicable for practical logistics operations. Additional features, such as the dashboard and filtering tools visible in the bottom section, allow users to analyze aggregated statistics, including total route length, CO_2 emissions, travel time, and delivery counts. The filtering options enable users to tailor route visualizations based on specific parameters such as vehicle type, CO_2 emissions, route lengths, and travel times, providing more actionable and customized insights.

This comparison emphasizes the essential role of the custom QGIS plugins in bridging the gap between theoretical optimization results and their application in real-world scenarios. By offering tools for enhanced visualization and analysis, the system ensures that optimized routes align with operational constraints and adapt to a variety of logistical requirements.

Figure 4.7 further expands on these results by presenting CIACO-optimized routes mapped onto the road network. This visualization highlights key elements such as customer and depot locations, providing a clear depiction of how the optimized routes consider real-world constraints and align with practical logistics requirements.

Next, Figure 4.8 highlights the advanced filtering and toggling options provided by the QGIS plugins. These functionalities allow users to refine visualizations based on parameters such as vehicle type, CO_2 emissions, and travel time, offering tailored views of the optimized routes.

Expanding on the filtering capabilities, Figure 4.9 demonstrates a filtered visualization of routes based on specific criteria. For example, routes involving small vehicles, low CO_2 emissions, route lengths between 10–50 km, and travel times between 10–80 minutes are displayed. This targeted filtering supports precise analyses and enables planners to explore specific operational scenarios in depth.

Figure 4.10 shifts the focus to the aggregated route dashboard. This visualization presents key metrics for all routes, including total travel time, CO_2 emissions, route length, and delivery counts. By summarizing these statistics, the dashboard aids in evaluating overall route performance and logistics efficiency.

For a more detailed perspective, Figure 4.11 illustrates the interactive dashboard for an individual route (Route 7). This view provides detailed metrics such as travel time, CO_2 emissions, and delivery information, supporting an in-depth analysis of specific routes.

Finally, Figure 4.12 introduces the temporal controller interface, which allows users to visualize the progression of routes over time. By incorporating timestamps, this feature enables a time-based analysis of route schedules and operational efficiency, further enhancing the system's decision-making capabilities.

Together, these figures demonstrate the comprehensive functionality of the custom QGIS plugins developed for CIACO. They illustrate how these tools enable dynamic, tailored route analyses and enhance decision-making in logistics planning by bridging the gap between optimized outputs and real-world applications.

