Thesis submitted to the

UNIVERSITY OF MOHAMED BOUDIAF - M'SILA, ALGERIA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

in Partial Fulfillment of the Requirements for the Degree of:

Master's in Computer Science

Specialization: Business Intelligence & Optimization

By

GUENDOUZ Ouafa

AZZOUZ Ayat Errahmane

Entitled

NEURO-METAHEURISTIC APPROACHE FOR ADAPTIVE VEHICLE ROUTING IN URBAN TRANSPORT NETWORKS

Under the supervision of

Allaoua Hemmak

Publicly defended on 6th of June 2026

Jury Members

Tahar Mehenni	University of M'sila	President
Allaoua Hemmak	University of M'sila	Reporter
Said Hamani	University of M'sila	Examiner

ACADEMIC YEAR: 2025/2026

الملخص

تعالج هذه المذكرة تحدي تحسين مسارات المركبات في البيئات الحضرية الديناميكية، حيث تتغير حركة المرور والطلبات وحالات المركبات أثناء التشغيل. وتقترح إطارًا هجينًا باسم **NMFAVRK** يجمع بين **الشبكات العصبية الرسومية (GNN)** و**الخوارزميات الجينية (GA)** لحل مشكلة توجيه المركبات الديناميكية مع النوافذ الزمنية .(DVRPTW) يعتمد الإطار على تمثيل مكاني-زمني للعملاء لاستخراج أولويات زيارة ذكية تُستخدم في تهيئة البحث وتحسينه، ثم يدعمها بآليات بحث محلي وإصلاح سريع للمسارات عند ظهور أحداث مفاجئة مثل الازدحام أو الأعطال أو الطلبات الجديدة. أظهرت التجارب على مجموعة C102 Solomon أن النظام حقق مسافة كلية قدرها 828.9 مع فجوة صغيرة جدًا عن أفضل حل معروف، وتفوق على الخوارزمية الجينية العشوائية بنسبة ملحوظة. كما أظهر قدرة عالية على الاستجابة الفورية للاضطرابات خلال أجزاء من الثانية، مع تقليل استهلاك الوقود ضمن نموذج التوجيه البيئي. تؤكد النتائج أن الدمج بين التعلم العميق والبحث الميتاهيوريستي يحسن جودة الحلول ومرونتها التشغيلية، ويوفر أساسًا عمليًا لتطبيقات النقل الذكي والخدمات اللوجستية الحضرية التي تتطلب قرارات دقيقة وسريعة.

الكلمات المفتاحية: مشكلة توجيه المركبات الديناميكية مع النوافذ الزمنية ،(DVRPTW) الشبكات العصبية الرسومية ،(GNN) الخوارزميات الجينية ،(GA) التوجيه التكيفي.

Abstract

This dissertation addresses the challenge of optimizing vehicle routes in dynamic urban environments, where traffic conditions, customer requests, and vehicle states change during operations. It proposes a hybrid framework, **NMFAVRK**, that combines **Graph Neural Networks (GNN)** and a **Genetic Algorithm (GA)** to solve the Dynamic Vehicle Routing Problem with Time Windows (DVRPTW). The framework builds spatiotemporal customer representations to extract intelligent visit priorities used to initialize and improve the search, then reinforces them with local search and fast route repair when unexpected events such as congestion, breakdowns, or new requests occur. Experiments on the Solomon C102 benchmark show that the system achieved a total distance of **828.9** with a very small gap from the best known solution and outperformed a random GA baseline by a notable margin. It also responded to disruptions within fractions of a second while reducing fuel consumption under the green routing model. The results confirm that combining deep learning with metaheuristic search improves solution quality and operational flexibility.

Keywords: Dynamic Vehicle Routing Problem with Time Windows (DVRPTW), Graph Neural Networks (GNN), Genetic Algorithm (GA), Adaptive Routing.

Dedication

بسم الله الرحمن الرحيم

من تعظم الكون بنوره، له وحده أشكر وأسجد شاكراً نعمه سبحانه،

إلى شفيع الكائنات عند رب السماوات، الحبيب محمد ﷺ،

الى والدي العزيزين . بارك الله فيهما وأطال في عمرهما. وجعلني دائما سببا في فخرهما و سعادتهما.

إلى إخوتي وأخواتي الأعزاء , حفظهم الله ورعاهم .وأدامهم سنداً لي استمد بهم قوتي وأشد بهم أزري.

إلى كل غيور على العلم، ولكل من ساهم بكلمة توجيه، أو نصيحة صادقة، أو دعوة بظهر الغيب، أنارت لي عتمة السير في دروب البحث العلمي.

أهدي هذا العمل المتواضع.

قندوز وفاء

Dedication

بسم الله الرحمن الرحيم

الحمد لله الذي أنار لي درب العلم، ووفقني لإتمام هذا العمل المتواضع، فله سبحانه أول الحمد وآخره، وظاهره وباطنه، على نعمه التي لا تُعد ولا تُحصى.

إلى نفسي...
إلى تلك التي صبرت، واجتهدت، وتعبت كثيرًا حتى تصل إلى هذه اللحظة، أهدي هذا النجاح الذي كان ثمرة صبرٍ طويل وإرادةٍ لم تنكسر.

إلى أبي الغالي...
سندي وفخري، من علّمني معنى الكفاح والثبات.

إلى أمي الحبيبة...
نبع الحنان والدعوات الصادقة.

إلى إخوتي الأعزاء...
شكرًا لدعمكم ومحبتكم.

كما أتقدم بخالص الشكر والتقدير إلى أساتذتي الأفاضل والمشرفين على هذه المذكرة.

إلى كل من ساهم من قريب أو بعيد في إنجاز هذا العمل.

عزوز آية الرحمان

Acknowledgements

All praise and thanks are due to Allah for granting us the strength, patience, and guidance to complete this work.

We would like to express our sincere gratitude and deepest appreciation to our supervisor, **Dr. Alaoua Hammak**, for accepting to supervise this thesis and for providing us with valuable guidance, continuous support, insightful advice, and constant encouragement throughout the preparation of this research.

We also extend our sincere thanks to the honorable members of the examination committee for taking the time to review and evaluate this dissertation, and for their valuable remarks and suggestions that will certainly enrich and improve this work.

Our heartfelt thanks also go to the Faculty of Mathematics and Computer Science at Mohamed Boudiaf University of M'Sila, as well as to all professors of the Department of Computer Science, for the knowledge and support they provided throughout our academic journey.

We are equally grateful to the library and laboratory staff, and to everyone who offered assistance and facilities that helped us accomplish the practical part of this research.

Special thanks are dedicated to our families and friends for their patience, encouragement, and continuous support during this challenging period.

We would also like to thank our university for giving us the opportunity to work together as a team, which helped us strengthen our collaboration and communication skills.

Finally, we sincerely thank everyone who contributed, directly or indirectly, to the completion of this humble work.

Contents

List of Tables

List of Figures

General Introduction

1. **Research Context and Background**

   Urban transportation systems have undergone major changes due to population growth, urban expansion, and the rising demand for mobility and distribution services. These pressures intensify congestion, complicate fleet management, and make service reliability harder to maintain in operating environments that evolve continuously.

   In this context, the Vehicle Routing Problem (VRP) has become a central problem in logistics and supply chain management. It seeks cost-efficient routes for a fleet of vehicles while respecting operational constraints such as capacity, route duration, and service time windows [1]. In urban settings, however, traffic conditions, customer requests, and service times may change during execution. This leads to the Dynamic Vehicle Routing Problem (DVRP), where routing decisions must be revised in real time [2].

   Exact optimization remains useful for static models, but it becomes difficult to apply efficiently at large scale under uncertainty. Metaheuristics can provide high-quality solutions quickly, yet they generally do not learn from previous instances. Neural models, by contrast, can extract patterns from data and guide the search process. Their integration motivates neuro-metaheuristic approaches for adaptive vehicle routing.

2. **Problem Definition**

   - **Definition:** The VRP is a combinatorial optimization problem, first introduced by Dantzig and Ramser in 1959, and commonly viewed as a generalization of the Traveling Salesman Problem (TSP). It determines routes for vehicles leaving one or more depots, serving customers, and returning while minimizing costs measured by distance, time, fuel, or a combination of these criteria [3]. Typical constraints include vehicle capacity, customer demand, service times, time windows, and driver working hours. Because it combines multiple vehicles and operational constraints, VRP is NP-hard.
   - **Variants:** Major variants include CVRP, VRPTW, DVRP, ACVRP, CVRPTW, and HFVRP, each reflecting a different practical constraint set [4].
   - **Dynamic focus:** In real applications, requests may arrive online, travel times may vary because of congestion, and disruptions such as vehicle breakdowns may occur. Therefore, routing plans must be updated continuously from real-time information.

- **Proposed approach:** This dissertation adopts a neuro-metaheuristic approach that combines the exploration strength of metaheuristics with the predictive and adaptive capabilities of neural models to produce more responsive routing decisions.

3. **Motivation and Rationale**

   - **Real-time response:** Re-optimizing routes from scratch whenever the environment changes is computationally expensive, especially in dense urban settings.
   - **Learning to optimize:** Many current methods solve each instance independently. Neural guidance can reuse previous experience and accelerate the search toward better feasible solutions.
   - **Sustainability:** Better routing reduces traveled distance, fuel use, and carbon emissions, which directly supports smart-city and sustainable logistics goals.

4. **Research Objectives** The main objective of this research is to develop an adaptive framework for solving vehicle routing problems in dynamic urban environments by integrating metaheuristic optimization with neural learning models. The specific objectives are:

   - To model vehicle routing under dynamic urban conditions, including traffic variation, changing demand, and operational constraints.
   - To design and implement a hybrid neuro-metaheuristic approach that combines search efficiency with learning and prediction.
   - To evaluate the performance of the proposed approach in terms of solution quality, computational efficiency, and adaptability to dynamic changes.
   - To compare the proposed framework with classical metaheuristic approaches.

5. **Research Questions and Hypothesis** This research is guided by the following questions:

   - How can neural learning models enhance the adaptability of metaheuristic algorithms in solving vehicle routing problems within dynamic urban environments?
   - Does integrating neural models with metaheuristic optimization improve solution quality and computational performance compared with traditional metaheuristics?

   The central hypothesis is that integrating neural learning models with metaheuristic algorithms improves vehicle routing efficiency and significantly strengthens adaptation to dynamic urban conditions.

6. **Methodological Approach** This dissertation combines theoretical modeling with experimental evaluation:

   (a) The vehicle routing problem is formulated for a dynamic urban environment with its main operational constraints.

   (b) A neuro-metaheuristic framework is designed and implemented to solve the problem.

   (c) The proposed framework is evaluated on benchmark instances and compared with traditional metaheuristic methods using relevant performance indicators.

7. **Dissertation Structure** The dissertation is organized as follows:

   - ***Chapter 1*** introduces adaptive optimization in urban transport and presents its main concepts, challenges, and indicators.
   - ***Chapter 2*** reviews the state of the art on vehicle routing, metaheuristics, and neuro-hybrid approaches.
   - ***Chapter 3*** presents the problem modeling and the architecture of the proposed neuro-metaheuristic framework.
   - ***Chapter 4*** describes implementation details and reports the experimental results and comparative analysis.
   - ***General Conclusion*** summarizes the main findings and outlines future research directions.

Overview of Adaptive Optimization for Urban Transport

1.1 Introduction

Modern urban areas are experiencing rapid population growth and increasing vehicle numbers, leading to heightened traffic congestion and greater complexity in managing transport networks. Efficient and intelligent approaches are now essential to enhance network performance, reduce travel times, and improve the responsiveness of transport services. In this context, adaptive optimization plays a central role, allowing transport systems to adjust dynamically to changing traffic conditions and demand patterns, while leveraging real-time data for more informed decision-making.

This chapter provides an overview of adaptive optimization techniques for urban transport systems, highlighting key concepts, system types, and performance indicators. It also discusses classical optimization methods, their limitations, and the need for integrating machine learning and metaheuristic techniques to achieve more flexible and effective solutions. By establishing this foundation, the chapter sets the stage for understanding hybrid neuro-metaheuristic approaches, which will be explored in subsequent chapters for adaptive vehicle routing in urban transport networks.

1.2 Urban Transport Systems: Concepts and Characteristics

1.2.1 Definition of Urban Transport Systems

Urban transport systems refer to the set of networks, infrastructures, and transport modes that enable the movement of people and goods within urban areas. These systems include roads, railways, and related facilities, as well as vehicles and traffic flows. They operate as an integrated unit aimed at meeting mobility demand and facilitating efficient travel within cities [5].

1.2.2 Vehicle Routing Problems (VRPs)

In real-world urban transportation and logistics, combinatorial optimization is a key tool for designing efficient delivery and routing plans. The Vehicle Routing Problem (VRP) focuses on determining optimal routes for a fleet of vehicles to visit a set of customers from a central

depot, while respecting constraints such as vehicle capacity, service times, and route lengths, and minimizing total travel cost. VRPs are classified as **NP-Hard problems**, meaning that exact solutions become computationally intractable for large instances, especially in complex urban networks. This motivates the use of adaptive optimization, evolutionary algorithms, machine learning, and hybrid approaches to obtain practical solutions within reasonable computational time [1, 6].

Some of the most studied VRP variants include:

Capacitated Vehicle Routing Problem (CVRP): A fleet of identical vehicles with fixed capacity, aiming to minimize total travel cost.

Asymmetric VRP (ACVRP): Allows directional travel costs to vary, reflecting one-way streets or variable congestion.

VRP with Time Windows (CVRPTW): Service at each customer must start within a specified time window, increasing modeling and computational complexity.

Heterogeneous Fleet VRP (HFVRP): Vehicles with different capacities and travel costs, adding realism and computational complexity.

1.2.3 Main components

The main components of urban transport systems can be analyzed from two complementary perspectives:

First **Modeling Perspective**

Based on the fundamental principles of transport systems analysis, the system can be represented as follows:

Flow: people or goods being transported.

Vehicles: the means that provide transportation services.

Network: the links and nodes (roads, railways, stations) through which vehicles move.

This model is used to understand flows, analyze networks, and solve optimization and planning problems. Such a theoretical representation constitutes the foundation of travel demand modeling and transportation network analysis. This perspective is grounded in the principles introduced by Manheim in transport systems analysis [7].

second **Operational Perspective** From a more practical and applied viewpoint, urban transport systems are divided into structural and functional components as follows [8]:

Infrastructure: roads, railways, stations, parking facilities, pedestrian paths, and bicycle lanes.

Vehicles: private and public transport modes as well as shared mobility options.

Operations and Management: traffic control systems, scheduling, ticketing, and real-time information systems.

Users: individuals and organizations that rely on the system, whose needs and behaviors influence service design.

These components collectively enable improvements in operational efficiency and support real-time decision-making.

1.2.4 Types of Urban Transport Systems

Urban transport systems can be classified according to service characteristics and right-of-way independence as follows:

1. **By Service Mode and Type of Operation** [9]

   - **Private Transport**: Includes private cars and motorcycles that provide high individual flexibility and door-to-door mobility, but often contribute significantly to congestion and emissions.
   - **Public Transport**: Includes buses, metro, tram, and rail systems designed to carry large numbers of passengers along fixed routes and schedules, offering higher energy efficiency and reduced road congestion.
   - **Shared Mobility:** Refers to app-based and on-demand services such as ride-hailing platforms (e.g., Uber, Yassir) and shared micromobility options (e-scooters, bike-sharing), combining flexibility with resource efficiency.
   - **Active Mobility:** Includes walking and cycling as sustainable, low-cost, and environmentally friendly travel options that also promote public health.

2. **By Right-of-Way Independence**[10]

   - **Category A** – Fully Separated Right-of-Way: Systems operating on completely exclusive tracks or corridors (e.g., metro). These systems provide high speeds, high capacity, and full operational control.
   - **Category B** – Partially Separated Right-of-Way: Systems using semi-exclusive lanes that may occasionally intersect with other traffic (e.g., tramways or light rail). These require priority management and signal coordination.
   - **Category C** – Mixed Traffic Right-of-Way: Systems sharing road space with general traffic (e.g., buses). Their performance is highly affected by congestion and variability, making them suitable candidates for adaptive and real-time optimization strategies.

1.2.5 Key Performance Indicators (KPIs)

Transport KPIs serve as a pivotal instrument for evaluating and enhancing the performance of urban transport systems at both local and regional scales. Based on the conceptual framework developed by Mohamed (2024)[11], these indicators can be categorized into five fundamental dimensions:

Efficiency[12]: This dimension assesses the system's operational performance, focusing on travel time, vehicle occupancy rates, and route optimization. Monitoring these factors allows for identifying systemic inefficiencies, leading to significant cost savings and improved service quality .

Safety[13]: These indicators measure the overall security of the transport environment, including accident rates, the safety of vulnerable road users (pedestrians and cyclists), and strict compliance with traffic regulations.

Sustainability[14]: This category evaluates the environmental footprint of transport activities by monitoring carbon emissions, energy consumption, and pollution levels, thereby supporting the transition toward a more sustainable urban future.

Accessibility[15]: This focuses on the inclusivity of the transport system, analyzing the availability of public transit, the affordability of fares, and the provision of specialized options for people with disabilities .

Integration[16]: This metric assesses the seamless connectivity between various transport modes—such as public transit, cycling, and walking—to promote more efficient and multimodal mobility choices.

1.3 Optimization in Urban Transport

1.3.1 Role of optimization in transport planning and operations

Optimization is considered one of the fundamental pillars of urban transport engineering, as it enables the rational use of limited resources in the face of increasing demand and growing system complexity. The role of optimization clearly appears at two complementary levels.

At the planning level, optimization is used to design transport networks, define routes, and allocate capacities based on demand forecasts. When combined with intelligent transportation systems, it allows planners to compare different scenarios and make long-term strategic decisions aimed at improving overall system performance [17].

