République algérienne démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique

Département Génie Industriel
Grettia Laboratory

End of Study Project Dissertation

for **Obtaining State Engineer's Degree** in **Industrial Engineering**

option: **Industrial Management**

Integrated Optimization of Fleet Sizing, Deployment, and Routing for Electric Service Vehicles

-Case Study: STM-

Conducted by:
Bouchra Zohra BEN MESSABIH

Supervised by:
Dr. Iskander ZOUAGHI (ENP)
Dr. Sana BELMOKHTAR-BERRAF (UGE)
Dr. Walid BEHIRI (UGE)

Defended on **June 25, 2025**, before a jury composed of:

President:	Dr. BELDJOUDI Samia	MCA	ENP
Examiner:	Dr. BOUKABOUS Ali	MAA	ENP
Supervisor:	Dr. ZOUAGHI Iskander	MCA	ENP

ENP 2025

République algérienne démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique

Département Génie Industriel
Grettia Laboratory

End of Study Project Dissertation

for **Obtaining State Engineer's Degree** in **Industrial Engineering**

option: **Industrial Management**

Integrated Optimization of Fleet Sizing, Deployment, and Routing for Electric Service Vehicles

-Case Study: STM-

Conducted by:
Bouchra Zohra BEN MESSABIH

Supervised by:
Dr. Iskander ZOUAGHI (ENP)
Dr. Sana BELMOKHTAR-BERRAF (UGE)
Dr. Walid BEHIRI (UGE)

Defended on **June 25, 2025**, before a jury composed of:

President:	Dr. BELDJOUDI Samia	MCA	ENP
Examiner:	Dr. BOUKABOUS Ali	MAA	ENP
Supervisor:	Dr. ZOUAGHI Iskander	MCA	ENP

ENP 2025

République algérienne démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département Génie Industriel
Laboratoire Grettia

Université Gustave Eiffel
LABORATOIRE GRETTIA
GÉNIE DES RÉSEAUX DE TRANSPORT TERRESTRES ET INFORMATIQUE AVANCÉE

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en in **Génie Industriel**

option: **Management Industriel**

Optimisation intégrée de la dimensionnement, du déploiement et du routage d'une flotte de véhicules de service électriques

-Étude de cas : STM -

Présenté par:
Bouchra Zohra BEN MESSABIH

Sous la Direction de:
Dr. Iskander ZOUAGHI (ENP)
Dr. Sana BELMOKHTAR-BERRAF (UGE)
Dr. Walid BEHIRI (UGE)

Soutenu le **25 juin 2025**, devant un jury composé de :

Président:	Dr. BELDJOUDI Samia	MCA	ENP
Examinateur:	Dr. BOUKABOUS Ali	MAA	ENP
Promoteur:	Dr. ZOUAGHI Iskander	MCA	ENP

ENP 2025

ملخص

تلتزم شركة النقل في مونتريال (STM) بتحويل أسطول مركباتها بالكامل إلى مركبات كهربائية، كجزء من مبادراتها الأوسع للاستدامة. يتناول هذا العمل تحديًا حاسمًا ينشأ من هذا الالتزام: تحسين نشر وتوزيع مركبات الخدمة الكهربائية المخصصة للمشرفين على العمليات لضمان الحد الأدنى من التداخل مع شبكة الحافلات. تم صياغة المشكلة كبرنامج خطي مختلط الأعداد الصحيحة يقلل من عدد المركبات المنتشرة والوقت الإجمالي للاستجابة باستخدام نهج ترتيبي. يتم تقييم فعالية نهجنا من خلال تجارب حسابية باستخدام حل مشكلة الأمثل التجاري CPLEX. يوفر هذا العمل لشركة STM أداة قيمة لدعم اتخاذ القرارات الاستراتيجية والعملية، تهدف إلى تحسين حجم ونشر أسطولها الجديد من المركبات الكهربائية، مع دعم أهدافها في الاستدامة.

الكلمات المفتاحية: التحسين، نشر المركبات، تحديد حجم الأسطول، المركبات الكهربائية، المسارات، تخصيص المركبات.

Résumé

La Société de transport de Montréal (STM), s'est engagée à électrifier l'intégralité de sa flotte de véhicules dans le cadre de ses initiatives de durabilité. Ce travail aborde un défi majeur découlant de cet engagement : l'optimisation du déploiement et de l'affectation des véhicules électriques de service attribués aux superviseurs d'exploitation afin de garantir une perturbation minimale du réseau de bus. Le problème est formulé sous la forme d'un programme linéaire en nombres entiers mixtes qui minimise le nombre de véhicules déployés et le temps de réponse total en utilisant une approche lexicographique. La capacité du modele est évaluée par des expérimentations computationnelles utilisant le solveur d'optimisation commercial CPLEX. Ce travail fournit à la STM un outil précieux d'aide à la décision stratégique et opérationnelle, visant à optimiser la taille et le déploiement de sa nouvelle flotte de véhicules électriques, tout en soutenant ses objectifs de durabilité.

Mots clés : Optimisation, Déploiement des véhicules, Dimensionnement de la flotte, Véhicules électriques, Itinéraires, Affectation des véhicules.

Abstract

The Société de transport de Montréal (STM), the public transport agency of Montreal, has committed to electrifying its entire vehicle fleet as part of its broader sustainability initiatives. This work addresses a critical challenge that arises from this commitment: optimizing the deployment and dispatching of electric service vehicles assigned to operations supervisors to ensure minimal disruption to the bus network. The problem is formulated as a mixed-integer linear program that minimizes the number of deployed vehicles and total response time using a lexicographic approach. The capacity of our model is evaluated through computational experiments using the commercial CPLEX optimization solver. This work provides the STM with a valuable strategic and operational decision-support tool, aimed at optimizing the size and deployment of its new electric vehicle fleet, while supporting its sustainability objectives.

Keywords : Optimization, Vehicle deployment, Fleet sizing, Electric vehicles, Routing, Dispatching.