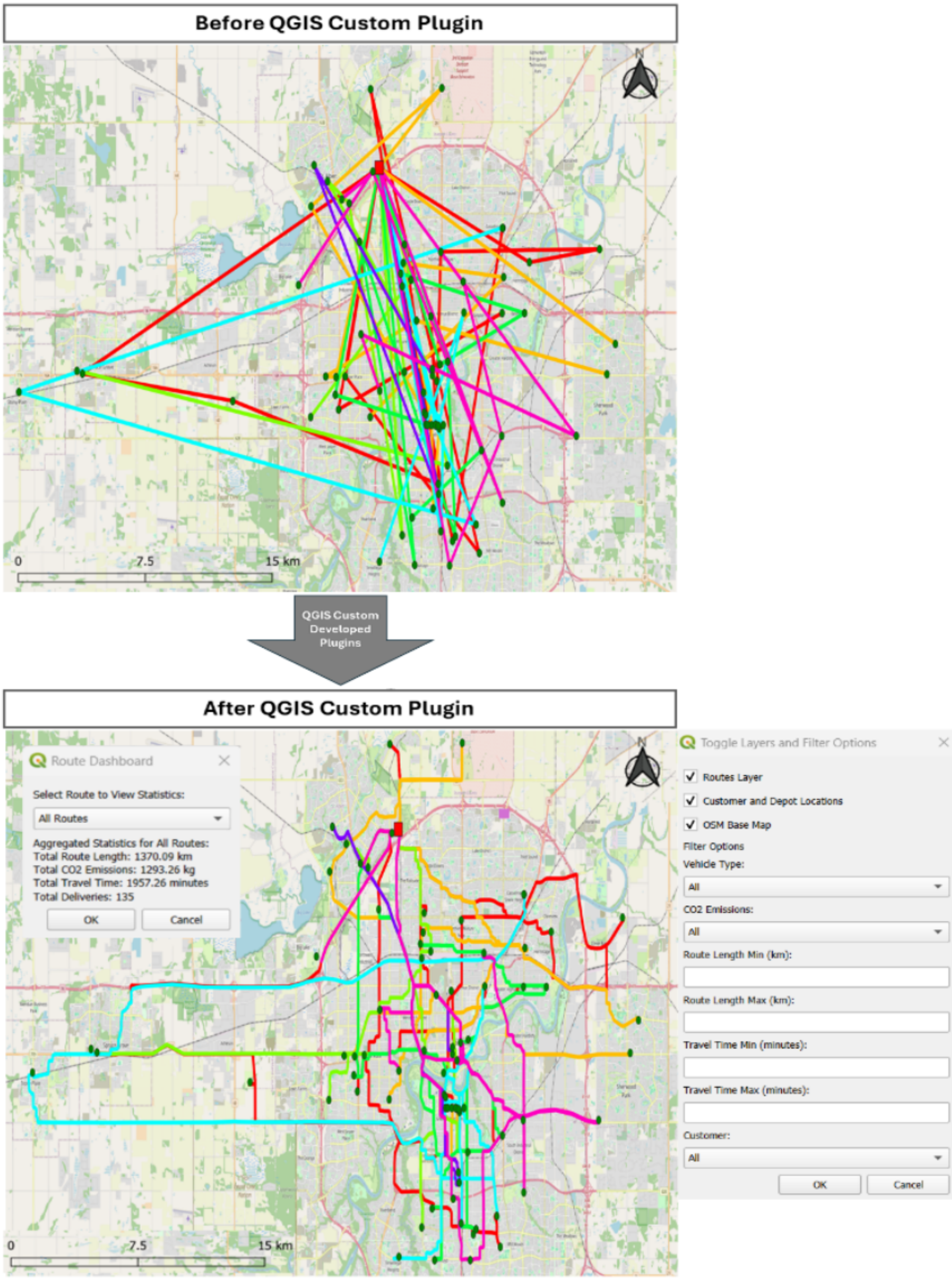

Figure 4.6: Routes before and after QGIS plugins transformation

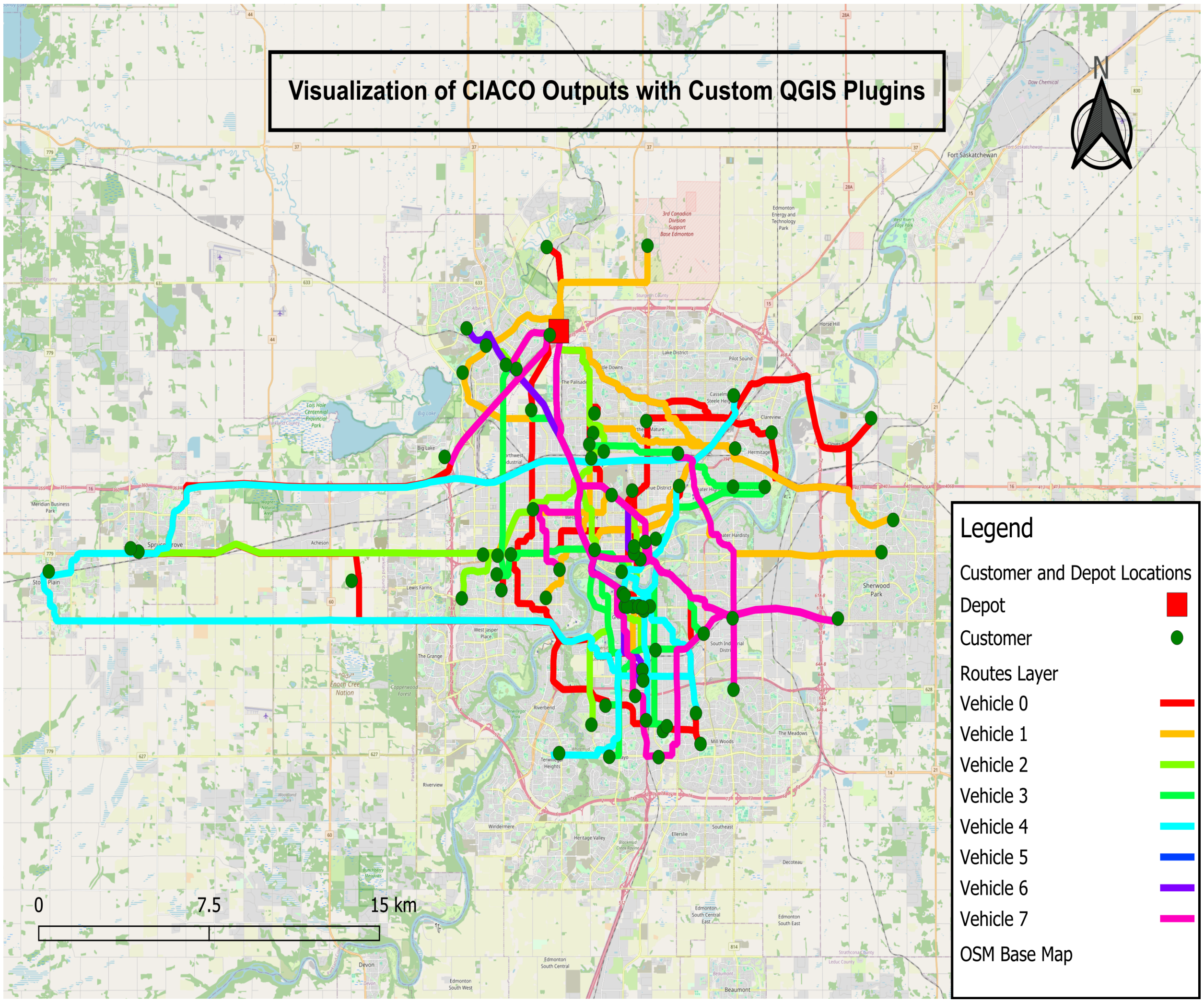

Figure 4.7: CIACO optimized routes with customer and depot locations

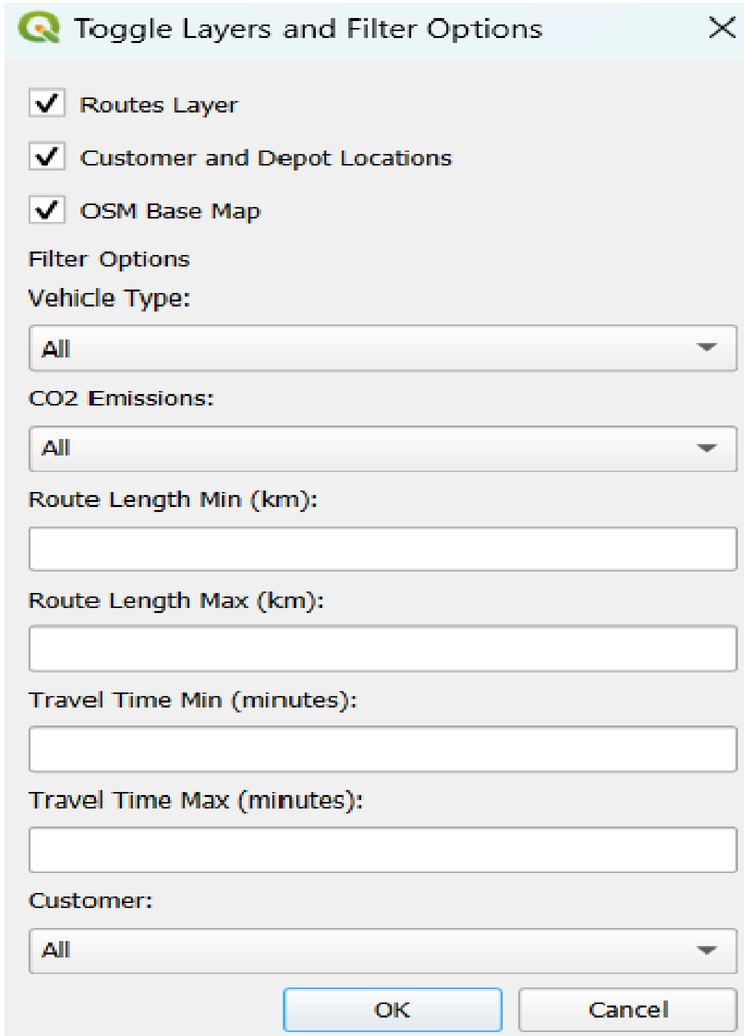