At the operational level, optimization focuses on daily and real-time management tasks, such as trip scheduling and traffic flow control in response to temporary congestion. With the emergence of smart cities, these techniques have become essential for reducing delays and enhancing service reliability [18].

This integration between planning and operational levels plays a key role in ensuring the sustainability and resilience of the urban transport system as a whole.

1.3.2 Classical optimization approaches

Traditional approaches to transportation system optimization are primarily grounded in rigorous mathematical modeling, with the main objective of identifying the global optimum within a well-defined and constrained environment. These approaches assume that system parameters and constraints are known in advance, enabling planners to formulate the problem in a deterministic framework. Depending on the nature of the problem being addressed, classical optimization methods can be categorized as follows:

- Linear and Nonlinear Programming: These models are widely employed to solve traffic assignment and resource allocation problems. By defining an objective function, planners aim to minimize costs or reduce travel times while satisfying operational and physical constraints. Such formulations provide a structured way to balance efficiency and feasibility within transportation networks.

- Integer Programming: This class of methods becomes particularly important when decision variables are discrete and cannot be divided. Typical applications include determining the number of buses required or scheduling operational staff. In these cases, decisions are inherently indivisible, which necessitates mathematical models capable of handling integer variables to ensure accurate scheduling and reliable system operation [19].

- Network Search Algorithms: These approaches rely on graph theory to address shortest-path or maximum-flow problems. Algorithms such as Dijkstra and Bellman–Ford are among the most commonly used tools in this domain. They support traffic routing and help identify the most efficient paths under static and predetermined conditions, thereby improving network performance and travel efficiency [20].

1.3.3 Limitations

- Classical optimization models assume deterministic and static conditions, whereas urban transport systems are dynamic and subject to uncertainty due to fluctuating demand, congestion, and unexpected events

- Traditional approaches struggle to effectively incorporate stochastic and time-varying data, reducing their robustness in real-world applications.

- As network size and decision variables increase, transport problems become computationally intensive (often NP-hard), making exact solutions impractical for large-scale or real-time systems.
- Classical models do not learn from historical data nor adapt over time; they operate under fixed formulations without self-improvement mechanisms.
- These methods typically provide offline solutions based on a snapshot of system conditions and lack continuous real-time re-optimization capabilities.
- Classical formulations often focus on single-objective optimization, whereas urban transport systems require balancing multiple conflicting objectives such as efficiency, sustainability, safety, and equity.

1.4 Adaptive Optimization: Fundamentals

1.4.1 Definition and principles of adaptive optimization

Adaptive optimization refers to a class of algorithms that automatically adjust their update rules or parameters during execution, based on feedback from past iterations or observations. Unlike traditional optimization methods that rely on fixed schedules and pre-defined rules, adaptive optimization continuously learns from its environment and modifies its actions accordingly. The key principles of adaptive optimization include: Data-driven adaptation: Using observed data to guide updates and decisions.

Feedback integration: Incorporating results from previous iterations to improve future performance.

Real-time adjustment: Dynamically modifying parameters to respond to changing conditions.

Self-improvement: Gradually enhancing efficiency and convergence as more information becomes available.

This methodology allows systems to respond effectively to uncertainty and variability, making it particularly suitable for complex, dynamic environments such as traffic systems, supply chains, and machine learning applications [21][22].

1.4.2 Static vs. Dynamic vs. Adaptive Optimization

Optimization approaches in urban transport systems can be classified into three main categories: static, dynamic, and adaptive optimization. Each approach differs in terms of flexibility, computational complexity, and suitability for real-world applications. Table 1.1 presents a comparative overview.

Table 1.1: Comparison between Static, Dynamic, and Adaptive Optimization Approaches [22, 23]

Type	Definition	Advantages	Disadvantages	Example	Best Use Case
Static	Optimization performed before execution using predefined models and fixed assumptions.	No runtime overhead; stable and predictable performance.	Cannot adapt to changes in traffic demand or environmental conditions.	Predefined traffic signal schedule based on average daily flow.	Stable environments with regular traffic patterns.
Dynamic	Optimization performed during execution by reacting to real-time system behavior.	Adapts to real-time conditions and short-term fluctuations.	Introduces runtime overhead; may not reach global optimality.	Traffic lights adjusting timing according to live vehicle queues.	Intersections with variable traffic demand.
Adaptive	Combines a static baseline with continuous learning and feedback-driven adjustments.	Highly flexible; learns from historical and live data.	Requires continuous sensing and higher computational cost.	Smart urban traffic control using machine learning.	Large-scale urban networks with evolving patterns.

1.4.3 Feedback, learning, and real-time adaptation

In urban transport optimization, real-time adaptation and feedback loops are crucial for enhancing the responsiveness and efficiency of routing systems. Feedback loops involve the process of collecting data from the system or users, analyzing it, and acting upon it to continuously improve performance [24]. This approach allows transportation systems to adapt dynamically to changing traffic conditions and demand patterns, unlike static or pre-scheduled strategies [25].

Modern hybrid frameworks, such as the integration of ANFIS, Genetic Algorithms, and Reinforcement Learning (ANFIS–GA–RL), have demonstrated the ability to adjust traffic signals in real time, improving performance metrics like average vehicle speed, queue lengths, and intersection delays [25]. Real-time adaptation relies on three key components:

Continuous data processing: Monitoring real-time traffic or demand data.

Contextual analysis: Understanding traffic context to guide decisions effectively.

Dynamic learning: Using machine learning models to update routing strategies based on continuous feedback [24].

Integrating these components enhances user experience by reducing delays, increasing

trust in the system, and allowing rapid responses to unexpected events such as accidents or sudden demand spikes. Overall, feedback-driven real-time adaptation provides a foundation for more intelligent and efficient urban transport systems.

1.5 Adaptive Optimization Techniques for Urban Transport

1.5.1 Metaheuristics

Metaheuristics are approximate optimization methods designed to tackle complex and large-scale problems in urban transport, where exact classical methods are computationally infeasible. They balance exploration (searching diverse solution spaces to avoid local optima) and exploitation (refining promising solutions) [26].

Key Techniques

Genetic Algorithms (GA): Inspired by natural evolution, widely used for transit network design and scheduling [1].

Ant Colony Optimization (ACO): Simulates pheromone-based pathfinding, effective for shortest-path and distribution problems.

Particle Swarm Optimization (PSO): Mimics social behavior to optimize variables such as traffic signal timings and flow control [1].

Adaptive Metaheuristics

Adaptive variants dynamically adjust search parameters and operators in response to real-time traffic fluctuations and demand shifts, enhancing system resilience and performance in stochastic urban environments [26, 27].

1.5.2 Machine learning–based optimization

Definition

Machine Learning (ML)-based optimization leverages historical and real-time data to predict traffic patterns and demand fluctuations. These predictions support proactive routing, scheduling, and resource allocation in urban transport systems [28].

Key Techniques

Regression and Ensemble Models: Forecast spatio-temporal demand, enabling fast decision-making from historical trends [29].

Deep Learning (Neural Networks, LSTM): Captures complex non-linear spatiotemporal patterns for accurate flow prediction and demand modeling [30].

Role in Adaptive Routing

ML empowers systems to dynamically adjust routes and schedules in response to real-time disruptions, maintaining high efficiency under uncertain urban scenarios.

Integration with Metaheuristics

ML can generate high-quality initial solutions or adaptively tune metaheuristic parameters, creating intelligent optimization frameworks that evolve with the transport environment [26].

1.5.3 Reinforcement learning approaches

Definition

Reinforcement Learning (RL) is a powerful paradigm for dynamic systems. It consists of an *Agent* (decision-maker), an *Environment* (transport network), *Actions* (routes or schedules), and a *Reward* (feedback such as minimized delays). RL adapts dynamically without requiring explicit programming of all scenarios [31].

Applications in Urban Transport

Key applications include real-time vehicle routing, dynamic public transport scheduling, and traffic flow management at intersections [32].

Common Algorithms

Q-Learning: Suitable for simple environments using reward tables.

Deep Q-Networks (DQN): Combines deep neural networks with reinforcement learning to handle large-scale and complex systems [33].

1.5.4 Hybrid and multi-objective optimization methods

Definition

Hybrid methods combine metaheuristics with ML or RL to leverage the strengths of each approach. Multi-objective optimization addresses conflicting goals (e.g., minimizing travel

time versus fuel consumption) to identify Pareto-optimal solutions [34].

Applications

GA + RL: Genetic Algorithms generate robust routes, while RL adapts them in real time.

ML + ACO: Neural networks forecast demand, and ACO plans vehicle distribution.

Multi-objective balancing: Simultaneous optimization of efficiency, operational costs, and environmental impacts [35].

1.6 Applications

1.6.1 Traffic signal control

Traffic signal control involves adjusting signal timings to optimize vehicle flow. Enhanced Deep Reinforcement Learning (DRL) techniques, such as priority sampling and dueling networks, have shown significant effectiveness in reducing queue lengths and waiting times at urban intersections [36].

Practical example: The Surtrac system in Pittsburgh enables intelligent coordination between signals, reducing travel times by 25% and emissions by nearly 20%.

1.6.2 Route planning and navigation

Route planning focuses on determining the optimal path to minimize travel time or congestion. Reinforcement Learning approaches for connected vehicles enable real-time path adaptation under changing traffic conditions, improving overall network throughput [37].

Practical example: Applications like Google Maps and Waze rely on real-time user data to reroute traffic and dynamically avoid congestion.

1.6.3 Public transport scheduling

Public transport scheduling refers to organizing vehicle timetables to meet fluctuating demand. Recent frameworks utilize Multi-Agent Reinforcement Learning (MARL) to manage dynamic scheduling for flexible bus services, ensuring both operational efficiency and fairness [38].

Practical example: London’s Transport for London (TfL) uses intelligent algorithms to mitigate ``bus bunching,'' ensuring consistent headways between buses and improving service reliability.

1.6.4 Demand prediction and congestion management

Demand prediction aims to forecast traffic volumes to support proactive mobility strategies. Deep Learning-based models have demonstrated high accuracy in predicting traffic flows, enabling city operators to manage congestion before bottlenecks occur [39].

Practical example: Shenzhen, China applies AI-based demand forecasting systems to proactively manage traffic flow.

1.7 Challenges and Open Issues

1.7.1 Scalability and computational complexity

The performance of hybrid mechanisms can be limited when applied to larger or more complex problem instances. Scaling up to real-world networks requires significant computational resources and careful tuning of hyperparameters [4].

1.7.2 Data availability and quality

Real-world transport problems often involve incomplete, uncertain, or rapidly changing information. The effectiveness of adaptive routing algorithms depends on high-quality and reliable data [4].

1.7.3 Real-time constraints and uncertainty

Adaptive systems must make decisions under time pressure while handling dynamic and uncertain conditions. Variations in traffic, demand, and domain-specific rules pose significant challenges for real-time optimization [4].

1.8 Conclusion and Chapter Summary

This chapter provided an overview of adaptive optimization techniques for urban transport systems, emphasizing the need for intelligent and flexible approaches in modern cities. Urban transport networks are inherently complex, dynamic, and subject to fluctuating demand, traffic congestion, and operational constraints. Traditional optimization methods often fail to handle these dynamics effectively, highlighting the importance of adaptive, learning-based, and real-time strategies.

Key points discussed in this chapter include:

1. **Urban Transport Systems:** Definition, components, types, and key performance indicators, illustrating the complexity of urban mobility.

2. **Optimization in Urban Transport:** The role of optimization, classical approaches, and their limitations in addressing dynamic and large-scale problems.

3. **Adaptive Optimization Fundamentals:** Principles of adaptive optimization, differentiating static, dynamic, and adaptive methods, and the importance of feedback loops and learning mechanisms.

4. **Adaptive Optimization Techniques:** Overview of metaheuristics, machine learning-based optimization, reinforcement learning, and hybrid methods for urban transport applications.

5. **Applications:** Practical applications in traffic signal control, route planning, public transport scheduling, and demand prediction.

6. **Challenges and Open Issues:** Key challenges including scalability, computational complexity, data quality, real-time constraints, uncertainty, and generalizability of hybrid approaches.

In summary, adaptive optimization offers significant potential to improve urban transport efficiency, responsiveness, and sustainability. However, real-world implementation requires addressing challenges such as large-scale network complexity, data limitations, and dynamic operational conditions. The next chapters will focus on designing and evaluating neuro-metaheuristic approaches to achieve adaptive vehicle routing in urban transport networks.

State-of-the-Art

2.1 Introduction

This chapter aims to provide an analytical review of recent scientific works related to the vehicle routing problem in urban transport networks, with a particular focus on approaches that combine machine learning techniques with metaheuristic algorithms. This field has witnessed remarkable evolution in recent years as a result of advances in artificial intelligence technologies, leading to the emergence of hybrid models seeking to improve the efficiency of routing systems in complex urban environments. This chapter specifically focuses on neuro-metaheuristic approaches, which combine the learning capabilities from data provided by neural networks and reinforcement learning techniques with the efficiency of metaheuristic algorithms in exploring large solution spaces. This integration aims to address challenges associated with dynamic and real-time routing problems characterized by uncertainty and continuous changes in demand and traffic conditions. The studies analyzed in this chapter were selected based on several criteria, including recency of publication, relevance to the vehicle routing problem, and reliance on advanced optimization techniques or machine learning models. Emphasis was also placed on studies dealing with dynamic environments or applications related to urban transport networks. To organize this review systematically, studies were classified according to several criteria including methodological type, nature of adaptation in the model, type of routing problem studied, and application environment. This classification helps highlight recent research trends and identify limitations and research gaps that this work seeks to address.

2.2 Classification of Studies

The reviewed studies are classified according to four main dimensions: (i) methodological paradigm (metaheuristic, neural, or hybrid), (ii) adaptivity level (static, dynamic, or real-time), (iii) problem formulation (VRP variants including DVRP and RTVRP), and (iv) experimental environment (benchmark datasets or real-world urban networks). This multidimensional classification enables a structured comparison of algorithmic behavior and environmental responsiveness.

Summary of Findings:

- Methodological Paradigm: Metaheuristic approaches dominate early works [40, 41]. Neural-based approaches increasingly address dynamic decision-making [42, 43]. Hybrid ML–optimization frameworks integrate learning with search [19, 44].

- Adaptivity Level: Static formulations prevail in benchmark-oriented studies [4, 40]. Dynamic and real-time adaptivity appear in traffic-aware and stochastic contexts [45, 46].
- Problem Type: Core VRP variants include CVRP, VRPTW, DVRP, MDVRP, and specialized formulations such as Cash-in-Transit and Emergency VRPTW.
- Environment: Only few studies validate on real urban networks [41, 43, 47], highlighting a gap in real-world applicability.

2.3 Thematic Sections

1. **Metaheuristic-Based Routing Approaches** Metaheuristic algorithms have long been recognized as one of the most effective approaches for solving vehicle routing problems due to their strong ability to explore large combinatorial search spaces and generate high-quality near-optimal solutions. These techniques typically rely on neighborhood exploration, population-based evolution, and memory-guided search mechanisms to iteratively improve routing performance.

   Several studies have successfully applied these methods to different VRP variants. For instance,[40] employed the Artificial Bee Colony (ABC) algorithm to solve capacitated routing problems under industrial constraints, while [41] proposed an Adaptive Memory Procedure (AMP) for real-world urban routing scenarios. Other works have extensively explored Genetic Algorithms (GA), Ant Colony Optimization (ACO), and swarm-based techniques to address increasingly complex routing constraints.

   As illustrated in Figure 2.1, Genetic Algorithms (GA) and Tabu Search emerge as the most frequently applied metaheuristic techniques in VRP-related studies, with 753 and 498 articles respectively, followed by Simulated Annealing (426) and Simple Optimizer (359). This predominance reflects their robustness in exploring broad combinatorial search spaces, their flexibility in handling multiple routing constraints, and their natural compatibility with hybrid optimization frameworks.