Dedication

To the one person who matters the most to me, ***my beloved mom****, who constantly pushed me toward excellence, encouraged me, and made sure I had everything I needed to pursue my goals. Thank you for your unconditional love, your sacrifices, your prayers, and your constant presence by my side.*

To ***my two brothers****,*
for always believing in me even when I doubted myself, for cheering me up, lifting me up, and reminding me that I was never alone in this journey. In silence or in words.

To ***my exceptional friends****, Nadhir, Samah, Alyce, Fatma, Farouk, Amine, Younes, Abdelkader and Amro, with whom I shared the best memories. Thank you for always being there for me. I couldn't have done this work without you.*

& ***To my new friends****, Ryan, Fatemeh, Maissa, Moha, and Khadidja, the ones who make you believe that there's still good in people, and that we can help each other without expecting anything in return.*

& To all those who shared with me the joys and challenges of these long years of study.

Bouchra Zohra.

Acknowledgements

I would like to begin by thanking God for writing this beautiful experience into my destiny. I am grateful for the strength, knowledge, and opportunities granted to me, which allowed me to undertake this research study and persevere through challenges to complete it successfully. Without His blessings, this achievement would not have been possible.

I would also like to express my sincere thanks to my supervisors, **Dr. Iskander Zouaghi, Dr. Walid Behiri, Dr. Sana Belmokhtar-Berraf, Dr. Tesseda Boukheroub** and **Dr. Abderrahim Sahli** for believing in me and for their invaluable support. I am deeply grateful for their time, their exceptional guidance, and their unwavering encouragement throughout my journey. Their trust in me, their insightful feedback, and their profound expertise pushed me to sharpen my thinking and elevate my work to a higher level. Their patience, motivation, and immense knowledge were instrumental in helping me navigate the challenges I encountered.

I want to thank my school the one and the only **National Polytechnic school of Algiers (ENPA)**, thank you for these amazing five years. I am grateful for the invaluable experiences and knowledge gained during my time there. I want to thank every person who was part of my journey and helped me become a better version of myself whether through academic guidance, involvement in clubs and events, or simply through their support and encouragement. A special thanks goes to the professors, staff, and peers who contributed to my growth, as well as to all those who made my time at the school truly memorable.

Lastly, I would like to express my sincere gratitude to everyone who contributed, whether directly or indirectly, to the completion of this project. Your support has been deeply appreciated. This accomplishment is as much a result of your collective help as it is of my individual effort.

Bouchra Zohra.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

- **GHG** : Global greenhouse gas
- **STM** : Société de transport de Montréal
- **OS** : Operations Supervisors
- **COC** : Central Operations Center
- **VRP** : Vehicle Routing Problem
- **VSP** :Vehicle Scheduling Problem
- **MDVRP-TW** Multi-Depot Vehicle Routing Problem with Time Windows
- **LP** : Linear Program
- **ILP** : Integer Linear Program
- **MILP** : Mixed-integer Linear Programming
- **B&B** : Branch and Bound
- **DP** : Dynamic Programming
- **CG** : Constraint Programming
- **DP** : Column generation
- **OR** : Operational research
- **EVRP** : Electric Vehicle Routing Problem
- **MOO** : Multi-Objective Optimization
- **MST** : Minimum Spanning Tree
- **TSP** : Traveling Salesman Problem
- **CFLP** : Capacitated Facility Location Problem
- **MA** : Moving Average
- **SES** : Simple Exponential Smoothing
- **ARIMA** : Autoregressive Integrated Moving Average

General Introduction

Global greenhouse gas (GHG) emissions surged by 51% between 1990 and 2021, driving an alarming rate of planetary warming and contributing to the increasingly devastating environmental crises worldwide [15]. This profound impact has established climate change as a preeminent challenge to sustainable economic development. A primary driver of this global crisis is often identified as "fossil" capitalism, reflecting the extensive pollution from greenhouse gas emissions, with traffic, heavily reliant on fossil-fueled vehicles, being a major contributor to this ecological and economic predicament [16].

The transportation sector is central to this issue, exhibiting the highest reliance on fossil fuels of any sector and accounting for nearly a quarter of global GHG emissions in 2021. Within this, road transport encompassing passenger cars, buses, and commercial vehicles is the dominant emitter, responsible for approximately 77% of transportrelated GHG emissions [17]. Consequently, transitioning to sustainable, low-carbon mobility has become a high policy priority globally. In response, vehicle electrification has been identified as a major lever for achieving sustainable and decarbonized mobility [18]. Notably, the electrification of third-party transport fleets, such as those used for public and operational services, offers benefits up to three times greater than electrifying private passenger vehicles, due to their higher utilization rates and mileage [19].

In urban contexts, sustainable mobility has emerged as a critical objective. As cities grow and environmental concerns intensify, public transportation systems are increasingly viewed as pivotal in reducing greenhouse gas emissions, minimizing air pollution, and optimizing energy use [20]. This shift towards cleaner mobility is an environmental imperative and an economic and social necessity for effective city management. Accordingly, public transit agencies worldwide are undertaking a significant transition: the electrification of their vehicle fleets. This ambitious move promises substantial reductions in emissions and local pollution, alongside potential long-term cost savings and operational enhancements [21].

Our work focuses on the Société de transport de Montréal (STM), Montreal's public transport agency, which has committed to electrifying its entire fleet by 2030. This initiative includes not only passenger transport vehicles but also the crucial fleet of Operational Supervisor (OS) service vehicles. These OS vehicles are pivotal for maintaining network fluidity, responding proactively to incidents such as detours, malfunctions, and passenger-related events, thereby ensuring service punctuality and contributing to Montreal's urban quality of life. However, electrifying these critical support vehicles introduces complex strategic and operational questions, particularly concerning optimal fleet sizing and deployment. Determining the ideal number of vehicles and their strategic distribution across