Figure 4.8: Toggle layers and filters for route analysis

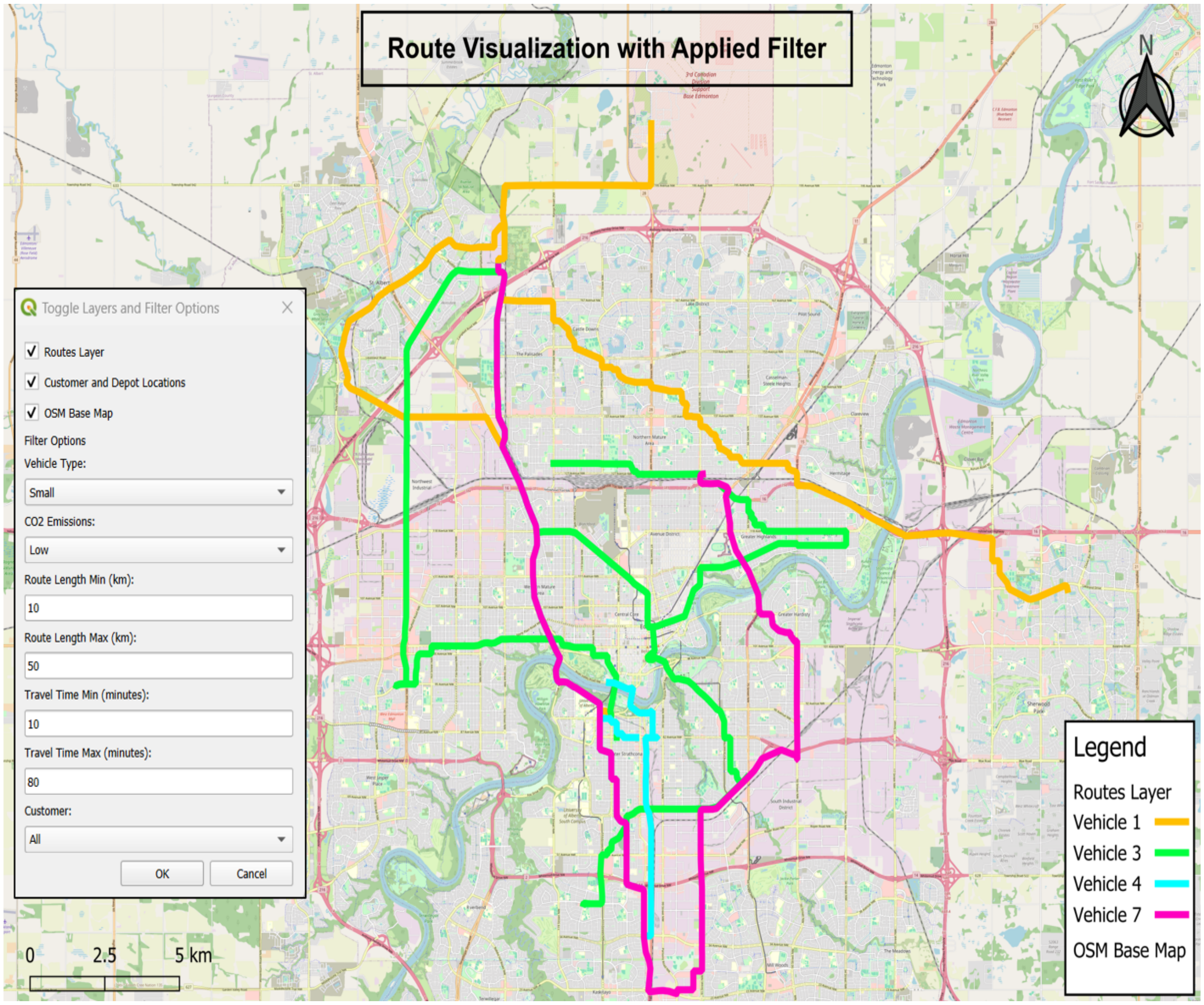

Figure 4.9: Route visualization with applied filter: small vehicles, low CO_2 emissions, route length 10–50 km, and travel time 10–80 minutes