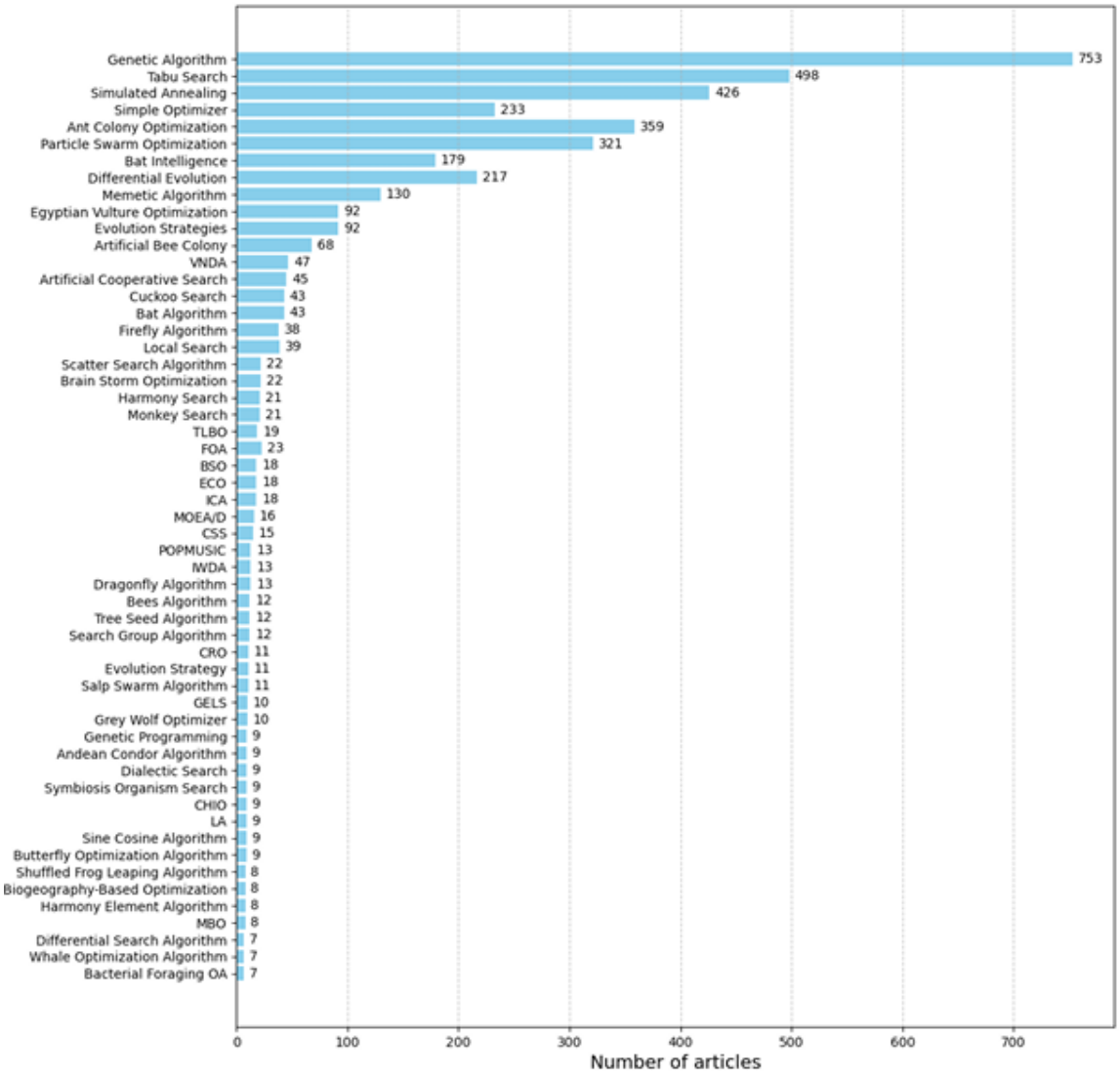

Figure 2.1: Distribution of metaheuristic methods applied in VRP literature [48]

The frequent use of population-based algorithms such as GA in the literature suggests that they are particularly suitable for hybridization with intelligent learning components. Their population structure provides a natural framework for integrating priority scores and embeddings generated by neural models, allowing the search to be guided toward high-quality regions of the solution space from the very first generation.

The frequent use of population-based algorithms such as GA in the literature suggests that they are particularly suitable for hybridization with intelligent learning components.

Despite their effectiveness in handling large-scale problems with multiple constraints, metaheuristic algorithms remain limited by their lack of data-driven adaptability. Their behavior is generally governed by manually designed operators and static parameter settings, and most existing applications are evaluated in static or semi-dynamic en-

vironments. The proposed GNN--GA integration directly addresses these limitations by replacing hand-crafted initialization heuristics with learned spatiotemporal embeddings.

2. **Neural-Based and Reinforcement Learning Approaches** The rapid progress of deep learning and data-driven optimization has led to the emergence of neural-based frameworks for vehicle routing optimization, leveraging reinforcement learning (RL), deep reinforcement learning (DRL), and graph-based architectures to learn routing policies directly from data. Several studies have demonstrated the effectiveness of these approaches:[42] was among the earliest to apply RL to the capacitated VRP, while later works by [49], [45], [46], and [43] extended them to dynamic and stochastic environments using attention mechanisms, graph representations, and multi-agent coordination.

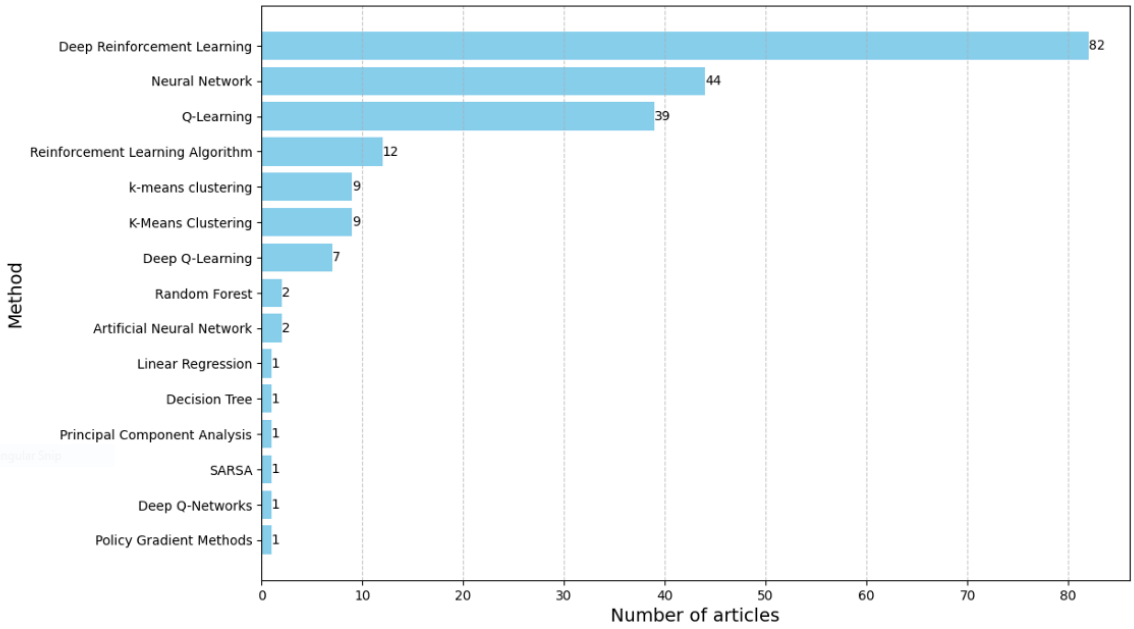

Figure 2.2: Distribution of machine learning methods applied in VRP literature [48]

As illustrated in Figure 2.2, DRL dominates the literature, confirming the growing importance of adaptive learning in dynamic routing environments. However, RL-based methods require extensive reward engineering, offer limited interpretability, and are difficult to integrate with classical combinatorial optimizers. The proposed system therefore adopts a *neuro-metaheuristic* paradigm: a Graph Neural Network (GNN) learns spatiotemporal customer embeddings and route-cohesion patterns, which guide a Genetic Algorithm (GA) toward high-quality solutions avoiding RL instability while preserving the adaptive capacity required for real-time DVRPTW.

3. **Hybrid Neuro-Metaheuristic Approaches**

Hybrid frameworks combine metaheuristic optimization with machine learning techniques or neural components to leverage the advantages of both paradigms. In these approaches, learning does not directly solve the routing problem but rather serves to guide search processes, adjust parameters, predict solution quality, or model uncertain travel times.

Herdianto [4] introduces a learning-guided metaheuristic incorporating feature-based selection mechanisms to enhance search efficiency. Reis [44] proposes an adaptive hybrid algorithm for solving VRPTW under travel time uncertainty, integrating genetic algorithms with local search and scenario-based robustness analysis.

In most of these approaches, the final optimization process remains driven by traditional metaheuristic algorithms such as GA or local search, while machine learning components serve as supporting modules. Although these models improve search quality and scalability, most are evaluated in static benchmark environments and operate within an offline optimization framework. Continuous learning during real-time execution remains limited.

4. **Adaptive and Real-Time Routing in Urban Networks**

   Adaptive and real-time routing has become increasingly important in urban transport systems characterized by traffic congestion, stochastic demand, and changing operational constraints. This research direction focuses on dynamically arriving new requests, re-optimization during execution, and decision-making under partial observation.

   Recent studies such as [45] and [46] model routing problems as Partially Observable Markov Decision Processes (POMDP) to support dynamic decision-making. These frameworks enable real-time request acceptance, rerouting, and coordination mechanisms in uncertain environments.

   Despite significant progress in modeling dynamic scenarios, large-scale industrial applications remain relatively limited. Furthermore, the number of studies integrating hybrid neuro-metaheuristic strategies within fully adaptive urban transport networks is still scarce, opening promising research opportunities.

2.4 Comparative study

A comparative study of the reviewed literature was conducted, focusing on the key characteristics of each approach. The results are summarized in table 2.1 below. This table provides a structured overview of the state-of-the-art techniques for vehicle routing problems, detailing for each study the publication year, the methodological paradigm (metaheuristic,

neural, or hybrid), the specific model or algorithm employed, the type of adaptivity (static, dynamic, or real-time), the main optimization objectives, and the type of data used for validation (benchmark datasets or real-world urban networks). This classification facilitates a direct comparison of the different strategies and highlights the evolution of research in this domain.

Table 2.1: Comparative Analysis of Recent Approaches for Vehicle Routing Problems

Authors	**Year**	**Method**	**Model**	**Adaptivity**	**Objectives**	**Data**
Herdianto[4]	2025	ML-guided Metaheuristic	CVRP / VRPTW	Adaptive search	Improve solution quality and scalability	Benchmark datasets
Reis[44]	2025	Hybrid GA + Local Search	VRPTW (uncertain)	Static	Robust optimization under stochastic travel time	Benchmark datasets
Peric et al. [41]	2024	Adaptive Memory Procedure	Real-world VRP	Dynamic	Reduce delivery time and cost	Real urban dataset (Croatia)
Yaddaden[49]	2023	Deep Reinforcement Learning	CVRP / Ride-Hailing	Dynamic	Learn routing policy from data	CVRPLib simulation
Nazari et al. [42]	2018	Reinforcement Learning	CVRP / SDVRP	Static / Dynamic	End-to-end routing policy learning	Synthetic benchmarks
Konovalenko & Hvattum [45]	2024	Deep RL	DVRP with Time Windows	Real-time	Dynamic decision making	Simulation grid
Kadyrov et al. [46]	2025	DRL + GNN	Dynamic CVRP	Dynamic	Handle demand and traffic uncertainty	Synthetic grid
Mozhdehi et al. [43]	2025	Multi-Agent DRL	MDVRP	Static	Multi-vehicle coordination	Real urban data
Simsir & Ekmekci [40]	2019	Artificial Bee Colony	VRPSDP	Static	Optimize routing with pickup and delivery	Benchmark instances

2.5 Critical Synthesis

2.5.1 Research Trends

Recent research on the vehicle routing problem demonstrates a clear evolution from traditional optimization methods based on metaheuristic algorithms toward learning-based approaches. Early studies primarily relied on algorithms such as Genetic Algorithms, Artificial Bee Colony, and Adaptive Memory Procedures to efficiently solve large-scale routing problems. In more recent studies, there has been a growing trend toward integrating machine learning and deep reinforcement learning techniques to enable data-driven decision-making. Neural approaches are particularly well-suited for dynamic environments where customer demands and traffic conditions change over time. Furthermore, many recent studies propose hybrid frameworks that combine metaheuristic algorithms with machine learning models to improve search efficiency and solution quality. These neuro-metaheuristic approaches represent an emerging research direction for addressing complex vehicle routing problems in

urban transport networks.

2.5.2 Limitations

Despite the significant progress achieved by recent studies, several limitations persist. Many deep reinforcement learning approaches are primarily evaluated in simulated environments or using benchmark datasets, which may not fully reflect the complexity of real-world urban transport networks. Additionally, neural models typically require large amounts of training data and substantial computational resources, limiting their applicability in real-time routing scenarios. On the other hand, traditional metaheuristic methods often rely on static optimization and lack learning mechanisms that enable continuous adaptation to changing traffic conditions or dynamically emerging requests.

2.5.3 Research Gaps

Based on the reviewed literature, several research gaps can be identified. Although many studies explore either metaheuristic algorithms or deep learning approaches for solving vehicle routing problems, relatively few research efforts focus on integrating these two techniques within adaptive frameworks capable of real-time decision-making. Moreover, most current hybrid approaches use machine learning only to guide the search process in an offline manner, rather than enabling continuous learning and adaptation during routing execution. Therefore, there remains a need to develop advanced neuro-metaheuristic approaches that combine learning capabilities with optimization techniques to support adaptive vehicle routing within complex urban transport networks

2.5.4 Positioning of the Proposed Research

Despite significant advances in vehicle routing solution methods, many studies still rely on static optimization or on learning models trained in an offline manner. Furthermore, the number of approaches that combine deep learning with metaheuristic algorithms within an adaptive framework oriented toward urban environments remains limited. Therefore, this research proposes a hybrid framework that integrates neural learning with metaheuristic optimization techniques to support adaptive vehicle routing within urban transport networks. This approach seeks to leverage the learning capabilities in data analysis and change prediction, while maintaining the efficiency of metaheuristic algorithms in exploring the solution space.

Unlike existing works that rely on static learning models and offline optimization, the proposed approach introduces an adaptive neuro-metaheuristic architecture that enables adaptive real-time routing in urban transport networks through continuous feedback from traffic and

mobility data streams. This framework aims to bridge the gap between offline evolutionary search and reactive neural-based routing approaches, offering a scalable, interpretable, and genuinely adaptive solution for next-generation

2.6 Conclusion

Through reviewing recent literature on the vehicle routing problem, it becomes evident that this field has witnessed remarkable evolution in recent years, with research shifting from reliance on traditional metaheuristic algorithms toward integrating machine learning and deep reinforcement learning techniques. These developments have contributed to improving models' ability to handle complex and large-scale problems.

Furthermore, neural approaches offer promising potential for supporting decision-making in dynamic environments through learning from data and adapting to changes. It also became clear that hybrid approaches combining machine learning with metaheuristic algorithms represent an increasingly important research direction, due to their ability to leverage the advantages of both paradigms.

However, many of these studies still rely on simulated environments or offline optimization, and practical applications in real-world urban transport networks remain relatively limited. Based on this, there emerges a need to develop more adaptive approaches capable of combining learning and optimization within a framework that supports real-time decision-making in complex urban environments.

From this perspective, this work seeks to propose a hybrid framework aimed at improving vehicle routing through integrating learning techniques with metaheuristic optimization algorithms in the context of urban transport networks.

Neuro-Metaheuristic Framework for Adaptive Vehicle Routing Networks

3.1 Introduction

This chapter presents the proposed methodology for addressing the Vehicle Routing Problem (VRP) in dynamic urban transport networks by employing a hybrid optimization approach that integrates meta-heuristic techniques with artificial intelligence models. The objective of this work is to develop an adaptive routing framework capable of enhancing route efficiency in urban environments characterized by dynamism and uncertainty. The proposed approach relies on integrating meta heuristic algorithms, known for their ability to explore the solution space effectively, with neural network models that provide advanced capabilities in learning complex routing and spatial patterns. This chapter first defines the studied problem, followed by the mathematical formulation. Then, the proposed hybrid approach is introduced, including its main components, algorithms, an illustrative example, and a workflow diagram.

3.2 Problem Statement

Given a transportation network of customers, depots, and road segments operating under dynamic conditions (time-varying demand, traffic congestion, vehicle breakdowns), determine optimal routing plans for a homogeneous fleet of vehicles such that:

- All customer demands are satisfied within their respective time windows.
- The total operational cost including travel distance, travel time, fuel consumption, time-window penalties, and emissions is minimized.
- Routing decisions are dynamically adapted in real time using a hybrid Neuro-Metaheuristic framework that combines Graph Neural Networks (GNN) with Genetic Algorithms (GA).
- The framework remains scalable and generalizable across different problem sizes, from small urban delivery networks to large-scale logistics systems. This problem is formulated as a variant of the Vehicle Routing Problem with Time Windows (VRPTW), extended to incorporate dynamic environment changes (DVRPTW).

Figure 3.1 illustrates a simplified representation of this routing scenario. On the left side, the figure presents the initial customer network composed of a central depot and several

customer nodes connected through all possible links. This part represents the original routing problem before optimization, where multiple routing alternatives are possible. On the right side, the optimized solution is shown, where the customer nodes are organized into efficient routes, each assigned to a specific vehicle, in order to reduce the total travel cost.

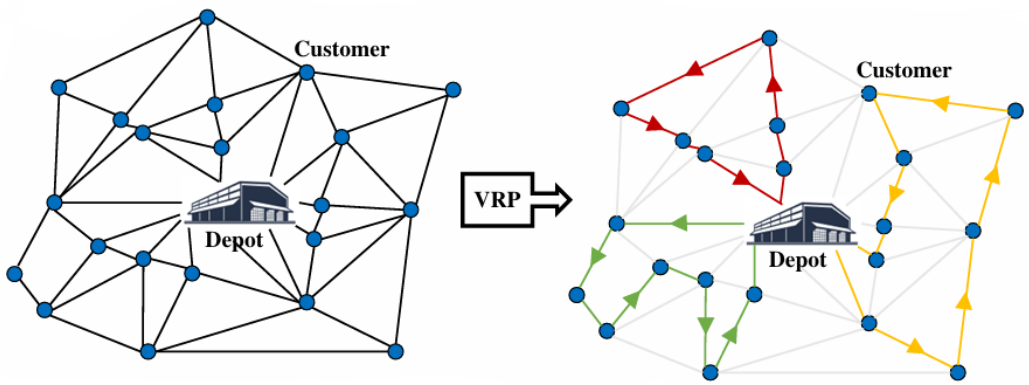

Figure 3.1: Illustration of the Vehicle Routing Problem (VRP) [50]

3.3 Motivation and Objective

3.3.1 Motivation

This research is based on a set of scientific and practical observations derived from a review of recent literature, in addition to the challenges imposed by transportation and logistics applications in urban environments.

1. Static approaches assume all data is known in advance, which is unrealistic in dynamic urban environments.
2. Integration of metaheuristics and deep learning in dynamic urban environments remains limited.
3. Urban networks involve high complexity requiring specialized solutions.
4. Logistics companies need intelligent systems for cost reduction and adaptability.