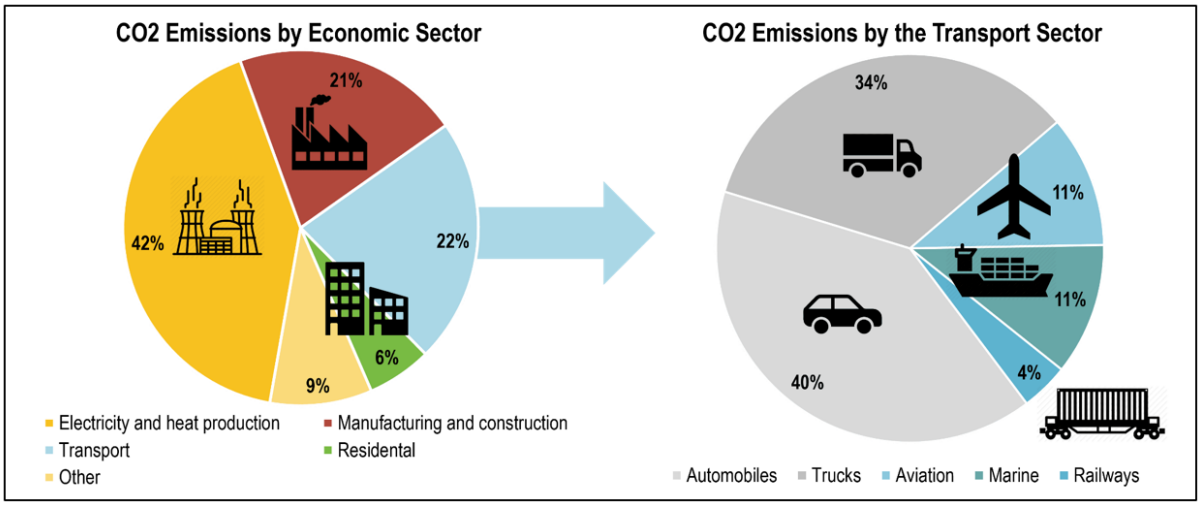

Figure 1: Global greenhouse gas emissions by the transportation sector [1]

depots is vital for minimizing incident response times and ensuring network resilience. This leads to the central research question guiding this work: How can the Société de transport de Montréal optimally size its electrified Operational Supervisor vehicle fleet and strategically deploy these vehicles across its depots to minimize incident response times while considering operational constraints and the specific characteristics of electric vehicles?

This Master's thesis proposes a methodological framework to guide critical decisions regarding the electrification of STM's OS vehicle fleet. It focuses on determining the optimal fleet size and strategic deployment across depots to minimize response times to network incidents. To this end, we develop an optimization-based decision support tool that combines both strategic fleet allocation and operational routing. By addressing these challenges, the project contributes to STM's broader efforts to streamline support operations in an electrified context and improve service reliability for Montreal residents.

This thesis is structured as follows:

Chapter 1 This chapter introduces the research problem by detailing the operational context of the STM. It describes the specific challenges related to the optimization of its electrified OS vehicle fleet, outlining the characteristics and scope of the problem addressed in this thesis.

Chapter 2 will provide a comprehensive review of relevant literature on classical optimization problems pertinent to our study. This includes focusing on facility location, fleet sizing, vehicle assignment/deployment, and Vehicle Routing Problems (VRP) literature, highlighting existing approaches and their limitations in the specific context of electrified operational support vehicles.

chapiter 3 This chapter focuses on the formal development of the solution methodologies. It will present the detailed mathematical formulation of the optimization model designed to capture the complexities of the STM's OS vehicle fleet problem. Additionally, a decomposition-based heuristic approach, developed to address potential scalability challenges, will be introduced and elaborated upon.

Chapter 4 will be dedicated to the proposed solution approaches and discussion of results. It will detail the exact method and the heuristic used to solve the problem and the numerical experiments conducted. The chapter will also include a critical analysis of these results, offering insights into the effectiveness and limitations of each approach.

Finally, the thesis will conclude with a general conclusion summarizing the contributions of this research and outlining potential avenues for future work.

Chapter 1

Industrial Engineering Context and Problem Definition

This chapter establishes the foundational context for the research presented in this thesis. Before delving into the specific operational challenge faced by the Société de transport de Montréal (STM), we will first frame the problem within the broader principles of industrial engineering and operations management. This involves understanding the nature of service logistics, the hierarchical levels of decision-making involved in fleet management, the economic imperatives driving optimization, and the critical link between operational performance and service quality. By establishing this framework, the specific problem of the STM will be positioned as a practical and relevant case study in applied industrial engineering.

1.1 Industrial Engineering Context

1.1.1 Service Logistics

Logistics, in its traditional sense, is primarily concerned with the efficient planning, implementation, and control of the flow and storage of tangible goods from a point of origin to a point of consumption. Service logistics, however, extends this paradigm to a context where the "product" being delivered is an intangible service or an on-site intervention. Instead of managing an inventory of physical goods, service logistics focuses on managing the inventory and deployment of service capacity namely, skilled personnel, specialized vehicles, and necessary equipment to points of demand in a timely and cost-effective manner.

In many urban service systems, such as public transportation, energy utilities, or telecommunications, this demand does not manifest as a predictable customer order but rather as an unplanned, stochastic event. These events, or "incidents," are often spatially and temporally distributed across a wide operational area. The core challenge of service logistics in this environment is therefore to strategically position and dispatch mobile service units to these unpredictable demand points to perform tasks such as repairs, inspections, or operational support.

The effectiveness of a service logistics system is defined by a fundamental trade-off, a classic industrial engineering problem, between service level and operational cost. On one hand, the goal is to maximize the quality and responsiveness of the service, often measured by minimizing the response time the duration between the notification of an incident and the arrival of a service unit. On the other hand, organizations must minimize the total operational costs, which include significant capital investment in the vehicle fleet and ongoing expenses related to personnel, maintenance, and energy.

Addressing this trade-off requires solving a set of interconnected optimization problems that are central to industrial engineering. These include the strategic placement of operational bases (a facility location problem), determining the optimal number of service units required (a fleet sizing problem), and designing efficient daily routes to serve incidents (a vehicle routing problem). The goal is to design a system that is both highly responsive and economically efficient, ensuring operational resilience and high-quality service delivery.