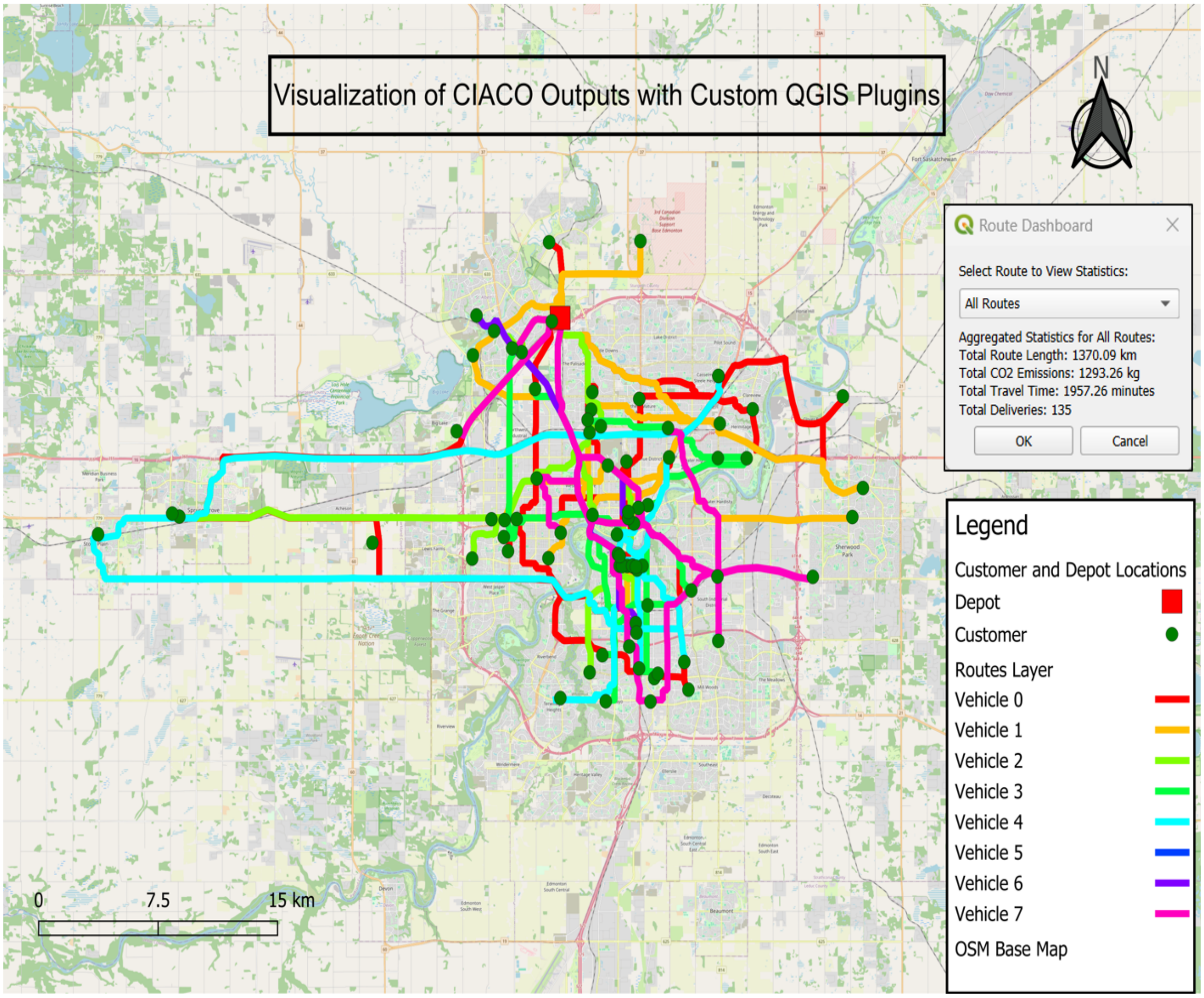

Figure 4.10: Dashboard showing aggregated route statistics

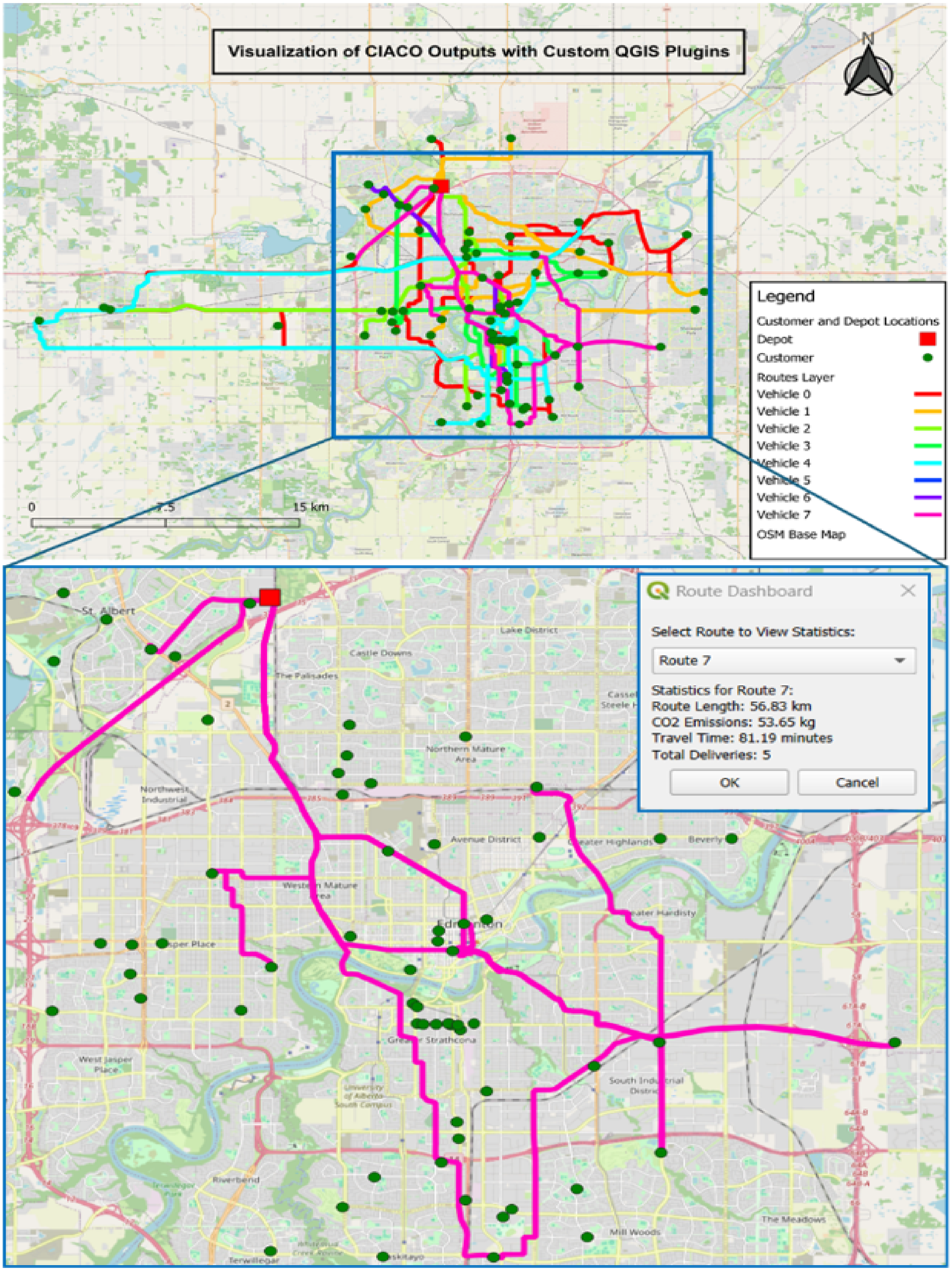

Figure 4.11: Route-specific dashboard with key metrics

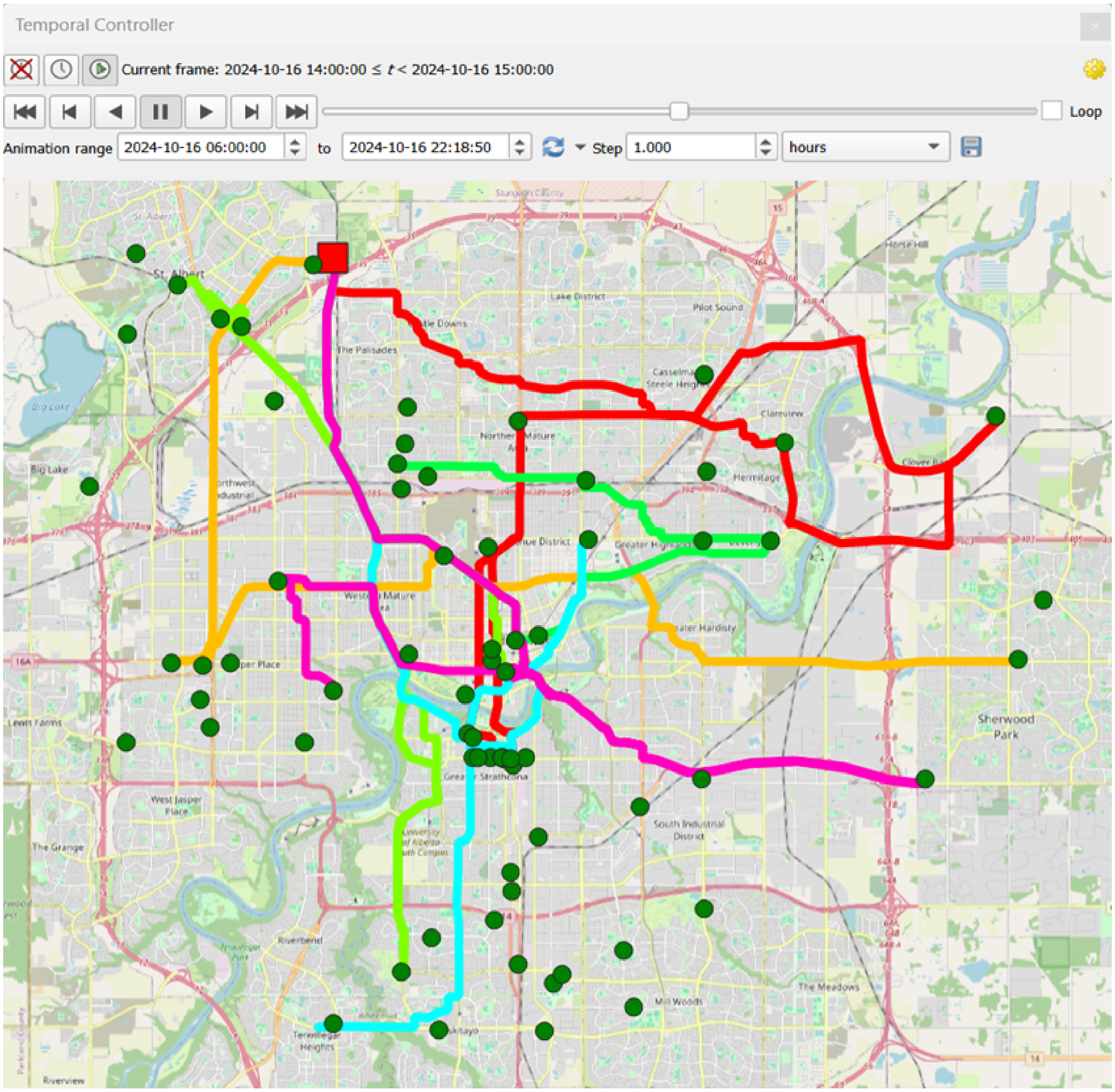

Figure 4.12: Temporal visualization of CIACO-optimized routes in QGIS activated by custom QGIS plugins

4.4.2 System Testing and Validation

The system testing and validation phase was designed to evaluate the practical applicability of the developed GIS system for optimizing vehicle routing in Edmonton, Alberta. Unlike the data validation phase described in Section 4.3, which focused on ensuring the correctness of the outputs from CIACO and their visualization in QGIS, this phase aimed to assess the system's overall functionality, usability, and ability to provide actionable insights for logistics planning.