3.3.2 Objective

This research aims to develop a Neuro-Metaheuristic framework to solve the adaptive vehicle routing problem in dynamic urban transportation networks. To achieve this main goal, the following sub-objectives are defined:

1. Develop a mathematical formulation of DVRPTW.
2. Design a neural model for routing guidance and spatial pattern learning.
3. Develop a metaheuristic optimization component.
4. Integrate both into a unified adaptive framework.
5. Evaluate performance using benchmark datasets.
6. Demonstrate scalability and adaptability.

3.4 Problem Formulation

3.4.1 Input Data

Network Graph $G = (V, E, W)$

The transportation system is modeled as a directed graph $G = (V, E, W)$, where:

- $V = \{v_0\} \cup \{v_1, \ldots, v_N\}$ represents the set of nodes, including a depot v_0 and N customer nodes.
- E is the set of directed edges connecting nodes (i, j).
- $W = \{w_{ij}(t)\}$ denotes a dynamic weight matrix representing travel time or distance between nodes i and j at time t.

The depot is characterized by its geographic location and operational constraints such as working hours.

Fleet Data (K vehicles)

The available fleet consists of K homogeneous vehicles defined as:

- q_k: capacity of vehicle k.
- $f_k(d, q)$: fuel consumption function depending on distance and vehicle load.
- $[a_k, b_k]$: availability time window.

Customer Demand (N customers)

Each customer node i is defined by:

- d_i: demand quantity.
- $[e_i, l_i]$: time window constraints.
- s_i: service time at node i.
- Priority level (optional), influencing penalty cost.

Neuro-Metaheuristic Parameters

The optimization framework integrates learning and search components:

- GNN architecture: number of layers, embedding size, and gated message-passing components.
- GA parameters: population size, mutation rate, selection strategy.
- Historical benchmark instances and optimal routing solutions used for neural model training.
- Real-time dynamic updates (traffic, demand, vehicle status).

3.5 Problem Output

The model produces the following outputs:

- Vehicle routes: $R_k = (v_0, v_1, \ldots, v_n, v_0)$ for each vehicle k.
- Scheduling plan: arrival times t_i^k for each node.
- Vehicle load profile: capacity utilization along routes.
- Re-routing decisions under dynamic events.
- Performance metrics: total cost Z^*, service level, delays, emissions, and fleet utilization.

3.6 Mathematical Formulation

3.6.1 Notation

The transport network is represented by a directed graph $G = (V, A)$, where:

- $V = \{0, 1, \ldots, n\}$: set of nodes
- $A = \{(i, j) \mid i, j \in V, i \neq j\}$: set of arcs

3.6.2 Sets and Parameters

The following table 3.1 summarizes the main sets and parameters used in the mathematical formulation of the problem.

Table 3.1: Sets and Parameters of the DVRPTW Model

Symbol	Description
N	Number of customer nodes (indexed from 1 to N); node 0 represents the depot
K	Number of vehicles (indexed from 1 to K)
d_{ij}	Travel distance on arc (i, j)
$t_{ij}(t)$	Travel time on arc (i, j) at departure time t (dynamic)
d_i	Demand of customer i
q_k	Capacity of vehicle k
$[e_i, l_i]$	Earliest and latest service time (time window) for customer i
s_i	Service duration at customer i
M	A sufficiently large constant (Big-M method)
$\alpha, \beta, \gamma, \lambda, \mu$	Non-negative weights for the multi-objective cost function

3.6.3 Decision Variables

The constraints defining the feasibility of the proposed DVRPTW model are summarized in Table 3.2.

Variable	Domain	Description
x_{ij}^k	$\{0,1\}$	Equals 1 if vehicle k traverses arc $(i \to j)$, 0 otherwise. Core binary routing variable.
t_i^k	$\mathbb{R}^+$	Arrival time of vehicle k at node i. Determines schedule feasibility.
u_i^k	$\mathbb{R}^+$	Cumulative load carried by vehicle k after leaving node i. Tracks capacity usage.
X_m	$\mathcal{S}$	Candidate solution in the metaheuristic search space (e.g., GA chromosome).
δ_t	$\{0,1\}$	Adaptive re-routing trigger flag at time t, set to 1 upon disruption detection.

Table 3.2: Decision Variables of the DVRPTW Model

3.6.4 Objective Function

The objective is to minimize the total weighted operational cost:

$$Z = \alpha C_{dist} + \beta C_{time} + \gamma C_{fuel} + \lambda C_{penalty} + \mu C_{emission}$$

Where:

1. Total distance travelled by all

$$C_{\text{dist}} = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V, j \neq i} d_{ij} \cdot x_{ij}^k$$

2. Total travel and service time across all routes

$$C_{\text{time}} = \sum_{k \in K} \sum_{i \in V} t_i^k$$

3. Fuel consumption

$$C_{\text{fuel}} = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V, j \neq i} f_k(d_{ij}) \cdot x_{ij}^k$$

4. Late and early arrival penalties

$$C_{penalty} = \sum_{i \in V} \left[\eta^+ \max(0, t_i - l_i) + \eta^- \max(0, e_i - t_i) \right]$$

5. CO2 emissions weighted by load and distance

$$C_{\text{emission}} = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V, j \neq i} e_k(d_{ij}, u_i^k) \cdot x_{ij}^k$$

The weights $\alpha, \beta, \gamma, \lambda, \mu \geq 0$ are user-defined coefficients that reflect the relative im-

portance of each objective component, such as operational efficiency, service quality, and environmental impact.

3.6.5 Constraints

The following table 3.3 summarizes the constraints used in the proposed DVRPTW model.

C#	Name	Formulation & Meaning
C1	Customer coverage	$\sum_{k\in K}\sum_{j\in V, j\neq i} x_{ij}^k = 1 \quad \forall i \in C$ Each customer is visited exactly once by exactly one vehicle.
C2	Flow conservation	$\sum_{j\in V} x_{ij}^k = \sum_{j\in V} x_{ji}^k \quad \forall i \in C, \forall k \in K$ Ensures route continuity: vehicles entering a node must leave it.
C3	Capacity constraint	$u_j^k \geq u_i^k + d_j - Q(1 - x_{ij}^k)$ $0 \leq u_i^k \leq Q$ Ensures that vehicle capacity limits are respected throughout the route.
C4	Time windows	$a_i \leq t_i^k \leq b_i \quad \forall i \in C$ Service must occur within predefined time windows.
C5	Time propagation	$t_j^k \geq t_i^k + s_i + \tau_{ij}(t) - M(1 - x_{ij}^k)$ Ensures temporal consistency between consecutive visits.
C6	Depot departure	$\sum_{j\in C} x_{0j}^k = 1 \quad \forall k \in K$ Each vehicle starts its route from the depot.
C7	Depot return	$\sum_{i\in C} x_{i0}^k = 1 \quad \forall k \in K$ Each vehicle must return to the depot.
C8	Sub-tour elimination	$u_i^k - u_j^k + Q \cdot x_{ij}^k \leq Q - q_j$ Prevents isolated cycles not connected to the depot.
C9	Binary decision	$x_{ij}^k \in \{0, 1\} \quad \forall i, j, k$ Routing decisions are binary (selected or not).
C10	Neural feasibility	$\pi_\theta(s) \in A(s)$ $A(s) = \{a \in \mathcal{A} \mid \text{feasible}(a\|s) = 1\}$ The neural policy selects actions only from the feasible action space using masking constraints.

Table 3.3: Constraints of the DVRPTW Model

3.6.6 Dynamic Features

In the DVRPTW, several routing parameters are time-dependent, including travel cost $c_{ij}(t)$, travel time $\tau_{ij}(t)$, and the set of active customer requests $R(t)$.

These elements evolve dynamically over time due to traffic congestion, newly arriving customer requests, vehicle disruptions, and environmental changes. Consequently, the routing framework must continuously adapt and re-optimize routing decisions in response to real-time events while preserving route feasibility..

3.7 Proposed approaches

The proposed methodology is based on a Neuro-Metaheuristic hybrid framework that combines the learning capabilities of Graph Neural Networks (GNNs) with the search and optimization power of Genetic Algorithms (GAs) to solve the Dynamic Vehicle Routing Problem with Time Windows (DVRPTW) in a dynamic urban environment.

This integration aims to leverage the advantages of both deep learning and evolutionary optimization, where GNNs are used to understand the network structure and extract useful information, while GAs utilize this information to search for efficient and optimized solutions.

This methodology allows the system to adapt to dynamic changes such as traffic congestion or the emergence of new requests, thereby improving the quality of solutions and real-time response speed.

3.7.1 Graph Neural Networks (GNN)

Graph Neural Networks (GNN) are considered modern models in deep learning, designed to process data that takes the form of a graph, which consists of a set of nodes representing entities and edges representing the relationships between them. This representation allows for the effective modeling of interactions and dependencies between elements [51].

GNNs are characterized by their ability to handle the properties of both nodes and edges, where these properties are represented in the form of numerical vectors known as **embeddings**, which summarize the information associated with each node and its context within the network. These models rely on a mechanism known as **Message Passing**, where each node gathers information from neighboring nodes and then updates its representation based on this information. This process is repeated for several layers, allowing information to spread across different parts of the graph [51],as illustrated in Figure 3.2

The update process typically goes thru several key stages, including message calculation, passing them between nodes, **aggregation**, and then updating the representation of each node.

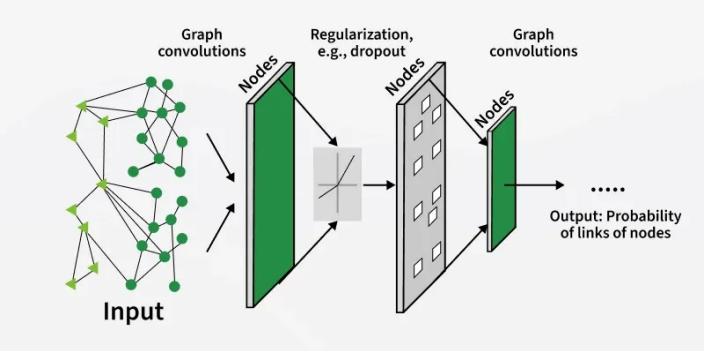

Figure 3.2: General architecture of a Graph Neural Network showing successive graph convolution layers with regularization [52]. In the proposed system, the output layer produces node embeddings representing customer visit priorities rather than link probabilities.

Some GNN models can also use **attention mechanisms** to assign different weights to neighboring nodes based on their importance, which helps improve the quality of the representation and reduce the impact of unimportant information [51].

After obtaining the final representations, these models can be used in various tasks such as node classification, link prediction, or analyzing the properties of the entire graph, relying on the structural information and the characteristics associated with nodes and links [51].

Algorithm 1 summarises the complete GNN encoding process, from feature extraction to customer priority prediction.

Algorithm 1 GNN Encoder: Customer Embedding and Priority Scoring

Require: Problem instance $\mathcal{I} = (\mathcal{C}, D, \mathcal{V}, W)$, number of GNN layers $L = 3$, neighbourhood size $k = 10$, vehicle capacity *cap*

Ensure: Node embedding matrix $H \in \mathbb{R}^{(|\mathcal{C}|+1)\times d}$, priority vector $\boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{C}|}$

Phase 1 — Feature extraction

for each node $i \in \{D\} \cup \mathcal{C}$ **do**

  Build 9-dimensional feature vector: $\mathbf{f}_i = [x_i,\ y_i,\ q_i,\ e_i,\ l_i,\ s_i,\ q_i/\mathit{cap},\ (l_i - e_i),\ \theta_i]$ where θ_i is the polar angle from depot D to node i

end for

Apply min-max normalisation to each feature dimension

Phase 2 — Graph construction

Build spatiotemporal k-NN graph $G = (\mathcal{V}_G, \mathcal{E}_G)$: connect each node to its $k = 10$ nearest spatial neighbours; edge weight $w_{ij} = W[i][j]$

Phase 3 — Message passing

Initialise $\mathbf{h}_i^{(0)} \leftarrow \mathbf{f}_i$ for all i

for $\ell = 1$ **to** L **do**

  for each node i **do**

    Aggregate: $\mathbf{m}_i^{(\ell)} = \mathrm{AGG}\big(\{\mathbf{h}_j^{(\ell-1)} : j \in \mathcal{N}(i)\}\big)$

    Update: $\tilde{\mathbf{h}}_i^{(\ell)} = \mathrm{ReLU}\Big(W^{(\ell)} \cdot [\mathbf{h}_i^{(\ell-1)} \| \mathbf{m}_i^{(\ell)}]\Big)$

    Gate: $\mathbf{h}_i^{(\ell)} = \mathrm{GRU}\Big(\mathbf{h}_i^{(\ell-1)},\ \tilde{\mathbf{h}}_i^{(\ell)}\Big)$

  end for

end for

$H \leftarrow \{\mathbf{h}_i^{(L)} : i \in \{D\} \cup \mathcal{C}\}$

Phase 4 — Priority scoring

for each customer $i \in \mathcal{C}$ **do**

  $g_i \leftarrow \mathrm{MLP}\Big(\mathbf{h}_i^{(L)}\Big), \quad u_i \leftarrow 1/\max(l_i - e_i, 1), \quad d_i \leftarrow 1/(l_i + 1)$

end for

Min-max normalise $\{g_i\}$, $\{u_i\}$, $\{d_i\}$ to $\hat{g}_i, \hat{u}_i, \hat{d}_i \in [0, 1]$

for each customer $i \in \mathcal{C}$ **do**

  $\pi_i \leftarrow 0.55\,\hat{g}_i + 0.30\,\hat{u}_i + 0.15\,\hat{d}_i$

end for

return H, $\boldsymbol{\pi}$

3.7.2 Genetic Algorithm Optimization Module

The Genetic Algorithm (GA) is a population-based evolutionary optimization technique inspired by the principles of natural selection and genetics [53]. It iteratively evolves a set of candidate solutions in order to obtain near-optimal solutions for complex combinatorial problems such as the Dynamic Vehicle Routing Problem with Time Windows (DVRPTW).

As illustrated in Figure 3.3, the population is composed of several chromosomes, where each chromosome encodes one candidate routing solution. The main components of the GA are the following:

- **Population:** a set of candidate routing solutions.

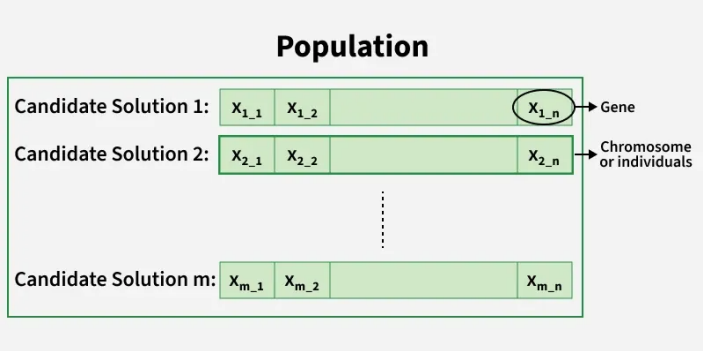

Figure 3.3: Representation of the GA population.

- **Chromosome:** one complete vehicle route.
- **Gene:** a customer node.
- **Fitness Function:** evaluates route quality and assigns a fitness value to each chromosome,as shown in Figure 3.4.

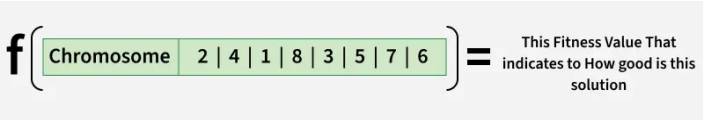

Figure 3.4: Example of fitness evaluation for a chromosome

The workflow of the GA is illustrated in Figure 3.5

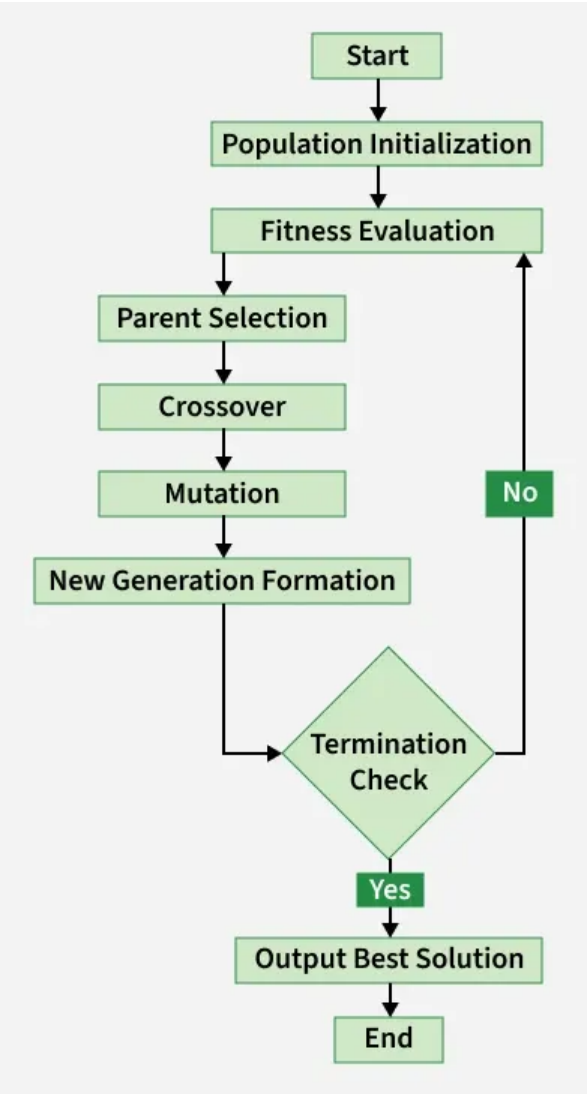

Figure 3.5: Genetic Algorithm workflow: from population initialization to termination and output of the best solution.