1.1.2 The Levels of Decision-Making in Operations Management

Effective operations management in any large-scale organization relies on a structured and hierarchical approach to decision-making. These decisions are typically categorized into three distinct levels based on their time horizon, scope, and impact on the organization: strategic, tactical, and operational. Understanding this hierarchy is crucial for contextualizing the specific challenges of fleet optimization.

1. **Strategic Decisions** are made at the highest level and focus on the long term,

typically spanning several years. They are characterized by high capital investment, low flexibility, and a significant impact on the organization's overall capabilities and competitive positioning. For a public transit agency, strategic decisions include committing to a major technological transition, such as the electrification of its vehicle fleet, making significant investments in infrastructure like the construction of new depots or charging facilities, and determining the overall long-term size of the fleet through major procurement contracts. These decisions set the stage for all subsequent planning.

2. **Tactical Decisions** bridge the gap between long-term strategy and daily operations, with a medium-term focus (e.g., monthly, quarterly, or annually). They involve the allocation of resources within the framework established by strategic choices. Examples in fleet management include determining the optimal number of vehicles to have active during specific seasons, creating master schedules for personnel, and the general deployment of vehicles to specific home depots to best cover anticipated demand across different regions of the network.

3. **Operational Decisions** are concerned with the day-to-day, short-term management of resources to execute tasks as efficiently as possible. These decisions are highly frequent, have an immediate impact, and are constrained by the tactical and strategic plans already in place. In the context of a service fleet, operational decisions include the real-time dispatching of a specific vehicle to an incoming incident, the detailed routing and sequencing of multiple incidents for a single vehicle during its shift, and managing immediate schedule adjustments.

The problem addressed in this thesis, which involves determining the optimal fleet size and its deployment across various depots, primarily resides at the intersection of the strategic and tactical levels. Fleet sizing is a strategic decision that directly impacts long-term capital investment. The deployment of these vehicles is a tactical choice that dictates resource availability for daily operations. However, the effectiveness of these strategic and tactical decisions is ultimately measured by their impact on operational performance, specifically the ability to provide fast and efficient routing to incidents, thereby minimizing response times. This interplay across all three levels makes the problem a complex and representative challenge in industrial engineering.

1.1.3 Industrial Investment: Economic Justification and Cost Optimization

From an industrial engineering perspective, operational decisions are inextricably linked to their economic implications. The management of a service fleet, particularly one undergoing a significant technological transformation like electrification, is not merely a logistical exercise but a major strategic investment that requires rigorous economic justification. The decision to electrify a fleet introduces a fundamental trade-off between long-term operational savings and substantial upfront capital expenditure.

The primary economic challenge lies in balancing two categories of costs:

- **Capital Expenditures (CAPEX):** These are the significant, one-time investments required to acquire the assets. In the context of fleet electrification, CAPEX includes not only the purchase price of the electric vehicles themselves which are often more expensive than their internal combustion engine counterparts but also the substantial costs associated with building the necessary charging infrastructure at depots and potentially along routes.
- **Operating Expenditures (OPEX):** These are the ongoing, recurring costs associated with the daily operation of the fleet. Fleet electrification promises considerable long-term reductions in OPEX, primarily through lower energy (electricity vs. fuel) costs and reduced maintenance needs, as EVs have fewer moving parts than traditional vehicles.

The role of the industrial engineer is to provide a quantitative basis for validating that the long-term OPEX savings justify the initial CAPEX. This is where optimization becomes an indispensable tool for strategic financial planning. A key risk in such a large-scale investment is the misallocation of capital. Simply replacing the existing fleet on a one-for-one basis fails to account for the different operational characteristics of EVs (e.g., range limitations, charging times) and misses a critical opportunity for process re-engineering.

Optimization, therefore, serves as a crucial tool for "right-sizing" the investment. The model developed in this thesis aims to answer the fundamental question: "What is the minimum number of electric vehicles required to maintain or improve the current level of service?" By determining this optimal fleet size, the model directly informs the investment decision, preventing two costly errors:

1. **Over-investment:** Purchasing too many vehicles leads to underutilized assets and unnecessary capital expenditure, tying up funds that could be used elsewhere.
2. **Under-investment:** Purchasing too few vehicles results in a degradation of service quality (e.g., longer response times), which can lead to larger operational disruptions, decreased customer satisfaction, and ultimately, higher indirect costs.

Ultimately, this work provides a quantitative methodology to support management in making an informed, data-driven investment decision. It helps justify the significant expenditure on electrification by demonstrating how an optimally sized and deployed fleet can meet service objectives efficiently, thereby ensuring that the financial and operational benefits of this strategic shift are fully realized.

1.1.4 Service Quality and Operational Performance

In public transportation, the ultimate goal of any operational system is to deliver a high-quality service to its users. From an industrial engineering standpoint, service quality is not an abstract concept but a set of measurable performance characteristics that directly impact customer satisfaction and system reliability. For a public transit network,

key dimensions of service quality include punctuality (adherence to schedules), regularity (consistent intervals between vehicles), and overall reliability (the predictability and dependability of the service).

Operational incidents such as vehicle breakdowns, traffic accidents causing detours, signal malfunctions, or passenger-related issues are primary sources of non-quality. These unplanned events introduce variability and disruptions into the system, leading to delays, service gaps, and a degradation of the passenger experience. Consequently, the ability to manage and mitigate the impact of these incidents is a critical component of maintaining a high level of service quality.

This is precisely where the role of the Operational Supervisor (OS) vehicle fleet becomes paramount. These vehicles function as an essential corrective action system, dispatched to resolve disruptions and restore normal operations as quickly as possible. The efficiency of this response system is, therefore, directly linked to the overall quality of service delivered by the transit agency. The effectiveness of the OS fleet can be measured by a set of Key Performance Indicators (KPIs) that are fundamental to industrial engineering and service management:

- **Average Response Time:** The time elapsed from the moment an incident is reported to the arrival of an OS vehicle on the scene. This is a primary driver of service restoration speed; a lower response time directly translates to a shorter disruption for passengers.
- **Incident Resolution Time:** The total time required to manage and clear an incident. While partly dependent on the nature of the incident itself, it is heavily influenced by the timely arrival of the OS unit.
- **Network Availability/Uptime:** The percentage of time the transportation network operates according to its planned schedule. By efficiently resolving incidents, the OS fleet helps maximize this crucial system-level KPI.