Dataset and Tools

The GIS system was tested on a dataset consisting of 150 customer locations distributed across Edmonton, Alberta. Each customer location featured heterogeneous demand, unique time windows, and assigned vehicle types, representing real-world logistical challenges. The GeoJSON data included detailed attributes for routes, such as route lengths, CO_2 emissions, travel times, customer sequences, OSMnx network graph, customer and depot locations, time windows, vehicle indices, and vehicle types.

The testing process relied on the following key tools:

- **QGIS**: Used for visualizing CIACO outputs and utilizing custom-developed plugins for advanced filtering and analysis.
- **OSMnx and NetworkX**: OSMnx was used to extract accurate road network data from OSM and to construct the network graphs, while NetworkX provided the underlying framework for graph-based computations, including the implementation of Dijkstra's algorithm for pathfinding.
- **Python**: Responsible for processing CIACO outputs, recalibrating routes, and generating GeoJSON files for integration into QGIS.

Validation Approach

The validation process focused on assessing five key areas: customer and depot representation, route adherence to real-world networks, interactive filtering and visualization capabilities, temporal analysis, and dashboard functionality. Each area addressed specific functional aspects of the interactive GIS system to ensure it met operational requirements. Route adherence was validated using real-world road network data, while temporal analysis ensured that route timelines accurately reflected operational schedules. Interactive filtering and visualization capabilities were tested for usability and flexibility. The dashboard was validated for its accuracy in aggregating route statistics and its ability to dynamically update metrics such as route length, CO_2 emissions, travel time, and delivery count based on user-applied filters.

Customer and Depot Representation

The GeoJSON file was examined to ensure that all customer and depot locations were accurately represented. Attributes such as `Customer_ID`, `Customer_Name`, `Longitude`, and `Latitude` were verified to confirm proper alignment. The depot was uniquely identified (e.g., `Customer_ID = 1`) and visually distinguished from customer points in QGIS. All 150 locations were plotted without errors during testing, ensuring the integrity of spatial data

Route Validation

The system validated the alignment of CIACO-generated routes with real-world road networks to ensure their practicality for logistics planning. Using **OSMnx**, a network graph was created to recalibrate the routes based on actual road conditions. This process eliminated reliance on Euclidean-based paths, ensuring accuracy and feasibility. Route attributes, including route length (km), CO_2 emissions (kg), and travel time (minutes), were recalculated and cross-referenced with the GeoJSON data. All 28 optimized routes were successfully visualized in **QGIS**, and their adherence to the road network was verified without errors. The recalibration process ensures that the system delivers realistic and operationally efficient routes.

Interactive Filtering and Visualization

Custom QGIS plugins were tested for their ability to provide dynamic filtering and visualization. Users could toggle layers and apply filters based on vehicle types (`vehicle_type`), route distances, CO_2 emissions, and travel times. The plugins performed accurately, enabling tailored route analysis and interactive exploration of scenarios. The validation confirmed that these features enhanced the usability and flexibility of the GIS system.

Temporal Analysis

Time-based properties, such as `start_time` and `end_time`, were calculated using custom-developed plugins. Once the plugins loaded the GeoJSON data, they dynamically computed the `start_time` and `end_time` for each route based on the assumed working hours, loading times and service times. Additionally, the plugins activated and applied temporal properties in QGIS, enabling time-based visualization through the temporal controller. The system allowed users to examine route progression over time, including service times at customer locations and travel durations between nodes. Validation demonstrated that the temporal analysis tools functioned as intended, supporting time-sensitive logistical planning.

Validation Results

The system testing and validation phase demonstrated the following outcomes:

- **Accuracy and Reliability**: The interactive GIS system, built using custom-developed QGIS plugins, accurately represented CIACO outputs, including customer locations, depot positions, and optimized routes. All attributes of the CIACO outputs were preserved and properly stored in the QGIS environment, ensuring seamless integration and accessibility for further analysis.

- **Real-World Applicability**: All routes adhered to real-world road networks, offering practical solutions for logistics planning in Edmonton, Alberta.

- **Enhanced Usability**: Interactive filtering and temporal analysis features, implemented through custom plugins, provided users with actionable insights, enabling them to analyze scenarios and make informed decisions.

By combining these functional and analytical capabilities, the interactive GIS system, built using custom-developed QGIS plugins, proved to be a robust and versatile tool for optimizing vehicle routing. The successful validation of the system's outputs, combined with its user-friendly interface, highlights its potential to address diverse logistical challenges effectively.

4.5 Discussion and Challenges

The integration of the CIACO algorithm with the interactive GIS system, developed using custom QGIS plugins, resulted in a highly functional and versatile system for vehicle route optimization. By combining geospatial visualization with advanced optimization techniques, the system addressed key logistical challenges. However, several limitations and challenges were encountered during its development and implementation, highlighting areas for potential improvement.

4.5.1 Data and Spatial Challenges

The system's reliability depended heavily on the quality of OSM data, which occasionally included missing road segments or outdated information, leading to incomplete or misaligned routes. Additionally, discrepancies between the Python (OSMnx) and QGIS environments caused slight variations in coordinate precision and road network representation. Recalibrating routes using Dijkstra's algorithm resolved these issues but emphasized the need for tighter integration and higher-quality data sources.

Ensuring consistency in the CRS between the Python-based CIACO outputs and the QGIS environment was a recurring challenge. Small discrepancies in CRS settings often led to visual misalignments, requiring manual corrections during data loading. Additionally, in cases where nodes were located very close to each other, the choice of CRS had a noticeable impact on the precision of spatial representations, occasionally causing overlaps or distortions in route visualization. Standardizing the CRS throughout the system mitigated these issues, but it added an additional layer of complexity during data integration and recalibration.

4.5.2 Computational Overhead and Scalability

Recalculating route alignments using Dijkstra's algorithm introduced significant computational overhead, particularly for large datasets with multiple routes and complex road networks. This issue was compounded by the need to dynamically update attributes such as `start_time` and `end_time` for each route during temporal analysis.

The system's reliance on static historical data limited its ability to handle real-time updates or dynamic logistical scenarios. Scaling the system to support live traffic data or larger datasets would require substantial computational and architectural enhancements.

4.5.3 Visualization and Usability Challenges

Rendering large datasets in QGIS occasionally led to performance limitations, particularly when toggling layers, applying filters, or interacting with the temporal controller. These challenges became more evident with larger datasets or complex logistical scenarios, such as routes covering extensive geographical areas or involving high-resolution road networks. Optimizing visualization performance is crucial for effectively handling such cases and maintaining system responsiveness.

While the system provided advanced filtering, temporal analysis, and interactive dashboards, ensuring ease of use for logistics planners required user training. Inexperienced users might face challenges in applying advanced filters or interpreting results, particularly when performance limitations impact real-time interactivity. These challenges highlight the importance of user-friendly documentation, accessible training resources, and continued efforts to streamline the interface for improved usability.

4.5.4 Balancing Optimization with Practical Constraints

The CIACO algorithm focused on theoretical optimization metrics such as travel distance, travel time, and CO_2 emissions. However, practical constraints such as traffic conditions, traffic signals, area-specific speed limits, or unforeseen delays were not accounted for. Incorporating real-time traffic data, dynamic routing capabilities, and consideration of such constraints could bridge this gap and enhance the system's practicality.

4.6 Conclusion

The integration of the CIACO algorithm with the interactive GIS system, developed using custom QGIS plugins, demonstrated the feasibility and effectiveness of combining advanced optimization techniques with geospatial visualization for vehicle route planning. This approach enabled accurate representation of optimized routes, dynamic filtering, temporal analysis, and interactive visualization capabilities, directly addressing key logistical challenges in complex routing scenarios.