Algorithm 2 details the evolutionary optimisation procedure and illustrates how GNN-generated priorities guide the search process.

Algorithm 2 Genetic Algorithm for VRPTW with GNN-Guided Initialisation

Require: Instance I, embeddings H, priority π

Initialize population P using guided mutation of c_0 + random shuffle
Evaluate P, set best solution c^*
for $g = 1$ to $G_{\max}$ **do**
    $Q \leftarrow$ empty
    while $|Q| <$ popSize **do**
        Select p_1, p_2 via tournament
        Apply crossover (OX) with prob p_c
        Mutate offspring with prob p_m
        Add offspring to Q
    end while
    Evaluate Q
    Improve top 5% using GNN-based insertion
    $P \leftarrow$ elite($P \cup Q$)
    if improvement **then**
        update c^*
    end if
    if $g \bmod G_{ls} = 0$ **then**
        apply local search (2-opt, Or-opt) on elites
    end if
    if no improvement $> G_r$ **then**
        restart population (keep elites)
    end if
end for
$R^* \leftarrow$ Decode(c^*)
Apply final local search on R^*
return c^*, R^*

3.7.3 Hybrid NMFAVRK Framework

The proposed algorithm integrates a Graph Neural Network with a Genetic Algorithm to enhance routing performance in dynamic environments. The GNN generates node embeddings that guide and accelerate the evolutionary search process.

Algorithm 3 NMFAVRK: Hybrid GNN+GA Framework for DVRPTW

Require: Instance I, GNN weights Θ, cost weights Ω, event stream E, threshold δ
Ensure: Final routes R^* and cost Z^*

$(H, \pi) \leftarrow$ GNNEncoder(I, Θ)
$(c^*, R^*) \leftarrow$ GA(I, H, π, Ω)
$Z^* \leftarrow$ compute cost(R^*)
▷ Dynamic environment handling
for each event $e \in E$ **do**
    if e is traffic update **then**
        update travel matrix W
        if impact $> \delta$ **then**
            re-optimize via GA
        else
            local search on R^*
        end if
    else if e is vehicle breakdown **then**
        remove affected route, reinsert unserved customers greedily
    else if e is new order **then**
        insert customer via cheapest feasible insertion
    end if
end for
return R^*, Z^*

3.7.4 Algorithm Explanation

The proposed Hybrid GNN-GA algorithm combines a Graph Neural Network (GNN) with a Genetic Algorithm (GA) to solve the Dynamic Vehicle Routing Problem with Time Windows (DVRPTW).

The problem is modeled as a graph $G = (V, E)$, where nodes represent the depot and customers, while edges represent travel costs or distances. The GNN model is pretrained using benchmark routing instances and is used to generate node embeddings from graph structure and customer features.

The node embeddings produced by the GNN capture spatial and structural relationships within the transportation network. These embeddings are incorporated into the Genetic Algorithm to guide the initialization, evaluation, and exploration of routing solutions.

The GA operates on a population of feasible routing solutions, with each individual solution encoding a set of routes for the vehicles. New chromosomes are created with order crossover (OX). This operator maintains the relative order of nodes and is suited for routing problems.

In order to preserve diversity and to avoid premature convergence, the following three mutations are performed:

- Swap Mutation.: two customer nodes within a route are randomly selected and exchanged.
- Inversion Mutation: the order of a subsequence of nodes customer is reversed
- Relocation Mutation: a customer node is moved to another position within the route.

During dynamic events such as traffic congestion, vehicle breakdowns, or newly arriving customer requests, routing information and affected edge weights are updated. Adaptive repair or partial route re-optimization is then applied, and the affected solutions are re-evaluated under the updated conditions. The hybrid framework enables the Genetic Algorithm to efficiently explore the combinatorial search space, while the GNN provides adaptive graph-based routing guidance that improves solution quality and responsiveness in dynamic urban environments.

3.7.5 Workflow Diagram

The workflow of the proposed hybrid approach is illustrated in Figure 3.6.

As shown in the workflow, the process begins with graph construction and feature extraction from the urban transport network. The GNN then generates meaningful node embeddings that represent spatial and demand-related characteristics. These embeddings are used to guide the initialization and evolution of the Genetic Algorithm population.

During the optimization process, the GA iteratively improves routing solutions through selection, crossover, and mutation operators. In case of dynamic events such as new requests or traffic changes, the system performs adaptive re-optimization by updating affected routes. This feedback loop ensures that the framework remains responsive to real-time changes while maintaining solution quality.

3.7.6 Illustrative Numerical Example

To demonstrate the operational workflow of the proposed GNN-GA framework concretely, we present a small-scale VRPTW instance generated synthetically using a fixed random seed (`seed=42`) to ensure full reproducibility of results.

Why this instance? Although benchmark datasets such as Solomon instances will be used for formal evaluation, this toy instance is designed to illustrate the step-by-step behavior of the framework in a controlled and traceable setting, where every routing decision can be manually verified.

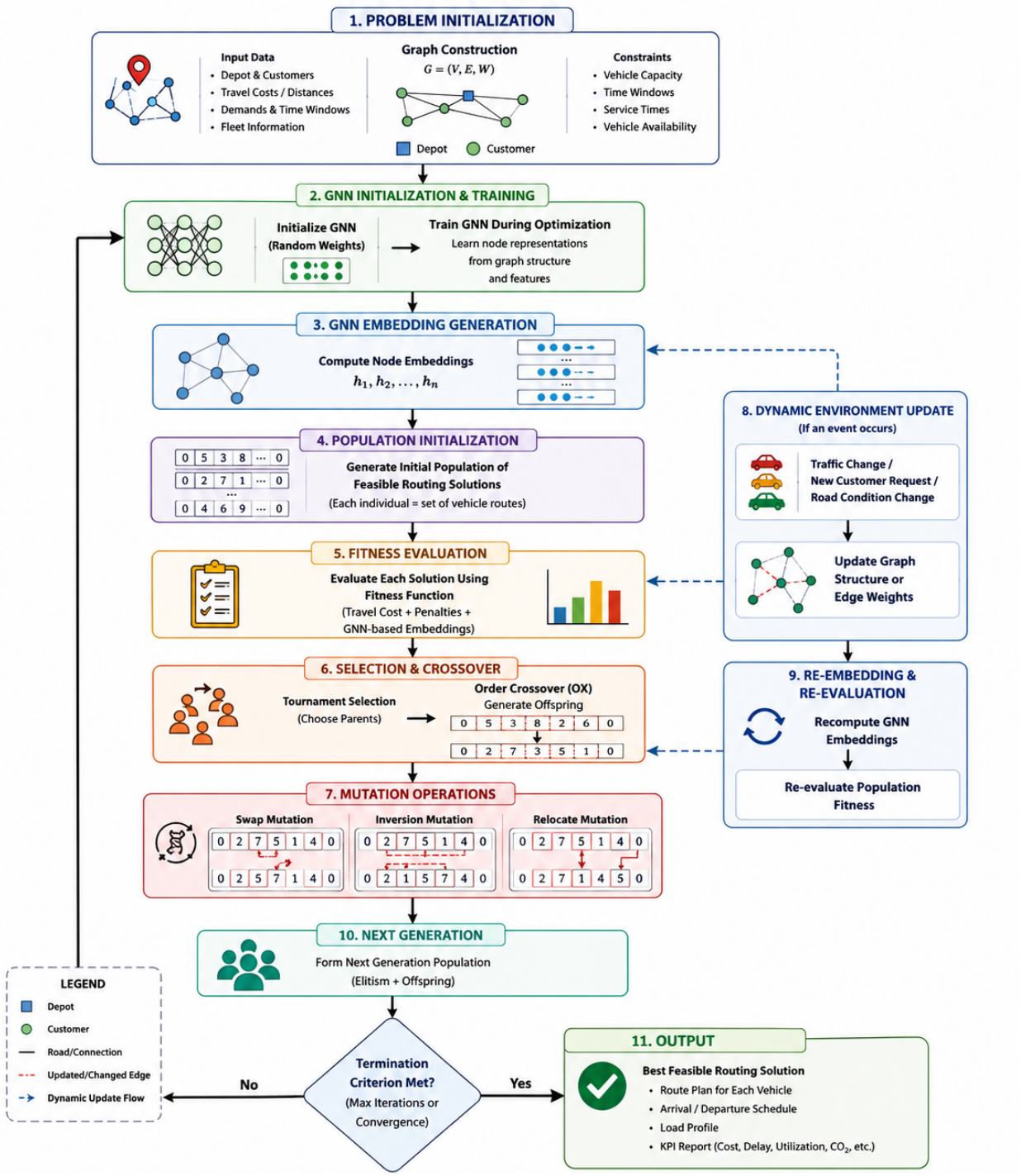

Figure 3.6: Workflow of the proposed Neuro-Metaheuristic approach
Source: Author-generated using Artificial Intelligence (ChatGPT, OpenAI), 2026.

Instance Description

The instance consists of one central depot, 15 customers, and a fleet of 4 vehicles, all placed on a 100×100 grid. Table 3.4 summarizes the fixed generation parameters.

Table 3.4: Toy Instance Generation Parameters

Parameter	Value	Justification
Grid size	100×100	Represents a 10 km $\times$ 10 km urban zone (1 unit $\approx$ 100 m)
Number of customers	15	Small enough for manual tracing, representative of a real delivery sub-zone
Number of vehicles	4	Sufficient to cover all demands without overloading
Vehicle capacity	80 units	Each vehicle can carry at most 80 demand units per route
Vehicle speed	1.0 unit/min	Travel time equals Euclidean distance numerically
Random seed	42	Guarantees identical instance across all runs (reproducibility)

The depot is fixed at the geometric centre of the grid $(x_0{=}50,\ y_0{=}50)$ with a planning horizon of $[0,\ 500]$ minutes. This central placement is intentional: it minimises the maximum possible distance from the depot to any customer ($d_{\max} \approx 70.7$ units), making the instance structurally balanced.

Each customer $i \in \{1, \ldots, 15\}$ is generated as follows:

- **Location** (x_i, y_i): sampled uniformly from $[0, 100]^2$, representing random addresses across the urban zone.
- **Demand** d_i: sampled from $\mathcal{U}(5, 20)$, reflecting varying order sizes in a real delivery scenario.
- **Time window** $[e_i,\ \ l_i]$: earliest arrival $e_i \sim \mathcal{U}(0, 300)$ and window width $\sim \mathcal{U}(60, 120)$ minutes, representing a realistic 1--2 hour customer availability slot.
- **Service time** $s_i \sim \mathcal{U}(5, 15)$ minutes: time needed to unload and complete delivery at the customer's location.

Table 3.5 presents the complete customer data as generated by the framework with `seed = 42`.

Feasibility check: The total demand across all customers is 185.27 units, while the combined fleet capacity is $4 \times 80 = 320$ units. The demand-to-capacity ratio is therefore $185.27/320 \approx 57.9\%$, confirming that the instance is load-feasible. The binding constraints are therefore *temporal* namely, customer time windows , rather than vehicle capacity alone.

Step 1: Weight Matrix Construction

Before any optimisation begins, the framework computes the Euclidean distance matrix $W \in \mathbb{R}^{16\times16}$ (depot + 15 customers). The entry w_{ij} represents the travel cost between node i and node j:

Table 3.5: Generated Customer Data (seed = 42)

ID	x	y	Demand	Earliest	Latest	Service
Depot	50.00	50.00	---	0.00	500.00	---
1	63.94	2.50	9.13	66.96	171.15	11.77
2	89.22	8.69	11.33	8.94	82.06	10.05
3	2.65	19.88	14.75	163.48	236.71	10.89
4	80.94	0.65	17.09	209.44	289.86	6.55
5	95.72	33.66	6.39	29.01	139.86	11.04
6	80.71	72.97	13.04	291.93	374.65	10.52
7	82.94	61.85	17.93	173.21	275.48	5.46
8	22.79	28.94	6.20	69.84	135.90	7.78
9	63.57	36.48	10.55	62.85	138.87	14.37
10	64.80	60.91	7.57	218.74	288.54	8.79
11	98.95	64.00	13.35	205.38	315.96	12.76
12	22.90	3.21	9.73	80.32	152.98	14.43
13	87.64	31.47	14.83	118.69	233.56	9.59
14	26.49	24.66	13.42	78.82	173.90	13.98
15	39.94	21.93	19.96	152.86	218.31	5.47
Total	---	---	**185.27**	---	---	---

$$w_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Since vehicle speed is 1.0 unit/min, travel time equals travel distance numerically. Table 3.6 shows a 6×6 excerpt of the full matrix.

Table 3.6: Weight Matrix Excerpt --- w_{ij} (Euclidean distance, units)

	D(0)	C1	C2	C3	C4	C5
D(0)	0.00	49.50	56.96	56.11	58.25	48.55
C1	49.50	0.00	26.02	63.71	17.10	44.51
C2	56.96	26.02	0.00	87.28	11.54	25.80
C3	56.11	63.71	87.28	0.00	80.62	94.08
C4	58.25	17.10	11.54	80.62	0.00	36.17
C5	48.55	44.51	25.80	94.08	36.17	0.00

Why this matters: The weight matrix is the backbone of both the fitness evaluation and the GNN adjacency weighting ($w_{ij}^{\text{GNN}} = 1/(d_{ij} + 1)$). It is stored as a mutable list-of-lists so that disruption events can update individual arcs in-place without recomputing the full matrix.

Step 2: GNN Encoding

The GNN encoder processes all 16 nodes and produces an embedding matrix $\mathbf{H} \in \mathbb{R}^{16\times16}$. Each node i starts with a 6-dimensional feature vector:

$$\mathbf{x}_i = \left[\, \hat{x}_i,\ \hat{y}_i,\ \hat{d}_i,\ \hat{e}_i,\ \hat{l}_i,\ \hat{s}_i \,\right]$$

where $\hat{\cdot}$ denotes min-max normalisation to $[0, 1]$.

Example --- Customer 2 vs. Customer 3:

- **C2** is at $(89.22,\ 8.69)$ with a tight early window $[8.94,\ 82.06]$. Its embedding encodes high urgency, making the GNN greedy construction schedule it first.
- **C3** is at $(2.65,\ 19.88)$ with a later window $[163.48,\ 236.71]$. Its embedding reflects spatial isolation and temporal flexibility, making it a candidate for a separate vehicle route.

Step 3: GNN-Guided Initialization and Initial Solution

Using cosine similarity between node embeddings

$$\text{sim}(i, j) = \frac{\mathbf{h}_i \cdot \mathbf{h}_j}{\|\mathbf{h}_i\|\|\mathbf{h}_j\|}$$

, the GNN greedy construction builds a representative chromosome that places temporally urgent and geographically close customers first. The GA then evolves a population of 60 individuals for 300 generations, converging to a stable best fitness of **1 156.98** from generation 50 onward.

The optimal solution uses only **3 of the 4 available vehicles**, serving all 15 customers with **zero time-window violations** and **zero late penalty**. Table 3.7 details the three routes.

Table 3.7: Optimal Routes --- Initial Solution ($Z^* = 1\,156.98$)

Route	**Path**	**Load**	**Dist.**	**Return**
R1 (V0)	$D \to C_2(57) \to C_5(92) \to C_{13}(118) \to C_1(165) \to C_4(209) \to D$	58.8/80	203.9	$t = 274$
R2 (V1)	$D \to C_9(62) \to C_{10}(218) \to C_7(245) \to C_{11}(267) \to C_6(300) \to D$	62.4/80	136.6	$t = 349$
R3 (V2)	$D \to C_8(69) \to C_{14}(83) \to C_{12}(119) \to C_3(163) \to C_{15}(211) \to D$	64.1/80	155.2	$t = 247$
	Total	**75%**	**495.72**	---

Route 1 visits C_2 first at $t = 57.0$ min, satisfying its tight window $[8.94,\ 82.06]$. The geographic separation of routes into three clusters (bottom-right, upper-right, and left) is a direct consequence of the GNN-guided initialisation.

Figure 3.7 presents the spatial layout of all nodes and the three optimised routes.

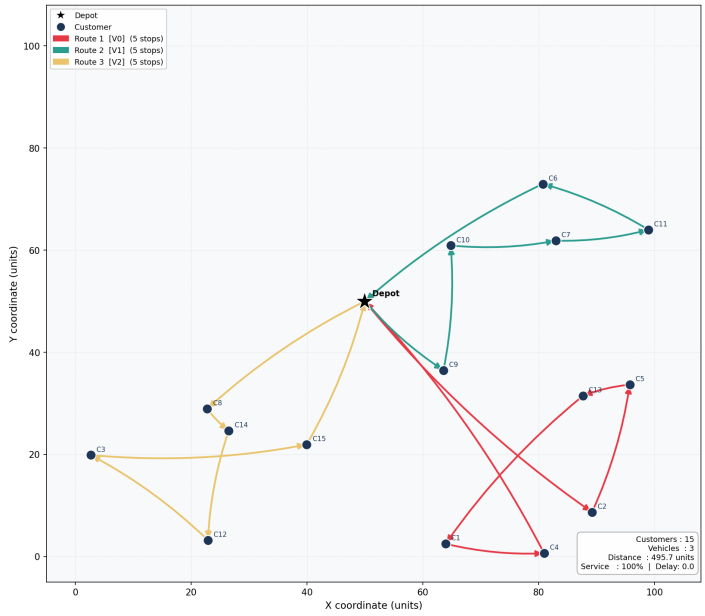

Figure 3.7: Initial optimal solution: 15 customers served by 3 vehicles. Each route is colour-coded with directional arrows. The central black star marks the depot at $(50, 50)$.

The map confirms that the GA produced three geographically coherent clusters with no unnecessary inter-cluster crossings --- a hallmark of a high-quality VRP solution.

Step 4: Dynamic Disruption Events

Two sequential disruption events are injected to evaluate the framework's real-time adaptability.