Therefore, the core problem of optimizing the OS vehicle fleet is not merely a cost-reduction exercise. It is fundamentally an effort to enhance the operational performance of the incident response system. Minimizing the response time, which is a central objective of the model developed in this thesis, is a direct lever for improving the reliability and punctuality of the entire public transport network, thereby contributing to a higher standard of service quality for all passengers.

1.1.5 Incident Management and Continuous Improvement

A core tenet of modern industrial engineering is viewing complex operations through the lens of process management and continuous improvement, concepts popularized by methodologies such as Lean and Six Sigma. From this perspective, a public transportation network can be modeled as a large-scale production process where the "product" being delivered is a reliable and timely mobility service. The success of this process depends on its stability, predictability, and ability to consistently meet defined quality standards.

Within this framework, operational incidents (e.g., vehicle breakdowns, unexpected detours) are not just isolated events but are analogous to defects or deviations in a manufacturing process. They represent departures from the standard, planned operation and introduce variability that degrades the final product's quality. The role of the industrial engineer, therefore, involves designing systems to manage this variability through two complementary strategies:

1. **Prevention:** Implementing proactive measures to reduce the frequency of incidents, such as robust preventative maintenance schedules for vehicles and infrastructure.
2. **Reaction:** Establishing an efficient and effective system to respond to incidents when they inevitably occur, in order to minimize their impact on the overall process.

The work presented in this thesis focuses squarely on the second strategy: optimizing the reactive system. The Operational Supervisor (OS) vehicle fleet constitutes the primary mechanism for "process correction." When a deviation occurs, an OS vehicle is dispatched to the "point of defect" to diagnose the problem, implement a solution, and restore the transportation process to its stable state as quickly as possible.

Therefore, the problem of sizing and deploying the OS fleet is fundamentally a problem of designing an optimal incident response process. An insufficient or poorly deployed fleet leads to a slow and inefficient correction process, allowing the negative effects of a single incident (e.g., a bus delay) to propagate and amplify throughout the network. Conversely, an optimally designed fleet with the right number of vehicles strategically positioned across the territory ensures that the "correction" process is executed efficiently and at the lowest possible cost. By providing a quantitative method to right-size and deploy this critical response asset, this thesis contributes directly to the continuous improvement of the STM's core service delivery process, enhancing its resilience and ability to manage operational disruptions.

1.2 Problem Description: STM case study

1.2.1 Introduction to the STM Context

Public transportation in Montreal boasts a rich history spanning over 150 years, originating in 1861 with the city's first horse-drawn trams. The operating entity, now known as the Société de transport de Montréal (STM), has since evolved into a major public corporation, progressively introducing buses (since 1919), the metro system (since 1966), and paratransit services (since 1980). Today, supported by its nearly 10,600 employees [22], the STM's official mission is to "develop and offer an essential public service to the Montreal community by delivering a safe, accessible, human, and high-performing mobility experience, while playing a key role in the fight against climate change" [22].

Aligning with this mission, and under the guidance of its Organizational Strategic Plan 2030 (PSO 2030), the STM has cemented its vision to become a leader in sustainable mobility. A cornerstone of this vision is the ambitious plan to electrify its entire fleet

Table 1.1: Overview of STM's Key Statistics for 2024

Financial Data	
Annual Budget	1.8 B$
10-Year Investment	21.1 B$
Credit Ratings	**Long-term Debt:** AA (Standard and Poor's), Aa2 (Moody's) **Short-term Borrowing:** A-1+ (Standard and Poor's), P-1 (Moody's)
Client Profile	
Unique Clients	500,000 per average working day
Female Clients	54%
Student Clients	40%
Place of Residence	82% Montreal, 9% North Shore, 9% South Shore
Workforce	
Number of Employees	10,603
Diversity	42% ethnic minorities, visible minorities, or Indigenous 22.7% women
Company Rank in Quebec	8th largest employer

by 2030. This represents a monumental undertaking, given the scale of its operations. For instance, as of its 2024 report, the bus network alone comprised 1,849 vehicles, of which only 41 were fully electric [22]. This thesis focuses on another critical component of this transition: the service vehicles utilized by Operations Supervisors (OS). The electrification of this specific fleet is already well underway, with the STM reporting a 37.8% electrification rate for its service vehicles in 2024 [22].

Figure 1.1: Example of a Bus from the STM Public Transport Fleet [2]

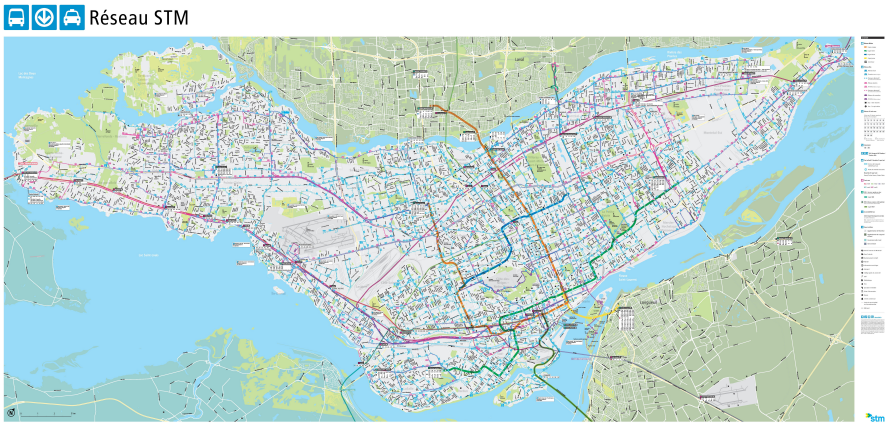

Figure 1.2: Overview of STM Bus Network [2]

The OS vehicles play a pivotal role in maintaining the fluidity and reliability of the bus network—the "operational performance" that the STM has oriented itself towards [22]. They enable supervisors to intervene quickly across the Island of Montreal to resolve a variety of issues that could disrupt service, such as unexpected detours, signage problems, light equipment failures, and passenger-related conflicts. The core mission of the OS team is to ensure a constant presence across the network, thereby minimizing the negative impact of these unforeseen events on passengers and contributing to the overall quality and reliability of Montreal's public transit system.