The custom-developed plugins played a crucial role in enabling interactivity, allowing users to toggle layers, apply filters, and analyze data dynamically. These features significantly enhanced the system's usability, making it more adaptable to various logistical needs and decision-making processes

While the system performed well in testing, challenges such as computational overhead, CRS inconsistencies, and visualization limitations highlighted areas for refinement. These limitations emphasize the need for continued development to enhance the system's usability and performance.

Future iterations of the system could incorporate real-time data integration, automate CRS alignment, and improve scalability. Additionally, simplifying user interaction through tailored training resources and more intuitive interfaces would make the system a more robust and adaptable tool for diverse logistical applications.

Chapter 5

Conclusion and Future Works

5.1 Conclusions

This thesis proposed and implemented the CIACO framework, along with an interactive GIS system developed through custom QGIS plugins, to address the MTVRPHFTW. The research aimed to optimize vehicle routes while considering the complex and practical constraints of multi-trip scheduling, heterogeneous fleets, time windows, and environmental sustainability.

At the core of the CIACO framework are DBSCAN-Plus, its micro-clustering capabilities, and an enhanced ACO algorithm, which work synergistically to address the complexities of the MTVRPHFTW. DBSCAN-Plus simplifies the problem space by forming clusters that align with demand, including weight and the number of skids, along with vehicle capacity constraints, time windows, and spatial proximity. This approach reduces computational overhead and optimally prepares the data for further optimization. Building on this foundation, the improved ACO algorithm balances solution quality, convergence speed, and scalability to optimize vehicle routes effectively. Extensive experiments using real-world data from a Canadian logistics company demonstrated CIACO's potential to reduce total travel distance, CO_2 emissions, and workload imbalance while improving vehicle utilization. These results emphasize the framework's ability to balance operational efficiency with environmental sustainability, offering practical solutions for complex logistics scenarios.

Complementing this optimization framework, an interactive GIS system was developed, incorporating custom plugins in QGIS specifically designed to connect CIACO outputs with their practical application in real-world scenarios. The system recalibrated CIACO-generated routes to align with real-world road networks using route correction mechanisms and Dijkstra's algorithm, ensuring accurate and actionable insights for logistics planners. Through the use of custom-developed QGIS plugins, this GIS system offered advanced features such as dynamic filtering, temporal analysis, and interactive dashboards. These tools allowed users to visualize optimized routes, analyze logistical scenarios, and adjust strategies interactively, empowering them to make data-driven decisions effectively.

This research makes several key contributions to the fields of optimization and GIS. It introduced CIACO, an advanced optimization algorithm designed specifically for MTVRPHFTW, incorporating innovative clustering techniques like DBSCAN-Plus with micro-clusters to simplify complex logistical problems. The algorithm also addressed workload balancing and CO_2 emissions reduction by utilizing smaller, appropriately sized vehicles during route planning. Furthermore, the study developed GeoJSON generation capabilities for seamless GIS integration and created a dynamic, interactive GIS system through custom QGIS plugins, providing logistics planners with tools to explore, validate, and implement optimization results in practical scenarios.

While the outcomes demonstrated the efficacy of the proposed framework and GIS system, several challenges emerged during the research. Parameter tuning in DBSCAN-Plus, such as neighborhood radius and minimum sample size, required careful calibration to ensure meaningful clusters. Balancing trade-offs in the ACO framework between solution quality, convergence speed, and scalability proved challenging, particularly for large datasets. Additionally, computational overheads during route recalibration and CRS inconsistencies between Python and QGIS environments posed hurdles that required iterative solutions. Despite these challenges, this research provides a robust foundation for future advancements in vehicle routing optimization and geospatial visualization.

5.2 Future Work

Building on the achievements of this research and addressing the challenges identified, several promising directions for future exploration and improvement include:

- **Real-Time Data Integration and Dynamic Optimization:** Incorporating live traffic updates, dynamic road conditions, and real-time vehicle tracking into CIACO and the GIS system would enhance their responsiveness to real-world scenarios. These improvements could enable the system to handle disruptions such as traffic congestion or weather changes and enhance dynamic route re-optimization. Extending CIACO to handle continuously changing inputs would require fundamental algorithmic adjustments while maintaining computational efficiency.

- **Advanced Clustering and Hybrid Optimization Techniques:** Expanding DBSCAN-Plus with adaptive clustering techniques that dynamically adjust parameters based on dataset characteristics could enhance cluster formation and scalability for large datasets. Integrating machine learning models to predict demand patterns or traffic conditions could further refine the clustering process. Additionally, hybrid optimization approaches, such as combining ACO with Genetic Algorithms or Simulated Annealing, could improve convergence speed and solution quality, though they may require careful tuning to balance complexity and performance.

- **Multi-Modal Transportation Networks:** Extending the framework to support multi-modal transportation networks, such as integrating electric vehicles, drones, rail, or public transit systems, would broaden its applicability. This extension could enable logistics planners to optimize routes across diverse transportation modes, addressing a wider range of operational scenarios.

- **Scalability through Parallel Computing and Cloud Deployment:** Implementing parallel computing across CIACO's clustering, optimization, and recalibration processes could significantly reduce computational overhead and improve scalability. By employing modern computing architectures, large-scale problems could be addressed more efficiently. Transitioning the system to a cloud-based platform would further enhance scalability, allowing support for larger datasets, multiple concurrent users, and seamless integration with industry-standard logistics software.

- **Enhanced Visualization and System Performance:** Improving GIS visualization capabilities is critical to effectively handling larger datasets and complex logistical scenarios. Enhancements could include:

  - Optimized rendering using tiling techniques to streamline map loading and asynchronous data processing to reduce lag during interactive operations.
  - Layered visual hierarchies to improve clarity when displaying overlapping routes.
  - Advanced analytical tools such as refined temporal analysis for detailed comparisons of route timelines and heatmaps to analyze travel patterns or congestion hotspots.

  These improvements would provide logistics planners with deeper insights while maintaining system usability and performance.

- **Usability and Accessibility:** Future iterations of the GIS system should focus on simplifying the user interface to accommodate users with varying expertise levels. Tailoring interfaces to specific user needs while maintaining access to advanced features would increase system adoption. Developing comprehensive training resources, including tutorials, interactive guides, and contextual tooltips, would further empower users to effectively utilize features like dynamic filtering, temporal analysis, and interactive dashboards.

- **Data-Driven Enhancements with Machine Learning:** Integrating machine learning models into the CIACO framework could complement its optimization capabilities by predicting traffic conditions, demand patterns, or customer preferences. These predictive models would enable more informed decision-making and improve overall system performance. The success of such enhancements would depend on the availability and quality of data and managing the computational costs associated with integrating machine learning into the optimization workflow.

- **Automation of Spatial Data Processes:** Automating spatial processes within the GIS system, such as CRS alignment and route recalibration, would streamline workflows and reduce manual intervention. Optimizing computational efficiency during these processes would be particularly important for large-scale logistical applications.

- **Broader Environmental Assessments:** Expanding the system's sustainability metrics to include factors such as noise pollution, energy consumption, and environmental degradation alongside CO_2 emissions would enable more comprehensive assessments of logistics operations. These broader metrics could support organizations in meeting sustainability goals while addressing regulatory and societal demands for greener practices.