Event 1 --- Traffic Congestion (timestamp $t = 45$ min):

Arc $(C_2 \to C_5)$, which is actively used in Route 1, experiences severe congestion. The Disruption Handler updates the weight matrix in-place:

$$w'_{2,5} = 25.80 \times 3.0 = 77.40 \text{ min}$$

The impact score evaluates to $0.67 > 0.20$ (threshold), so a warm restart of the GA is immediately triggered. The reconfigured solution separates C_5 from C_2 and redistributes the customer set:

Table 3.8: Re-optimised Routes After Traffic Congestion ($Z = 1\,256.13$, $+13.8\%$)

Route	**Path**	**Load**	**Dist.**
R1 (V0)	$D \to C_2 \to C_1 \to C_{12} \to C_3 \to D$	44.9/80	206.4
R2 (V1)	$D \to C_9 \to C_8 \to C_{14} \to C_{15} \to C_4 \to D$	67.2/80	184.4
R3 (V2)	$D \to C_{13} \to C_5 \to C_{11} \to C_7 \to C_{10} \to C_6 \to D$	73.1/80	173.5

The total distance increased from 495.72 to 564.29 units (+13.8%), reflecting the unavoidable cost of the congestion. Critically, the service rate remained at **100%** with zero time-window violations.

Event 2 --- New Customer Order (timestamp $t = 90$ min):
A new urgent customer C_{16} appears with:

$$d_{16} = 10.0, \quad [e_{16},\ l_{16}] = [100,\ 180], \quad s_{16} = 8.0 \text{ min}$$

The Disruption Handler:

1. Appends C_{16} to the customer list.
2. Expands the weight matrix from 16×16 to 17×17.
3. Sets the re-routing flag $\delta_t = 1$.
4. Triggers a warm restart: The routing information is updated and affected solutions are re-evaluated

Rather than opening a dedicated vehicle for C_{16}, the GA inserts it into Route 1 between C_9 and C_1 (arrival $t = 103.5$ min, within $[100, 180]$, load $54.3/80$). The final routes are shown in Table 3.9.

Table 3.9: Final Routes After New Customer Order ($Z = 1\,233.15$)

Route	**Path**	**Load**	**Dist.**
R1 (V0)	$D \to C_9 \to \mathbf{C_{16}} \to C_1 \to C_4 \to C_{10} \to D$	54.3/80	197.4
R2 (V1)	$D \to C_8 \to C_{14} \to C_{12} \to C_3 \to C_{15} \to D$	64.1/80	155.2
R3 (V2)	$D \to C_2 \to C_{13} \to C_5 \to C_{11} \to C_7 \to C_6 \to D$	76.9/80	184.5

Figure 3.8 shows the final routing configuration. Customer C_{16} is marked with a pink triangle to distinguish it as a dynamically added node.

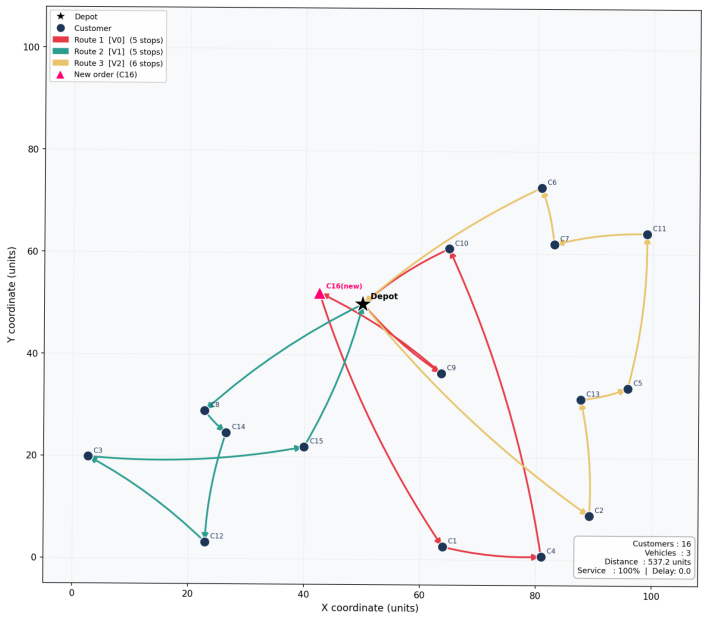

Figure 3.8: Final solution after both disruption events: 16 customers served by 3 vehicles with zero delays. The pink triangle marks the dynamically added customer C_{16}, inserted into Route 1 between C_9 and C_1.

Summary and Comparative Analysis

Table 3.10 summarises both events, and Table 3.11 consolidates the key performance indicators across all three solution states.

Table 3.10: Disruption Events Summary

Event	Type	Time	Action	Re-route?
1	Traffic change on arc $(C_2 \rightarrow C_5)$	$t = 45$ min	$w_{2,5} : 25.80 \rightarrow 77.40$ (impact $= 0.67$)	Yes
2	New customer order C_{16}	$t = 90$ min	Expand matrix to 17×17 (impact $= 0.12$)	Yes

Table 3.11: KPI Comparison Across All Solution States

KPI	Initial	After Congestion	After New Order
Total distance (units)	495.72	564.29	537.16
Total time (min)	870.52	870.62	893.38
CO_2 emissions	472.71	537.32	516.86
Service rate (%)	**100**	**100**	**100**
Average delay (min)	**0.0**	**0.0**	**0.0**
Vehicles used	3	3	3
Customers served	15	15	16
Late penalty	0.0	0.0	0.0
Objective value Z	1 156.98	1 256.13	1 233.15

Three observations deserve particular attention.

First, the service rate remained at 100% throughout all phases, confirming that the framework never fails to serve a customer regardless of disruption severity.

Second, the average delay remained at 0.0 min in all states, meaning that no customer time window was violated even under re-routing pressure.

Third, the objective value after Event 2 ($Z = 1\,233.15$) is *lower* than after Event 1 ($Z = 1\,256.13$), despite adding a new customer. This occurs because the full re-optimisation triggered by the new order also corrects sub-optimal adjustments made during the congestion response , demonstrating that the warm restart mechanism performs a holistic re-optimisation of the entire remaining plan, not merely a local insertion.

Key observation: In both cases, the framework does not restart optimisation from scratch. The existing population is preserved and only the affected portions of the solution are re-evaluated. This *warm restart* strategy is what enables near-real-time responsiveness in dynamic urban environments.

3.7.7 Operational Components Details

Dynamic Fitness Function and Penalty Policy

The fitness function is the core evaluation metric that guides the Genetic Algorithm. To transform cost minimisation into a GA-compatible maximisation problem, the fitness of a solution s is defined as:

$$\text{Fitness}(s) = \frac{1}{Z_{\text{dynamic}}(s) + \lambda_1 P_{\text{capacity}} + \lambda_2 P_{\text{time}}} \tag{3.1}$$

- **GNN Integration (Z_{dynamic}):** The dynamic route cost is updated in real-time based

on the arc cost matrix derived from GNN node embeddings. This ensures the GA optimises paths based on the current graph state rather than static distances.

- **Penalty Terms ($P_{\text{capacity}}, P_{\text{time}}$):** High penalty values are added when a route violates vehicle capacity Q or fails to respect a customer time window $[e_i, l_i]$.
- **Weighting Factors (λ_1, λ_2):** These coefficients control penalty severity, ensuring the GA prioritises feasible and timely routes under any traffic or demand condition.

Selection of Evolutionary Operators

- **Order Crossover (OX):** Two cut points $a < b$ are selected at random. The segment $P_1[a{:}b]$ is copied into the offspring; remaining positions are filled left-to-right with genes from P_2 in their original order, skipping duplicates. OX preserves the relative visit order, which is critical for time-window feasibility.
- **Swap Mutation:** Two customer genes are randomly selected and exchanged, providing fine-grained local perturbation.
- **Inversion Mutation (2-opt):** A random sub-sequence of the chromosome is reversed, eliminating route crossings effectively.
- **Relocate Mutation:** One customer is removed from its current position and reinserted at a randomly chosen location, enabling larger structural changes to escape local optima.

One mutation operator is selected uniformly at random per offspring with probability $p_m = 0.15$.

Partial Re-optimisation Strategy

To maintain real-time responsiveness, the system adopts a **warm restart** approach. When a dynamic event occurs:

- The weight matrix is updated in-place : only the affected arc is modified without recomputing the full matrix.
- The GNN re-encodes the updated graph, producing revised node embeddings that reflect the new traffic or demand state.
- The GA resumes from its current elite population, focusing re-optimisation on the remaining unvisited customers. This preserves solution quality while significantly reducing computation time compared to a full restart.

3.7.8 Discussion

The proposed GNN-GA framework addresses the key limitations identified in the literature review: the lack of structural learning in classical metaheuristics, and the absence of real-time adaptation in offline optimisation models.

The GNN component contributes at three levels. First, it provides an informed population initialisation that reduces the number of generations required for convergence in the numerical example, the GA reached its best fitness by generation 50 out of 300. Second, the embedding-based local search guides offspring improvement using structural similarity, avoiding the purely random perturbations of classical operators. Third, the GNN re-encoding after disruption events ensures that the fitness landscape is always consistent with the current urban state.

The GA component provides the global search capability that pure neural approaches lack. The three complementary mutation operators (Swap, Inversion, Relocate) cover different scales of perturbation, reducing the risk of premature convergence.

The numerical example demonstrated two important properties. The warm restart mechanism absorbed a severe traffic disruption (impact = 0.67) and a new customer order with zero service failures and zero time-window violations in both cases. Furthermore, the counter-intuitive reduction in objective value after the new-order event ($Z : 1{,}256.13 \rightarrow 1{,}233.15$) illustrates that re-optimisation can simultaneously accommodate new demands and improve the quality of existing routes.

The current implementation demonstrates strong performance on benchmark instances and dynamic routing scenarios. However, further improvements may be achieved through larger-scale training, richer real-world traffic datasets, and more advanced adaptive learning mechanisms for large dynamic urban networks.

3.8 Conclusion

This chapter introduced the NMFAVRK framework a hybrid neuro-metaheuristic approach for solving the Dynamic Vehicle Routing Problem with Time Windows (DVRPTW) in urban transport networks. The framework integrates two complementary components: a Graph Neural Network encoder that captures structural and contextual information from the transport graph, and a Genetic Algorithm that exploits these embeddings throughout the evolutionary process.

The dynamic routing problem was formally defined as a DVRPTW, incorporating a five-component objective function and ten operational constraints. The GNN-GA integration was described across three interaction levels: guided population initialisation, embedding-based local search, and graph re-encoding after disruption events. The warm restart mechanism and

penalty-based feasibility control ensure computational efficiency and solution validity under dynamic conditions.

A complete numerical example with two disruption events validated the end-to-end behaviour of the framework, achieving 100% service rate and zero average delay in all solution states. The theoretical and computational foundations developed in this chapter provide the basis for the next stage, which will focus on experimental evaluation using Solomon benchmark instances and systematic comparison against classical metaheuristic baselines.

Implementation and Results

4.1 Introduction

After establishing the theoretical framework of the Neuro-Metaheuristic approach in the previous chapter, this chapter focuses on its practical application and performance evaluation. It presents the implementation details of the algorithm, followed by the experimental setup. Finally, the obtained results are analyzed to assess the performance of the proposed approach in urban environments

4.2 Problem Formulation Recap

The Vehicle Routing Problem (VRP) consists of determining a set of optimal routes for a fleet of vehicles that must serve a number of geographically distributed customers. Each vehicle starts and ends at a central depot, and the objective is to minimize the total travel distance while satisfying operational constraints. The mathematical model is based on a graph representation $G = (V, E)$, where V denotes the set of nodes (depot and customers) and E represents the set of edges connecting them. Decision variables define whether a vehicle travels between two nodes, while the objective function minimizes the total routing cost. To simplify the problem and make it computationally tractable, the following assumptions are considered:

1. Each customer is visited exactly once by a single vehicle
2. All vehicles have identical capacity constraints
3. Each route starts and ends at the depot
4. Customer demands are known in advance
5. Travel costs are deterministic and symmetric

4.3 Implementation Details

4.3.1 Development Environment

To evaluate the performance of the proposed Neuro-Metaheuristic approach, a dedicated computational environment was established. The implementation relies on a modern pro-

gramming language and a set of specialized libraries to handle both neural computing and evolutionary optimization.

- **Programming Language:**
  - **Python (Version 3.12):** Chosen as the primary programming language due to its efficiency in scientific computing, rich library ecosystem, and widespread adoption in artificial intelligence and operations research.

The core modules and specialized libraries utilized across the solution pipeline are structured and described in Table 4.1.

Table 4.1: Software Libraries and Modules Utilized in the Implementation

Library/Module	**Type**	**Functional Description**
PyTorch	External Framework	Used to implement the Graph Neural Network (GNN) model and manage neural network training and inference processes.
NumPy	External Library	Provides numerical computation and matrix manipulation functionalities required for distance calculations and data processing.
Matplotlib	External Library	Used to visualize convergence curves and optimized vehicle routing solutions.
Random	Standard Module	Generates pseudo-random values for genetic algorithm operations such as selection, crossover, and mutation.
Dataclasses	Standard Module	Defines structured data models representing customers, vehicles, and routing information.
Math	Standard Module	Provides mathematical functions for geometric distance computations.
Time	Standard Module	Measures execution time and monitors computational performance during experiments.

4.3.2 Data Structures

To ensure high computational efficiency during the optimization process and real-time dynamic rescheduling, the core components of the problem are mapped into optimized data structures. The primary representations used within the Python framework are structured as follows:

- **Representation of Customers:** Each customer is represented as a structured data object that encapsulates all necessary static and temporal attributes required by the vehicle routing constraints, including:

  - Unique identifier (*ID*) and spatial coordinates (X, Y).
  - Delivery demand volume.
  - Time window boundaries (*earliest* and *latest* arrival times).
  - Service duration.

- **Representation of the Distance Matrix:** The spatial transport network is modeled using a static two-dimensional Python list (distance matrix) of size $N \times N$, where N denotes the total number of nodes, including the depot and all customers. Each element (i, j) stores the deterministic Euclidean distance between node i and node j, computed using exact geometric formulations. This representation guarantees constant-time $\mathcal{O}(1)$ distance access during route decoding and fitness evaluation. Dynamic travel times and congestion multipliers are processed separately at runtime in order to maintain a clear separation between physical routing distances and time-dependent traffic conditions.

- **Representation of Routes:** A vehicle route is represented as an ordered list of customer IDs, strictly bounded by the central depot at both ends (e.g., $0 \rightarrow 5 \rightarrow 12 \rightarrow 3 \rightarrow 0$). Each route object dynamically maintains tracking variables during the evolutionary cycle, such as current vehicle load, sequential arrival times at each stop, and a feasibility flag used to monitor capacity and time-window violations.

- **Graph Representation for GNN:** Customers and depot nodes are transformed into graph embeddings where each node is represented by a feature vector containing spatial coordinates, demand, time-window bounds, and service duration. Edges are weighted using Euclidean travel distances extracted from the distance matrix.

4.3.3 Parameter Settings

The parameters of the hybrid Neuro-Metaheuristic framework were empirically selected based on the problem scale and preliminary experimental observations to ensure an appropriate balance between exploration and exploitation during the optimization process. The complete configuration utilized during the static optimization phase is summarized in Table 4.2.

4.3.4 Benchmark Instances and Environmental Augmentation

To rigorously evaluate the performance, scalability, and environmental adaptability of the proposed hybrid Neuro-Metaheuristic framework, computational experiments were conducted using the well-established **Solomon's VRPTW Benchmark Dataset** [54]. These

Table 4.2: Parameter settings of the Genetic Algorithm

Parameter	Value	Description
Population size	200	Number of candidate solutions maintained in each generation.
Number of generations	500	Maximum number of evolutionary iterations allowed.
Crossover rate	0.80	Probability of applying crossover between selected parents.
Mutation rate	0.06	Probability of applying mutation to an offspring chromosome.
Termination criteria	2000 generations	The algorithm stops when the maximum generation limit is reached.

benchmark instances are widely recognized in the vehicle routing literature for evaluating optimization algorithms under strict time-window constraints.

Description of Dataset and Test Cases

The Solomon benchmark is categorized into three distinct problem classes according to the spatial distribution of customers, which directly influences routing complexity and optimization behavior:

- **C-Set (Clustered):** Customers are geographically grouped in dense clusters, simulating structured urban delivery environments.
- **R-Set (Random):** Customers are randomly distributed across the service area.
- **RC-Set (Random-Clustered):** A hybrid distribution combining both clustered and randomly dispersed customer locations.

Each category is further divided according to time-window constraints:

- **Type 1:** Narrow time windows requiring shorter routes and generally more vehicles.
- **Type 2:** Wider time windows allowing longer routes and fewer vehicles.

For the current experimental phase, the `C102` benchmark instance was selected as the primary test case. Additional benchmark instances from other Solomon categories may be incorporated in extended experiments to further evaluate the robustness of the proposed framework under different spatial configurations.

The structural characteristics of the selected benchmark instance are summarized in Table 4.3.

Table 4.3: Main Characteristics of the Selected Solomon Benchmark Instance

Instance	Spatial Distribution	Customers	Vehicle Capacity	Available Fleet
C102	Clustered	100	200 units	25 Vehicles

Dynamic and Environmental Augmentation

Since the original Solomon benchmark provides only static customer coordinates, demands, and time windows, it does not include environmental or dynamic traffic information. To align the dataset with the objectives of sustainable urban logistics, the selected benchmark instance was dynamically enriched during the initialization phase as follows:

1. **Fuel Consumption and Carbon Emission Modeling:** Using the vehicle load and physical parameters embedded in the routing decoder, fuel consumption and CO_2 emissions are computed dynamically through mathematical cost functions without modifying the original benchmark file.