1.2.2 Problem Statement and Objectives

As the STM undertakes the ambitious and significant investment of electrifying its entire vehicle fleet, including both buses and service vehicles, strategic planning is paramount to ensure both financial prudence and operational excellence. This thesis specifically focuses on the Operational Supervisor vehicle fleet, aiming to assist the STM in this transformative process. A key aspect of this assistance is to minimize the capital investment associated with electrification by determining the optimal fleet size for these specialized OS vehicles that is, the minimum number of electric vehicles required to effectively satisfy all anticipated operational incidents while maintaining high service standards.

Figure 1.3: Example of Electric OS Vehicles [3]

When an incident occurs on the bus network, the STM's Central Operations Center (COC) is responsible for maintaining service continuity by dispatching the nearest available OS vehicle to the scene. While this approach ensures a rapid response and is straightforward to implement, it raises challenges in the context of an electrified fleet, particularly due to the limited range and charging constraints of electric vehicles.

Figure 1.4 illustrates the key interactions within the STM system. An incident triggers an intervention request, which is communicated to the COC. The COC then dispatches an available OS vehicle, initiating both an information flow (from the bus to the COC, then from the COC to the vehicle) and a physical flow (the vehicle traveling from its current location or depot to the incident site, and subsequently between incidents or back to a depot).

The scope of this study focuses on the Island of Montreal, which is geographically divided into 15 distinct operational sectors as illustrate figure 1.5. The OS vehicles operate from eight strategically located depots, which serve dual purposes as bus depots and maintenance operations centers, including electric charging infrastructure. OS personnel are typically assigned to one of these depot and work in rotating shifts.

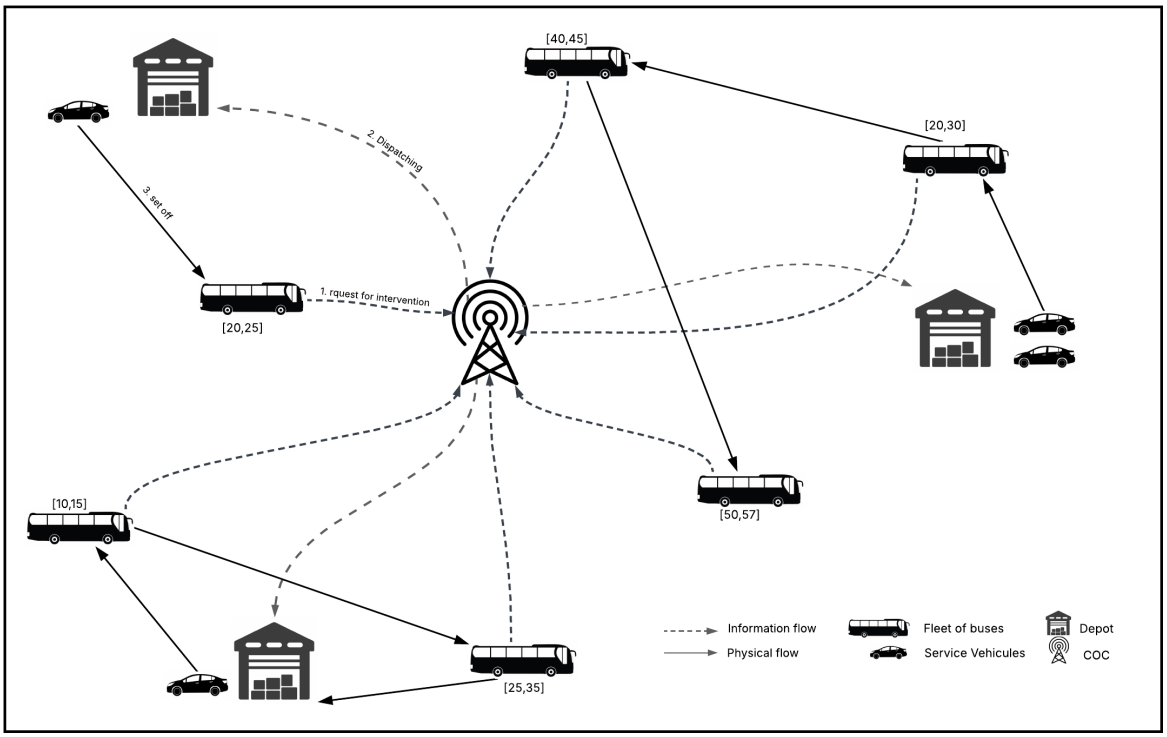

Figure 1.4: OS vehicle operations and system interactions

The primary objective is to determine the optimal size of the electrified OS service vehicle fleet and its strategic deployment across depots. This will support the STM in its vehicle electrification process by identifying the minimal number of vehicles required, considering the specific constraints of electric vehicle operation, while ensuring the continued efficiency and responsiveness of its service vehicle fleet.

It is important to clarify that the approach developed in this thesis is not intended as a real-time operational dispatch system, given the inherently unpredictable nature of incidents in both time and location. Rather, its primary objective is to serve as a strategic planning tool to optimize the sizing and spatial allocation of the OS vehicle fleet across depots. This aims to achieve cost-efficient investment decisions without compromising service quality or responsiveness. Since the core mission of these vehicles is to address operational incidents, this operational dimension is incorporated into the modeling framework by leveraging incident data, potentially generated by a forecasting model trained on historical observations, which will be detailed in the next section. The developed tool also supports post-hoc analysis, enabling the evaluation of ideal fleet sizes and deployment strategies for effectively managing past incidents, thereby offering valuable insights for future planning. This work constitutes an initial step toward the development of a more dynamic, operational system capable of integrating real-time incident data.