By pursuing these future directions, the CIACO framework and GIS system can evolve into more robust, scalable, and adaptable solutions for addressing the complexities of vehicle routing and logistics planning. These enhancements would enable them to meet evolving operational and environmental demands, supporting decision-makers in optimizing logistics operations with greater efficiency, precision, and sustainability.

Bibliography

[1] J. Han, A. Mozhdehi, Y. Wang, S. Sun, and X. Wang, “Solving a multi-trip vrp with real heterogeneous fleet and time windows based on ant colony optimization: an industrial case study,” in *Proceedings of the 15th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pp. 1–4, 2022.

[2] G. B. Dantzig and J. H. Ramser, “The truck dispatching problem,” *Management science*, vol. 6, no. 1, pp. 80–91, 1959.

[3] G. Laporte, “Fifty years of vehicle routing,” *Transportation Science*, 2009.

[4] X. Cui, “Logistics cost control research of fmcg industry,” *Advanced Materials Research*, 2012.

[5] P. Novak, D. Hrušecká, and L. Macurová, “Perception of cost behaviour in industrial firms with emphasis on logistics and its costs,” *FME Transactions*, vol. 46, pp. 658–667, 2018.

[6] Y. Yu, X. Wang, R. Y. Zhong, and G. Q. Huang, “E-commerce logistics in supply chain management,” *Industrial Management & Data Systems*, 2017.

[7] A. García-Nájera and J. A. Bullinaria, “An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows,” *Computers & Operations Research*, 2011.

[8] L. Xiao, M. Dridi, A. H. E. Hassani, H. Fei, and W. Lin, “An improved cuckoo search for a patient transportation problem with consideration of reducing transport emissions,” *Sustainability*, 2018.

[9] I. Uddin, “Role of china-pak economic corridor (cpec) in logistic system of pakistan’s and china’s,” *Journal of Management Info*, 2024.

[10] W. Hansen, I. B. Hovi, and K. Veisten, “Logistics costs in norway: Comparing industry survey results against calculations based on a freight transport model,” *International Journal of Logistics Research and Applications*, 2014.

[11] A. Supriadi and T. Oswari, “Analysis of geographical information system (gis) design application in the fire department of depok city,” *Technium Social Sciences Journal*, vol. 8, p. 1–7, May 2020.

[12] D. Kurniawan, "Understanding the landscape of usability evaluation in geographic information systems: A systematic literature review," *Journal of Applied Science Engineering Technology and Education*, 2023.

[13] J.-F. Cordeau, G. Laporte, M. W. Savelsbergh, and D. Vigo, "Vehicle routing," *Handbooks in operations research and management science*, vol. 14, pp. 367–428, 2007.

[14] A. Mozhdehi, M. Mohammadizadeh, and X. Wang, "Edge-DIRECT: A Deep Reinforcement Learning-based Method for Solving Heterogeneous Electric Vehicle Routing Problem with Time Window Constraints," *Proceedings of the Canadian Conference on Artificial Intelligence*, may 27 2024. https://caiac.pubpub.org/pub/vlg4rwhi.

[15] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE transactions on systems, man, and cybernetics, part b (cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[16] N. Frías, F. Johnson, and C. Valle, "Hybrid algorithms for energy minimizing vehicle routing problem: Integrating clusterization and ant colony optimization," *Ieee Access*, 2023.

[17] A. Ezzatneshan, "A algorithm for the vehicle problem," *International Journal of Advanced Robotic Systems*, 2010.

[18] R.-M. Chen, F.-R. Hsieh, and D.-S. Wu, "Heuristics based ant colony optimization for vehicle routing problem," 2012.

[19] S. Jelokhani-Niaraki, M. Tahmoorespur, Z. Minuchehr, and M. R. Nassiri, "An ontology-based gis for genomic data management of rumen microbes," *Genomics Informatics*, 2015.

[20] H. I. El-Gamily and K. Al-Rasheed, "Deploying an interactive gis system for facility and asset management: Case study - ministry of education, kuwait," *Journal of Geographic Information System*, 2015.

[21] X. Zhang, T. Kang, X. Xie, F. Zhao, S. Tang, and H. Hu, "Research on soa-based gis service for smart distribution system," *Proceedings of the International Conference on Advanced Energy Engineering*, 2015.

[22] M. W. Savelsbergh, "Local search in routing problems with time windows," *Annals of Operations research*, vol. 4, pp. 285–305, 1985.

[23] H.-F. Wang and Y.-Y. Chen, "A genetic algorithm for the simultaneous delivery and pickup problems with time window," *Computers & industrial engineering*, vol. 62, no. 1, pp. 84–95, 2012.

[24] S. Karakatič and V. Podgorelec, "A survey of genetic algorithms for solving multi depot vehicle routing problem," *Applied Soft Computing*, vol. 27, pp. 519–532, 2015.

[25] W. Ho, G. T. Ho, P. Ji, and H. C. Lau, "A hybrid genetic algorithm for the multi-depot vehicle routing problem," *Engineering applications of artificial intelligence*, vol. 21, no. 4, pp. 548–557, 2008.

[26] B. Pan, Z. Zhang, and A. Lim, "Multi-trip time-dependent vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 291, no. 1, pp. 218–231, 2021.

[27] D. Cattaruzza, N. Absi, D. Feillet, and T. Vidal, "A memetic algorithm for the multi trip vehicle routing problem," *European Journal of Operational Research*, vol. 236, no. 3, pp. 833–848, 2014.

[28] S. Ichoua, M. Gendreau, and J.-Y. Potvin, "Vehicle dispatching with time-dependent travel times," *European journal of operational research*, vol. 144, no. 2, pp. 379–396, 2003.

[29] Y.-J. Kwon, Y.-J. Choi, and D.-H. Lee, "Heterogeneous fixed fleet vehicle routing considering carbon emission," *Transportation Research Part D: Transport and Environment*, vol. 23, pp. 81–89, 2013.

[30] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013.

[31] F. Despaux and S. Basterrech, "A study of the multi-trip vehicle routing problem with time windows and heterogeneous fleet," in *2014 14th International Conference on Intelligent Systems Design and Applications*, pp. 7–12, IEEE, 2014.

[32] A. Mozhdehi, M. Mohammadizadeh, Y. Wang, S. Sun, and X. Wang, "EFECTIW-ROTER: Deep Reinforcement Learning Approach for Solving Heterogeneous Fleet and Demand Vehicle Routing Problem with Time-Window Constraints," in *Proceedings of the 32th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '24)*, October 29–November 1 2024.

[33] F. Despaux and S. Basterrech, "Multi-trip vehicle routing problem with time windows and heterogeneous fleet," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 8, pp. 9–9, 2016.

[34] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM computing surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[35] P. N. Ky Phuc and N. L. Phuong Thao, "Ant colony optimization for multiple pickup and multiple delivery vehicle routing problem with time window and heterogeneous fleets," *Logistics*, vol. 5, no. 2, p. 28, 2021.

[36] B. Yu, Z.-Z. Yang, and B. Yao, "An improved ant colony optimization for vehicle routing problem," *European journal of operational research*, vol. 196, no. 1, pp. 171–176, 2009.