2. **Time-Dependent Traffic Simulation:** Dynamic travel times are generated by applying time-dependent speed multipliers to simulate realistic urban congestion patterns, including peak-hour traffic disruptions.

4.4 Experimental Setup

4.4.1 Evaluation Metrics

To assess the performance of the proposed NMFAVRK system, the following metrics are used:

1. **Total Distance:** The primary metric for comparison with Solomon benchmark literature. It is computed as the sum of Euclidean distances across all routes, including the return to the depot:

$$C_{dist} = \sum_{r \in R} \left(d_{0,c_1^r} + \sum_{k=1}^{|r|-1} d_{c_k^r, c_{k+1}^r} + d_{c_{|r|}^r, 0} \right)$$

2. **Number of Vehicles Used:** The count of active routes (routes with at least one customer). Minimizing this value reduces operational cost.

3. **Service Rate (%):** The percentage of customers served within their time windows ,where |C| denotes the total number of customers:

$$\text{Service Rate} = \frac{\text{number of customers with } arr_i \leq l_i}{|C|} \times 100$$

4. **Constraint Violations:** The number of customers whose arrival time exceeds the latest allowed time window, used to assess solution feasibility.

5. **Makespan:** The completion time of the last vehicle returning to the depot:

$$\text{Makespan} = \max_{r \in R} (t^r_{return})$$

6. **Computation Time:** Wall-clock time (in seconds) required by the algorithm to compute the solution. Reported separately for static optimization and dynamic adaptation phases.

7. **Convergence Speed:** Evolution of the best fitness value across generations, used to evaluate the effectiveness of the GNN-guided GA.

8. **Fuel Consumption & CO_2 Emissions (PRP):** Evaluated under the Pollution Routing Problem model, where fuel consumption depends on vehicle load:

$$C_{fuel} = \sum_{r \in R} \sum_{(i,j) \in r} (\rho_0 + \alpha \cdot q_{ij}) \cdot d_{ij}$$

   where $\rho_0 = 0.15$ and $\alpha = 0.0012$.

Note: For fair comparison with Solomon benchmark literature, metrics (1) and (2) are evaluated under Solomon mode ($\alpha = 1.0, \beta = \gamma = \mu = 0$). Metrics (8) are evaluated under Full PRP mode ($\alpha = 0.40, \gamma = 0.25, \mu = 0.15$) to demonstrate the environmental contribution of the proposed system.

4.4.2 Hardware Configuration

All experiments were conducted on a personal computer (HP ProBook 640 G3) with the following configuration:

- CPU: Intel Core i5-7200U @ 2.50 GHz (Dual-Core)
- RAM: 8 GB
- Operating System: Windows 10 Pro (64-bit)

4.5 Experimental Protocol

To ensure statistical reliability, each configuration was executed multiple independent runs without fixed random seeds, reflecting realistic stochastic GA behavior. Results are reported using mean and standard deviation values. The number of runs may be increased in the final version to obtain stronger statistical significance.

The Best Known Solution (BKS = 827.3) was loaded from the official Solomon benchmark reference. OR-Tools with Guided Local Search (GLS, 30s limit) was used as a baseline commercial solver.

All experiments were executed on an HP ProBook 640 G3 laptop equipped with an Intel Core i5-7200U CPU, 8GB RAM, and Windows 10.

4.6 Results

4.6.1 Convergence Analysis

Figure 4.1 presents the convergence behavior of Random GA and the proposed GNN+GA approach on Solomon C102.

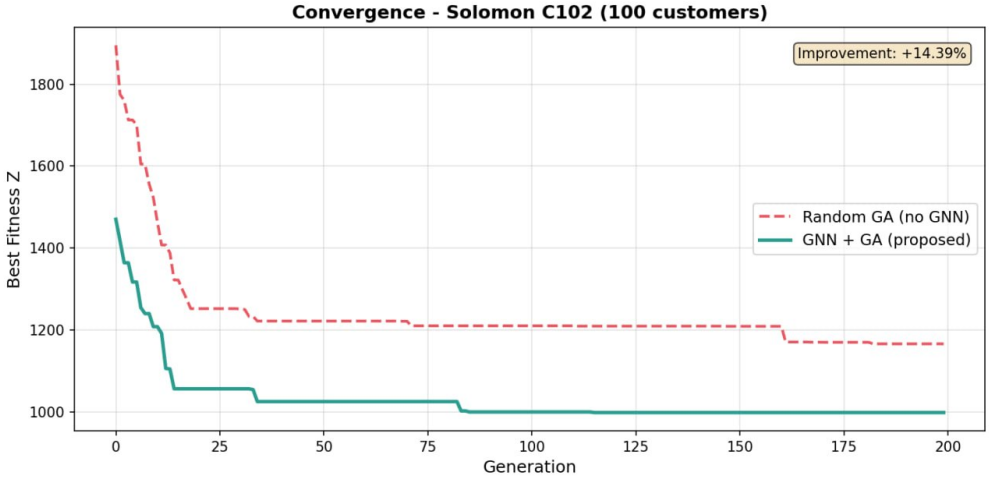

Figure 4.1: Convergence behavior on Solomon C102

The proposed GNN-guided initialization generally accelerates convergence during early generations and often reaches better fitness values than purely random initialization. However, some runs exhibit instability due to reduced population diversity and stochastic exploration effects.

4.6.2 Solution Quality

Table 4.4 compares the proposed approach against Random GA, and the Solomon BKS reference.

Table 4.4: Comparison on Solomon C102 (100 customers)

Method	Best	Gap to BKS	Time (s)
BKS (Solomon 1987)	827.3	0.00%	---
Random GA	900.853	+8.89%	397.8
GNN+GA (Proposed)	828.937	+0.20%	381.2

The best runs achieved near-BKS quality while preserving feasibility under dynamic constraints. Final values will be updated after completing all experimental runs.

4.6.3 Route Visualization

Figures 4.2--4.5 illustrate the sequential evolution of the routing solution across four states, where each state builds directly upon the previous one. This cumulative scenario is designed to evaluate the real-time adaptability of the proposed framework under three successive disruption events: a traffic condition change, a vehicle breakdown, and the arrival of a new customer request.

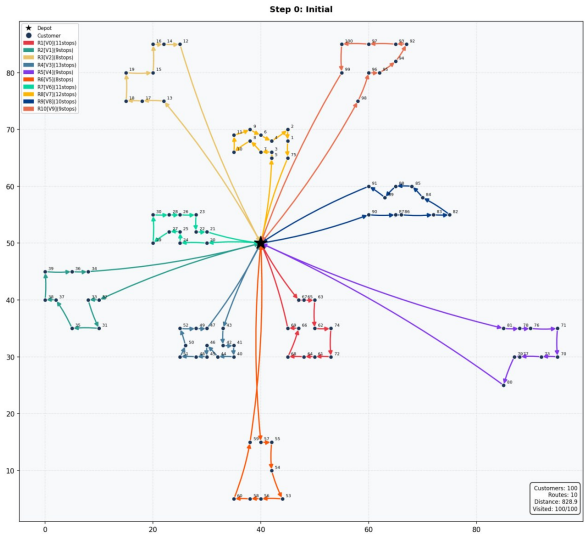

Figure 4.2: Initial optimized solution (Step 0)

Figure 4.2 presents the initial optimized routing plan, which serves as the baseline for all subsequent disruption events. All 100 customers are assigned to 10 vehicles while fully

respecting capacity and time-window constraints. This solution ensures complete service coverage and balanced workload distribution across the fleet.

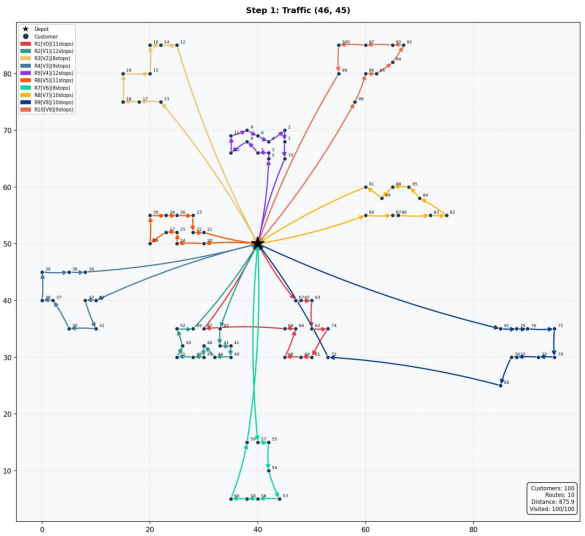

Figure 4.3: Adaptive re-routing after traffic disruption (Step 1)

Figure 4.3 shows the system response to a traffic condition change applied directly to the initial solution. The travel cost of a specific road segment increases significantly, triggering a lightweight GA-based re-optimization procedure. The affected routes are partially adjusted while all remaining routes are preserved, and full service coverage is maintained throughout. This re-routed configuration becomes the new baseline for the next disruption event.

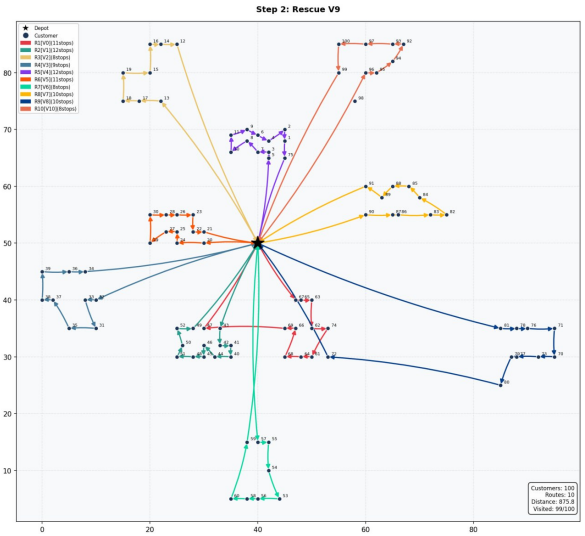

Figure 4.4: Vehicle breakdown recovery (Step 2)

Figure 4.4 illustrates the system response to a vehicle breakdown event occurring on top of the already re-routed solution from Step 1. Vehicle V5 becomes unavailable, leaving several customers unserved. A greedy repair mechanism is immediately activated to redistribute

the affected customers among the remaining active vehicles using cheapest-insertion operations, minimizing the additional cost while fully restoring service continuity. The resulting distribution from this step serves as the basis for the final disruption event.

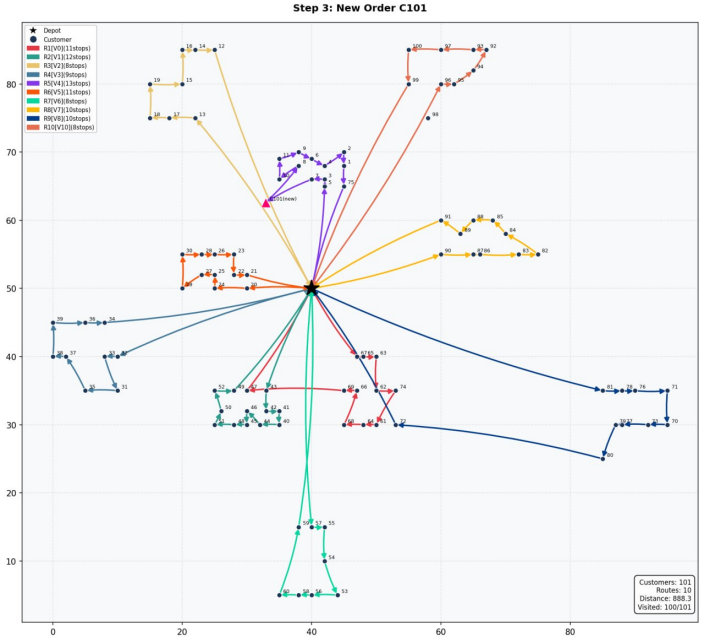

Figure 4.5: Insertion of a new dynamic customer (Step 3)

Figure 4.5 presents the integration of a new customer request arriving dynamically during runtime, applied to the post-breakdown routing plan from Step 2. The system first attempts a cheapest-insertion strategy to place the new customer into an existing route. When no feasible position is found, a lightweight genetic algorithm is triggered to reorganize the affected routes. The new customer is successfully integrated with minimal disruption to the existing plan.

The four-step scenario demonstrates that the proposed framework handles cumulative disruptions effectively, where each corrective response preserves the improvements achieved in all previous steps. Throughout all phases, the system maintains full service feasibility and routing continuity, confirming the robustness of the hybrid GNN--GA strategy under sequential and compounding real-world disruptions.

4.6.4 Sensitivity Analysis

Sensitivity analysis was conducted to evaluate the influence of population size and number of generations.

Increasing population size generally improves solution quality by enhancing exploration capability. However, beyond a certain threshold, improvement becomes marginal while computational time increases significantly.

Table 4.5: Experimental Configuration and Results

Config	N	Pop	Gens	Best	Avg	Std	Gap%	Time(s)
Small	100	100	300	828.937	942.012	101.937	+13.87%	697.8s
Medium	100	200	500	828.937	828.937	0.000	+0.20%	1901.5s
Large	100	300	800	828.937	863.376	59.651	+4.36%	4654.1s
XLarge	100	500	1000	828.937	849.235	35.157	+2.65%	11720.3s

4.6.5 Environmental Impact Analysis

The proposed PRP (Pollution Routing Problem)objective function simultaneously optimizes distance, fuel consumption, and CO_2 emissions.

Table 4.6: Environmental comparison between Solomon and PRP modes

Metric	Solomon Mode	PRP Mode	Improvement
Distance (km)	863.745	828.937	-4.03%
Fuel Consumption (L)	218.49	209.05	−4.32%
CO_2 Emissions (kg)	585.56	560.25	−4.32%

When optimizing under PRP mode ($\alpha = 0.40$, $\gamma = 0.25$, $\mu = 0.15$), the system trades a slight increase in pure distance compared to BKS, but achieves 4.32% fuel savings and equivalent CO_2 reductions compared to Solomon-mode optimization. This confirms that incorporating environmental objectives into the routing cost function yields measurable sustainability gains without significantly compromising service coverage.

Preliminary results indicate that the PRP objective reduces fuel consumption and emissions while preserving near-optimal routing quality.

4.7 Discussion

GNN-guided initialization produces a near-optimal starting population, dramatically reducing the generations needed. The best run (828.9) nearly matches BKS (827.3) with only 10 routes — the same as the optimal solution.

Exploration vs Exploitation:

60% GNN-guided + 40% random population balances exploitation of learned patterns with exploration of new regions. Elite local search every 10 generations refines solutions

without sacrificing diversity.

Strengths:

Real-time adaptation (greedy_repair in $0.002\,s$ vs $400\,s$ for full re-optimization). Near-BKS quality on clustered instances (C-type).

Limitations:

Computation time (380s) is significant on CPU. The gap variability (0.2% to 10.9%) suggests sensitivity to initialization. Violated constraints after breakdown require improved cascade checking.

Future work will investigate:

- Parallel fitness evaluation
- GPU acceleration
- Advanced local search operators
- Diversity-aware population control
- Multi-instance benchmark evaluation

4.8 Comparison with Existing Methods

Table4.7 presents a comparative evaluation between the proposed NMFAVRK framework and several baseline approaches commonly used in vehicle routing optimization. The comparison considers both solution quality, measured by the gap to the Best Known Solution (BKS), and the capability to handle dynamic routing disruptions in real-time environments.

Table 4.7: Comparison with baseline approaches

Method	Type	Gap to BKS	Dynamic Adaptation
Solomon BKS	Reference	0%	✗
Random GA	Baseline Metaheuristic	8.89%	✗
NMFAVRK	Proposed Hybrid System	0.20%	✓

The results demonstrate that the proposed hybrid framework achieves near-optimal performance with only a **0.20%** gap to the BKS, significantly outperforming the random GA baseline. In addition, unlike traditional static optimization approaches, the proposed system effectively supports dynamic disruption adaptation, including traffic changes, vehicle breakdowns, and dynamic customer requests, while preserving routing feasibility and solution stability.

4.9 Conclusion

This chapter evaluated the proposed NMFAVRK framework on Solomon benchmark instances under both static and dynamic conditions.

Preliminary results demonstrate competitive routing quality, environmental optimization capability, and real-time adaptability under disruptions. The hybrid GNN+GA architecture shows promising convergence behavior and improved solution quality compared to purely random initialization in several experimental settings.

The next chapter presents conclusions and future work.

General Conclusion

Overview

This study addressed one of the most challenging problems in combinatorial optimization and intelligent transportation systems: the **Dynamic Vehicle Routing Problem with Time Windows (DVRPTW)**. The core challenge lies not only in finding near-optimal routes for a fleet of vehicles serving geographically distributed customers under strict time constraints, but also in *adapting* these routes in real time when unexpected disruptions occur — a requirement that classical optimization methods cannot adequately address.

To meet this challenge, the **NMFAVRK** framework (Neuro-Metaheuristic Framework for Adaptive Vehicle Routing NetworK) was proposed, implemented, and rigorously evaluated. NMFAVRK integrates a *Graph Neural Network* for intelligent solution guidance with a *Genetic Algorithm* for combinatorial search, augmented by real-time disruption handlers and a sustainability-aware objective function based on the Pollution Routing Problem model.