Figure 1.5: STM Depots Across Montreal's Divided Sectors

1.2.3 Analysis of the Current STM Operational Supervisor System

To gain a comprehensive understanding of the existing operational dynamics of the STM's OS vehicle fleet, an analysis of current practices and incident data was undertaken. This section details observations derived from a representative dataset, focusing on incident distribution, depot workload, shift patterns, and current dispatching strategies.

For this analysis, operational data from a typical day, encompassing approximately 200 incidents requiring OS intervention, was meticulously examined. To effectively explore and present the key spatial and operational characteristics inherent in this data, visualizations were developed using Microsoft Power BI. Power BI is a business analytics service by Microsoft that provides interactive visualizations and business intelligence capabilities with an easy-to-use interface, enabling users to create their own reports and dashboards. It allows for the connection to, and transformation of, various data sources into coherent, visually immersive, and interactive insights.

An example of such a visualization is presented in Figure 1.6. This particular dashboard, created using Power BI, depicts the geographical distribution of the STM's depots along-

side the precise locations of the recorded interventions (incidents) across the Island of Montreal. Such visual representations are invaluable for gaining an intuitive understanding of demand patterns, depot coverage, and potential areas of operational focus.
Key Observations from Data Analysis:

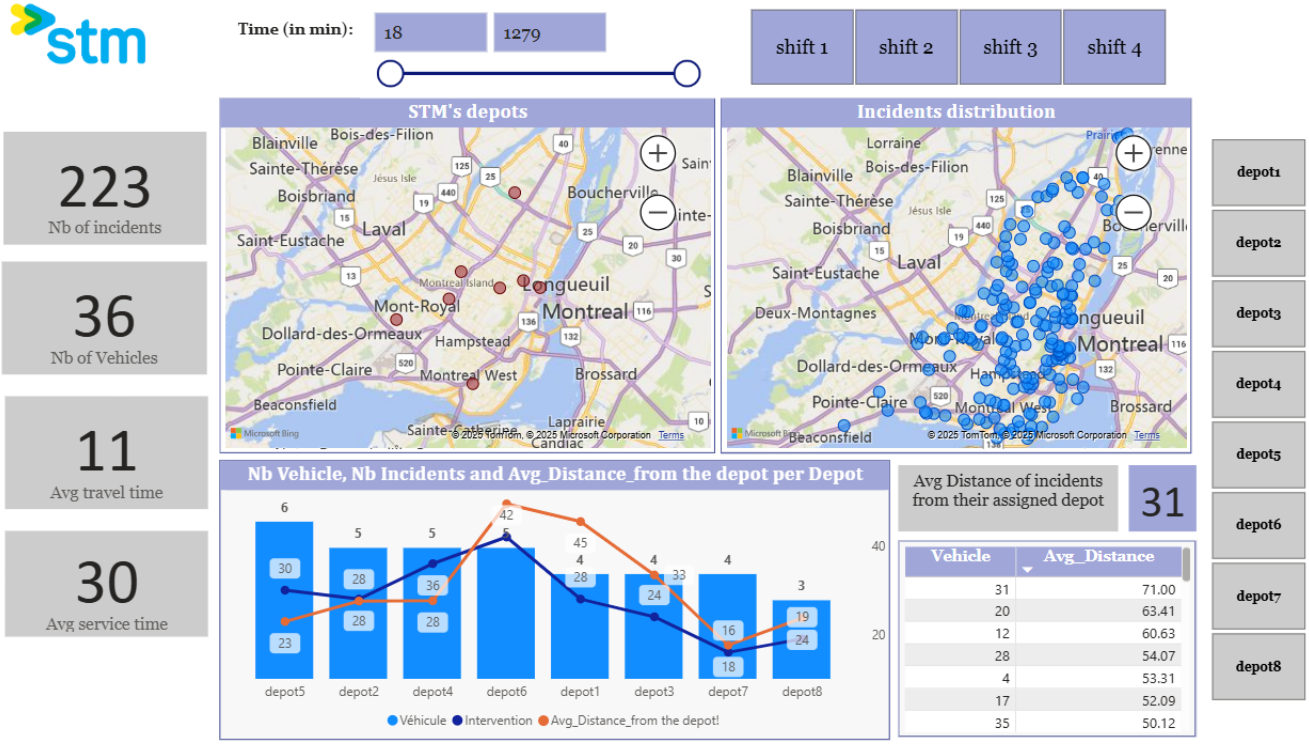

Figure 1.6: Analysis of OS Intervention Demand and Depot Workload on a Typical Day

1. **Centralized Demand:** A prominent observation from Figure 1.6 is the concentration of demand for OS interventions in the central areas of the island. This spatial clustering suggests that depots located centrally, or those with service areas covering these high-demand zones, inherently handle a larger proportion of incidents.

2. **Depot Workload and Vehicle Allocation:** The visualization also provides insights into the number of OS vehicles assigned to each depot (average of 5 vehicles) and, consequently, the number of interventions handled by vehicles originating from these respective depots which change depending on the time period and the sector. Notably, Depot 5, situated in a geographically central position, appears to manage a significant portion of the daily incidents.

3. **Shift-Based Demand Variation:** The OS operations are structured around distinct shifts. Our analysis indicates a clear variation in incident frequency across these shifts. Specifically, Shift 2 and Shift 3, which correspond to peak daytime and evening periods, exhibit a considerably higher volume of interventions compared to other shifts (e.g., early morning or late night). This temporal variation in demand has direct implications for vehicle availability and deployment strategies throughout the day.

4. **Current Dispatching Strategy and Routing Inefficiencies:** The current dispatching protocol employed by the STM appears to prioritize assigning the closest

available OS vehicle to an incident, irrespective of the vehicle's designated depot or optimal routing considerations. This "nearest available" strategy, while simple to implement, can lead to situations where vehicles travel extensive distances from their home depot to attend to an incident (in some observed cases, up to 70 km). This practice indicates a lack of consideration for predefined service zones or depot assignments in the dispatch decision, and suggests that the current system does not actively seek to optimize vehicle utilization or minimize overall travel distances for the fleet.