[37] S. Mazzeo and I. Loiseau, "An ant colony algorithm for the capacitated vehicle routing," *Electronic Notes in Discrete Mathematics*, vol. 18, pp. 181–186, 2004.

[38] A. Gupta and S. Saini, "An enhanced ant colony optimization algorithm for vehicle routing problem with time windows," in *2017 ninth international conference on advanced computing (ICOAC)*, pp. 267–274, IEEE, 2017.

[39] H. Wu, Y. Gao, W. Wang, and Z. Zhang, "A hybrid ant colony algorithm based on multiple strategies for the vehicle routing problem with time windows," *Complex & intelligent systems*, pp. 1–18, 2021.

[40] P. Matl, R. F. Hartl, and T. Vidal, "Workload equity in vehicle routing problems: A survey and analysis," *Transportation Science*, 2018.

[41] F. Lehuédé, O. Péton, and F. Tricoire, "A lexicographic minimax approach to the vehicle routing problem with route balancing," *European Journal of Operational Research*, 2020.

[42] M. Bender, J. Kalcsics, and A. M. Meyer, "Districting for parcel delivery services – a two-stage solution approach and a real-world case study," *Omega*, 2020.

[43] H. Abdullahi, L. S. Reyes-Rubiano, D. Ouelhadj, J. Faulín, and A. Juan, "Modelling and multi-criteria analysis of the sustainability dimensions for the green vehicle routing problem," *European Journal of Operational Research*, 2021.

[44] T. Bektaş and G. Laporte, "The pollution-routing problem," *Transportation Research Part B Methodological*, 2011.

[45] W. Musakwa, "Perspectives on geospatial information science education: An example of urban planners in southern africa," *Geo-Spatial Information Science*, 2017.

[46] P. P. Roy, "Machine learning empowered geographic information systems: Advancing spatial analysis and decision making," *World Journal of Advanced Research and Reviews*, 2024.

[47] M. Pollino, G. Fattoruso, A. B. Della Rocca, L. L. Porta, S. L. Curzio, A. Arolchi, V. James, and C. Pascale, "An open source gis system for earthquake early warning and post-event emergency management," 2011.

[48] F. M. Njiru, D. N. Siriba, and F. Karanja, "Developing a conceptual framework for gis audit in an enterprise setting," *Journal of Geoscience and Environment Protection*, 2022.

[49] S. Suhada, "Web-based tourism geographic information system in banggai laut regency," *Journal Cerita*, 2023.

[50] R. S. Afgun Usmani, M. Hashem, T. R. Pillai, A. Saeed, and A. M. Abdullahi, "Geographic information system and big spatial data," *International Journal of Enterprise Information Systems*, 2020.

[51] A. Sah, S. Suhardi, and S. Nurhayati, "Geographic information system of patient development in jayapura hospital during pandemic," *Jurnal Teknologi Dan Open Source*, 2021.

[52] D. Benatiallah, H. Hidaoui, B. Nasri, K. Bouchouicha, and A. Benatiallah, "Realization of a geographic information system for aid decision-making for forest management," *Algerian Journal of Renewable Energy and Sustainable Development*, 2020.

[53] Q. Bui and H. V. Pham, "Web-based gis for spatial pattern detection: Application to malaria incidence in vietnam," *SpringerPlus*, 2016.

[54] A. Skarlatidou, M. Haklay, and T. Cheng, "Trust in web gis: The role of the trustee attributes in the design of trustworthy web gis applications," *International Journal of Geographical Information Science*, 2011.

[55] B. Sadoun, O. Al-Bayari, J. Al-Azizi, and S. A. Rawashdeh, "Geographic information system using arcgis 10 and open source mapwindow - methodology and comparative study," *Proceedings of the International Conference on Geographic Information Systems*, 2012.

[56] B. Noviansyah, "Web gis of tourism: Voyage and forecast," *International Journal of Engineering Technology and Natural Sciences*, 2023.

[57] A. H. Liu and C. Ellul, "Quantifying geometric changes in bim-gis conversion," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-4/W2-2022, pp. 185–192, 2022.

[58] G. S. Floros, C. Ellul, and E. Dimopoulou, "Investigating interoperability capabilities between ifc and citygml lod 4 – retaining semantic information," *The International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*, 2018.

[59] F. Majidi, Y. Khosravi, and K. Abedi, "Determination of the equivalent continuous sound level (leq) in industrial indoor space using gis-based noise mapping," *Journal of Human Environment and Health Promotion*, 2019.

[60] L. Jingjing, Y. Fang, and N. Tang, "A cluster-based optimization framework for vehicle routing problem with workload balance," *Computers & Industrial Engineering*, vol. 169, p. 108221, 2022.

[61] J. Hickman, D. Hassel, R. Joumard, Z. Samaras, and S. Sorenson, "Methodology for calculating transport emissions and energy consumption," 1999.

[62] A. Karami and R. Johansson, "Choosing dbscan parameters automatically using differential evolution," *International Journal of Computer Applications*, 2014.

[63] X. Zhang and S. Zhou, "Woa-dbscan: Application of whale optimization algorithm in dbscan parameter adaption," *Ieee Access*, 2023.

[64] E. Talbi, O. Roux, C. Fonlupt, and D. Robillard, "Parallel ant colonies for the quadratic assignment problem," *Future Generation Computer Systems*, 2001.

[65] M. Randall and A. Lewis, "A parallel implementation of ant colony optimization," *Journal of Parallel and Distributed Computing*, 2002.

[66] W. Zeng and R. L. Church, "Finding shortest paths on real road networks: the case for a," *International journal of geographical information science*, vol. 23, no. 4, pp. 531–543, 2009.

[67] X. Yin, P. Cai, K. Zhao, Q. Zhou, and D. Yao, "Dynamic path planning of agv based on kinematical constraint a* algorithm and following dwa fusion algorithms," *Sensors*, vol. 23, no. 8, p. 4102, 2023.

Appendix: Publications During the Program

This appendix lists the publications produced during the program:

1. R. A. Bahrehdar, B. S. S. Kim, R. S. Ramhormozi, Y. Wang, S. Sun, and X. Wang, "The Impact of Extreme Weather Conditions on the Transportation Systems in Calgary," in Proceedings of the Transportation Association of Canada (TAC) 2023 Conference, Conference Abstract, 2023.

2. B. S. S. Kim, R. S. Ramhormozi, and X. Wang, "Quality Assessment of the OpenStreetMap Road Network in Calgary, Alberta," in Proceedings of the International Cartographic Association (ICA) 2023 Conference, Conference Abstract, 2023.

3. B. S. S. Kim, J. Han, A. Mozhdehi, X. Wang, Y. Wang, and S. Sun, "Ant Colony Optimization Approaches for a Multi-Trip Vehicle Routing Problem with Heterogeneous Fleet and Time Windows: An Industrial Case Study," in Proceedings of the National Research Council Canada (NRC) AI4L 2023 Conference, Conference Abstract, 2023.

4. B. S. S. Kim, A. Mozhdehi, Y. Wang, S. Sun, and X. Wang, "Clustering-Based Enhanced Ant Colony Optimization for Multi-Trip Vehicle Routing Problem with Heterogeneous Fleet and Time Windows: An Industrial Case Study," in Proceedings of the 17th ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS'24), pp. 1–10, Oct. 29–Nov. 1, 2024. https://doi.org/10.1145/3681772.3698216.