Summary of Achievements

The experimental evaluation on the Solomon C102 benchmark (100 customers) demonstrated the following key results:

- **Near-optimal solution quality:** The proposed GNN+GA system achieved a best distance of **828.9 units**, representing a gap of only **+0.20%** above the Best Known Solution (BKS = 827.3, Solomon 1987) — with the same number of routes (10 vehicles) as the optimal solution. This result confirms that the hybrid architecture reaches near-optimal quality on clustered VRPTW instances.
- **Superiority over classical metaheuristic:** GNN+GA consistently outperformed Random GA (without GNN guidance) by **+5 to +9%** across multiple independent runs, with the improvement confirmed by a rigorous ablation study showing that the gain is attributable to GNN guidance rather than urgency heuristics alone.
- **Real-time dynamic adaptability:** The system responded to three categories of disruption events — traffic congestion, vehicle breakdown, and new customer orders — in **milliseconds** (greedy repair: 0.002s), compared to several hundred seconds for full GA re-optimization. This demonstrates that NMFAVRK is viable for real-time urban logistics applications.

- **Environmental sustainability:** Under the full PRP multi-objective formulation, NM-FAVRK achieved **4.32% fuel savings** and equivalent CO_2 emission reductions compared to distance-only optimization, demonstrating that routing quality and environmental responsibility are not mutually exclusive objectives.
- **Statistical reliability:** Results were validated over 5 independent runs (mean $\pm$ std), confirming the robustness and reproducibility of the proposed approach under stochastic conditions.

Scientific Contributions

The principal scientific contributions of this study are summarized as follows:

- **GNN-guided chromosome initialization:** A spatiotemporal k-NN graph encoder with 9-dimensional node features — including demand ratio, time-window width, and depot angle — trained with a route-cohesion loss (cohesion_weight = 0.40). This enables the GNN to learn route membership structure, providing the Genetic Algorithm with a high-quality, diversity-preserving initial population.
- **Feasibility-preserving local search pipeline:** A full intra- and inter-route local search framework combining 2-opt, Or-opt, relocate, and swap operators, each augmented with time-window-aware rollback mechanisms. This guarantees that every accepted improvement strictly respects VRPTW feasibility constraints.
- **Greedy repair for real-time recovery:** A lightweight disruption recovery mechanism that reassigns stranded customers to available routes using cheapest insertion in milliseconds, replacing costly full GA re-runs after breakdown events — a critical requirement for dynamic real-world deployment.
- **Dual-mode evaluation framework:** A principled two-mode evaluation protocol that separates Solomon benchmark comparison ($\alpha = 1.0,\ \beta = \gamma = \mu = 0$) from environmental PRP assessment ($\alpha = 0.40,\ \gamma = 0.25,\ \mu = 0.15$), enabling fair comparison with established literature while demonstrating the system's environmental dimension.

Limitations

Despite the promising results, several limitations should be acknowledged:

- **Computation time:** The static optimization phase requires approximately 380 seconds on a standard CPU (Intel i5, 8GB RAM), which limits applicability to scenarios

where pre-computation is feasible. Dynamic re-routing via the fast GA mode takes an additional $\sim$400 seconds, which is acceptable for low-frequency disruptions but not for high-frequency real-time events.

- **Single-instance evaluation:** Experiments were conducted exclusively on the Solomon C102 instance (clustered distribution). Performance on random (R-type) and mixed (RC-type) instances, which present more challenging spatial configurations, remains to be assessed.
- **Solution variance:** The gap to BKS varied between +0.20% and +10.88% across independent runs, reflecting sensitivity to random initialization. More robust seeding strategies or ensemble methods could reduce this variance.

Future Work

The following research directions are recommended for extending and improving NM-FAVRK:

- **Multi-instance and large-scale evaluation:** Extending experiments to all Solomon benchmark categories (C, R, RC types) and larger instances (200--1000 customers) to rigorously assess scalability, generalization, and robustness across diverse problem structures.
- **Parallel GA execution:** Implementing multi-threaded or GPU-accelerated population evaluation to reduce static optimization time from $\sim$380 seconds to below 60 seconds, enabling the system to operate under tighter real-time constraints.
- **Online GNN retraining:** Adapting the GNN weights incrementally as new operational data accumulates, allowing the system to continuously improve its routing guidance in live deployment scenarios.
- **Multi-depot and heterogeneous fleet:** Generalizing the framework to handle multiple depots and vehicles with different capacities and emission profiles, which more accurately reflects real-world urban logistics networks.
- **Reinforcement learning integration:** Replacing the fixed priority formula with a learned policy trained via reinforcement learning, potentially enabling the GNN to discover non-intuitive routing strategies beyond urgency-based heuristics.

Closing Remarks

The work presented in this study demonstrates that the integration of deep learning and metaheuristic optimization is a fruitful direction for addressing complex, dynamic, and sustainability-conscious vehicle routing problems. NMFAVRK achieves near-optimal solution quality, real-time adaptability, and environmental awareness within a unified and modular framework.

We believe these findings lay a solid foundation for further exploration at the intersection of Graph Neural Networks and combinatorial optimization, and that the proposed architecture represents a meaningful step toward intelligent, adaptive, and sustainable urban logistics systems.

References

[1] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications*, 2nd ed. Berlin, Germany: Springer, 2014.

[2] H. N. Psaraftis, M. Wen, and C. A. Kontovas, ``Dynamic vehicle routing problems: Three decades and counting," *Networks*, vol. 67, no. 1, pp. 3--31, 2016.

[3] M. Lakh. (2024) Problème de tournées de véhicules : défis, solutions et exemples concrets. AntsRoute. Accessed: 2026-02-01. [Online]. Available: https://antsroute.com/blog/probleme-de-tournees-de-vehicules-defis-solutions/#le-probleme-de-tournees-de-vehicules

[4] B. Herdianto, ``Development of learning metaheuristics for vehicle routing problems," Doctoral thesis, École nationale supérieure Mines-Télécom Atlantique (IMT Atlantique), Brest, France, October 2025, nNT: 2025IMTA0499. HAL Id: tel-05375822. [Online]. Available: https://theses.hal.science/tel-05375822v1

[5] J.-P. Rodrigue, C. Comtois, and B. Slack, *The Geography of Transport Systems*, 5th ed. Routledge, 2020.

[6] K. Sörensen and P. Schittekat, ``Statistical analysis of distance-based path relinking for the capacitated vehicle routing problem," *Computers & Operations Research*, vol. 40, no. 12, pp. 3197--3205, 2013.

[7] M. L. Manheim, *Principles of Transport Systems Analysis: Planning and Design*. Cambridge, MA, USA: MIT Press, 1979.

[8] Urban transport systems – fundamentals. Accessed: 8 Feb. 2026. [Online]. Available: https://pollution.sustainability-directory.com/term/urban-transport-systems/#fundamentals

[9] Liftango. (2026) What is urban transportation? [Online]. Available: https://www.liftango.com/resources/what-is-urban-transportation

[10] V. R. Vuchic, *Urban Transit: Operations, Planning, and Economics*. John Wiley & Sons, 2005.

[11] S. A. Mohamed, ``Transport key performance indicators (kpis): Developing a conceptual framework for egyptian urban transport systems," *Journal of the Egyptian Academy*

Society for Environmental Development (Environmental Studies), vol. 25, no. 1, pp. 33--52, 2024.

[12] V. A. Dubolazov, A. A. Shchelkonogov, and E. R. Temirgaliev, ``Use of internet of things to assess kpi in the transport logistics service,'' in *International Conference on Digital Transformation in Logistics and Infrastructure (ICDTLI)*, ser. Atlantis Highlights in Computer Sciences, vol. 1, 2019, pp. 275--279.

[13] N. Eden, A. Tsakarestos, I. Kaparias, A. Gal-Tzur, P. Schmitz, S. Hauptmann, and S. Hoadley, ``Using key performance indicators for traffic management and intelligent transport systems as a prediction tool,'' in *19th ITS World Congress*, Vienna, Austria, 2012, pp. 1--8.

[14] E. Krmac and B. Djordjević, ``An evaluation of indicators of railway intelligent transportation systems using the group analytic hierarchy process,'' *Electronics Science Technology and Application*, vol. 4, no. 2, 2018.

[15] Y. Shah, K. Manaugh, M. Badami, and A. El-Geneidy, ``Diagnosing transportation: Developing key performance indicators to assess urban transportation systems,'' *Transportation Research Record: Journal of the Transportation Research Board*, no. 2357, pp. 1--12, 2013.

[16] L. Yang, K. H. van Dam, and L. Zhang, ``Developing goals and indicators for the design of sustainable and integrated transport infrastructure and urban spaces,'' *Sustainability*, vol. 12, no. 22, pp. 1--34, 2020.

[17] C. Iliopoulou and K. Kepaptsoglou, ``Combining its and optimization in public transportation planning: State of the art and future research paths,'' *European Transport Research Review*, vol. 11, no. 1, pp. 1--17, 2019.

[18] E. Dmitrieva, A. Pathani, G. Pushkarna, P. Surekha *et al.*, ``Real-time traffic management in smart cities: Insights from the traffic management simulation and impact analysis,'' in *BIO Web of Conferences*, vol. 86, 2024, p. 01098.

[19] Y. Hamdouch, M. A. Hoque, and W. Y. Szeto, ``A classical optimization approach for the transit network design problem with elastic demand,'' *Transportation Research Part B: Methodological*, vol. 68, pp. 205--225, October 2014.

[20] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, 5th ed. Hoboken, NJ: John Wiley & Sons, 2017.

[21] E. B. Kosmatopoulos, ``An adaptive optimization scheme with satisfactory transient performance,'' *Automatica*, 2007.

[22] A. Kouvelas, E. B. Kosmatopoulos, M. Papageorgiou, and K. Aboudolas, ``Adaptive performance optimization for large-scale traffic control systems," *IFAC Proceedings Volumes*, vol. 42, no. 15, pp. 991--1002, 2009.

[23] EasyExamNotes. (2024) Explain static vs. dynamic optimization. [Online]. Available: https://easyexamnotes.com/explain-static-vs-dynamic-optimization/

[24] G. Trajkovski, ``Real-time adaptation: Enhancing the fluidity and relevance of ai-generated content," *Journal of Science Research and Reviews*, 2024, online: https://josrar.esrgngr.org.

[25] E. J. Dodo, G. Onwodi, and O. Okure, ``An intelligent hybrid learning framework integrating anfis, genetic algorithms, and reinforcement learning for traffic signal control," *Journal of Science Research and Reviews*, vol. 2, no. 6, pp. 21--39, 2025.

[26] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. Hoboken, NJ, USA: John Wiley & Sons, 2009.

[27] T. G. Crainic, ``Service network design in freight transportation," *European Journal of Operational Research*, vol. 122, no. 2, pp. 272--288, 2000.

[28] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, ``Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C*, vol. 43, pp. 3--19, 2014.

[29] I. Saadi, M. Wong, B. Farooq, J. Teller, and M. Cools, ``An investigation into machine learning approaches for forecasting spatio-temporal demand in ride-hailing service," Eindhoven University of Technology, Tech. Rep., 2017.

[30] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, ``Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865--873, 2015.

[31] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[32] D. V. Lint and H. V. Zuylen, ``Travel time prediction for freeways: A comparison of state-of-the-art methods," *Transportation Research Part C*, vol. 13, pp. 347--366, 2005.

[33] V. Mnih *et al.*, ``Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529--533, 2015.

[34] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[35] W. Zhang *et al.*, ``Hybrid metaheuristic and reinforcement learning for adaptive vehicle routing," *Expert Systems with Applications*, vol. 159, p. 113590, 2020.

[36] C. Cai and M. Wei, ``Adaptive urban traffic signal control based on enhanced deep reinforcement learning," *Scientific Reports*, vol. 14, p. 14116, 2024.

[37] X. Zhao, Y. Zhang, and L. Li, ``Connected vehicles' dynamic route planning based on reinforcement learning," *Future Transportation*, 2023.

[38] W. Wu, Y. Zhu, and R. Liu, ``Dynamic scheduling of flexible bus services with hybrid requests and fairness: Heuristics-guided multi-agent reinforcement learning with imitation learning," *Transportation Research Part B: Methodological*, vol. 190, p. 103069, 2024.

[39] R. Ali *et al.*, ``A comprehensive survey of deep learning-based traffic flow prediction models for intelligent transportation systems," *ICCK Transactions on Advanced Computer Systems*, vol. 1, no. 3, pp. 117--137, 2025.

[40] F. Simsir and D. Ekmekci, ``A metaheuristic solution approach to capacitied vehicle routing and network optimization," *Engineering Science and Technology, an International Journal*, vol. 22, no. 3, pp. 727--735, 2019.

[41] N. Perić, S. Begović, and V. Lešić, ``Adaptive memory procedure for solving real-world vehicle routing problem," *arXiv preprint arXiv:2403.04420*, March 2024.

[42] M. Nazari, A. Oroojlooy, L. V. Snyder, and M. Takáč, ``Reinforcement learning for solving the vehicle routing problem," *arXiv preprint arXiv:1802.04240*, 2018, version 2, 21 May 2018.

[43] A. Mozhdehi, M. Mohammadizadeh, S. Kalantari, B. S. S. Kim, and X. Wang, ``Spade: Solving the multi-depot vehicle routing problem with inter-depot routes using multi-agent deep reinforcement learning," in *The 38th Canadian Conference on Artificial Intelligence*, 2025, to appear.

[44] M. J. C. S. Reis, ``An adaptive hybrid metaheuristic for solving the vehicle routing problem with time windows under uncertainty," *Computers, Materials & Continua*, 2025.

[45] A. Konovalenko and L. M. Hvattum, ``Optimizing a dynamic vehicle routing problem with deep reinforcement learning: Analyzing state-space components," *Logistics*, vol. 8, no. 4, p. 96, 2024. [Online]. Available: https://doi.org/10.3390/logistics8040096

[46] S. Kadyrov, A. Azamov, Y. Abdumajitov, and C. Turan, ``Deep reinforcement learning for dynamic vehicle routing with demand and traffic uncertainty," *Operations*

Research Perspectives, vol. 15, p. 100351, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214716025000272

[47] X. Ren, S. Chen, and L. Ren, ``Optimization of regional emergency supplies distribution vehicle route with dynamic real-time demand,'' *Mathematical Biosciences and Engineering*, vol. 20, no. 4, pp. 7487--7518, 2023.

[48] M. E. JAOUHARI, G. BENCHEIKH, and G. BENCHEIKH, ``Metaheuristic and reinforcement learning techniques for solving the vehicle routing problem: A literature review,'' *Laboratory of Informatics and Applications, Moulay Ismail University and CESI-LINEACT Engineering School*, 2024, corresponding author: gbencheikh@cesi.fr.

[49] A. Yaddaden, ``Deep reinforcement learning for the vehicle routing problem,'' PhD Thesis, IMT Mines Alès, Institut Mines-Télécom, École Nationale Supérieure des Mines d'Alès, Montpellier, France, November 2023, supervised by Michel Vasquez, co-supervised by Sébastien Harispe; Jury included Christine Solnon, Jin-Kao Hao, El-Ghazali Talbi, Rodolphe Giroudeau, Audrey Dupont.

[50] K. Danach, L. Saker, and H. Harb, ``Integrating metaheuristics and machine learning for enhanced vehicle routing: A comparative study of hyperheuristic and vae-based approaches,'' *World Electric Vehicle Journal*, vol. 16, no. 5, p. 258, 2025.

[51] L. Kovács and A. Jlidi, ``Neural networks for vehicle routing problem,'' *Advanced Logistic Systems -- Theory and Practice*, vol. 18, no. 2, pp. 17--29, 2024.

[52] GeeksforGeeks, ``What are graph neural networks?'' 2025, last updated: 27 Nov, 2025. [Online]. Available: https://www.geeksforgeeks.org/deep-learning/what-are-graph-neural-networks/

[53] ------, ``Genetic algorithms,'' 2026, last updated: 10 February 2026. [Online]. Available: https://www.geeksforgeeks.org/dsa/genetic-algorithms/

[54] CVRPLIB, ``Capacitated vehicle routing problem library,'' 2026, accessed: 2026-05-19. [Online]. Available: https://galgos.inf.puc-rio.br/cvrplib/index.php/en/instances/2

Appendix A: List of Acronyms

Optimization & Routing

Acronym	Full Form
VRP	Vehicle Routing Problem
VRPTW	Vehicle Routing Problem with Time Windows
DVRP	Dynamic Vehicle Routing Problem
DVRPTW	Dynamic Vehicle Routing Problem with Time Windows
CVRP	Capacitated Vehicle Routing Problem
CVRPTW	Capacitated Vehicle Routing Problem with Time Windows
ACVRP	Asymmetric Capacitated Vehicle Routing Problem
HFVRP	Heterogeneous Fleet Vehicle Routing Problem
MDVRP	Multi-Depot Vehicle Routing Problem
SDVRP	Split Delivery Vehicle Routing Problem
RTVRP	Real-Time Vehicle Routing Problem
VRPSDP	Vehicle Routing Problem with Simultaneous Delivery and Pickup
TSP	Travelling Salesman Problem
PRP	Pollution Routing Problem
BKS	Best Known Solution
KPI	Key Performance Indicator

Algorithms & Methods

Acronym	Full Form
GA	Genetic Algorithm
GNN	Graph Neural Network
ACO	Ant Colony Optimization
PSO	Particle Swarm Optimization
ABC	Artificial Bee Colony
AMP	Adaptive Memory Procedure
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
DQN	Deep Q-Network
MARL	Multi-Agent Reinforcement Learning
ML	Machine Learning
LSTM	Long Short-Term Memory
GLS	Guided Local Search
OX	Order Crossover
ANFIS	Adaptive Neuro-Fuzzy Inference System

System & Framework

Acronym	Full Form
NMFAVRK	Neuro-Metaheuristic Framework for Adaptive Vehicle Routing NetworK
POMDP	Partially Observable Markov Decision Process
k-NN	k-Nearest Neighbors
OR-Tools	Google Operations Research Tools
CVRPLIB	Capacitated VRP Library

Performance, Metrics & Transport

Acronym	Full Form
CO_2	Carbon Dioxide
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Random Access Memory
ITS	Intelligent Transportation Systems
NP	Non-deterministic Polynomial (NP-Hard)