These observations highlight that while the current system aims for rapid response by dispatching the nearest unit, it may inadvertently lead to inefficiencies in terms of travel distances, imbalanced workload distribution among depots (beyond what natural demand dictates), and suboptimal utilization of the OS vehicle fleet. This analysis underscores the opportunity and the need for a more strategic approach to fleet sizing, deployment, and potentially dispatching, particularly as the STM transitions to an electrified fleet with its inherent range and charging considerations. The objective of our work, therefore, is to develop a framework that addresses these inefficiencies by optimizing the deployment and sizing of the OS vehicle fleet to better match demand patterns and improve overall operational performance.

The STM evaluates the performance of its OS interventions through a set of key performance indicators, which this study seeks to influence positively. These metrics offer a quantitative foundation for analyzing operational efficiency, service quality, and resource allocation. They are presented below in descending order of strategic importance to the company, from the most critical to the least:

1. **Overall Fleet Size:** Refers to the total number of vehicles available. A smaller fleet reduces capital and operational costs but must be large enough to ensure quick and effective responses to incidents.

2. **Average Fleet Utilization:** Measures how actively vehicles are used (e.g., number of incidents handled or time spent on service). High utilization indicates efficiency, but excessive levels may overwhelm the system.

3. **Average Response Time:** The average time between an incident report and vehicle arrival. It reflects service quality and is impacted by fleet size and how well vehicles are deployed.

4. **Average Traveling Time:** The average time vehicles spend traveling to incidents. Lower travel time reduces costs (e.g., energy, wear and tear) and improves availability for future tasks.

5. **Average Waiting Time:** The time incidents must wait for an available vehicle when all are occupied. Long waiting times may signal inadequate fleet size or inefficient deployment strategies.

6. **Total Daily Distance Traveled:** The cumulative distance covered by all vehicles in a day. Reducing this metric helps cut costs, conserve energy, and minimize environmental impact.

In our work, we aim primarily to optimize the most critical indicator the overall fleet size as it has the greatest impact on cost and strategic planning. At the same time, we strive to take into account the other indicators to ensure a balanced and effective operational performance.

1.2.4 Forecasting Approach for Incident Occurrence

A fundamental input for effectively planning and optimizing the OS vehicle fleet is the forecasted incident demand. Given the inherently unpredictable nature of when and where incidents will occur, and since our objective is to determine the optimal fleet size, we adopt a deterministic approach for the optimization model. To support this, we utilize an existing incident forecasting model previously developed for the STM. This model was built using four years of historical incident data provided by the STM, encompassing key information such as the time and date of each incident, its location, the duration of the intervention (service time), the specific OS vehicle and depot involved, and other relevant operational details.

The primary output of this forecasting model, relevant to our study, is the prediction of future incident occurrences specifically their expected timing and spatial distribution. Although our optimization model does not account for the dynamic or stochastic nature of real-time demand, these forecasts serve as a critical input to inform strategic decisions regarding fleet sizing and deployment.

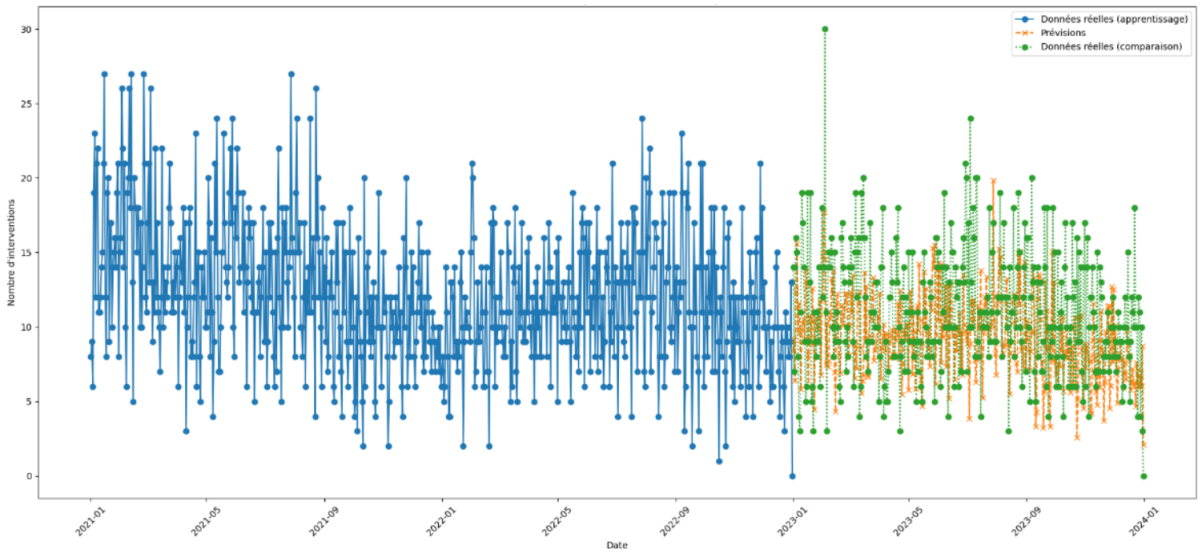

Figure 1.7: Demand Forecasting Model

1.3 Conclusion

This chapter has laid the essential groundwork for this thesis by providing a comprehensive description of the operational challenge faced by the STM concerning its OS vehicle

fleet, particularly in light of its ambitious electrification goals. The core problem addressed in this study has been clearly defined: to determine the optimal fleet size for the electrified OS vehicles and their most effective deployment across the STM's eight depots on the Island of Montreal. This strategic decision aims to minimize operational costs and ensure efficient incident response, directly supporting the STM's objectives of maintaining service reliability while transitioning to a more sustainable fleet. Key performance indicators, including overall fleet size, average fleet utilization, response times, and total distance traveled, have been identified as crucial metrics for evaluating the success of any proposed solution. Furthermore, a set of operational assumptions has been outlined to define the scope and boundaries of the problem, enabling a focused and tractable analysis. An analysis of the current OS operational system, supported by data from a typical day of operations, revealed several key insights. These include the current "nearest available" dispatching strategy. This strategy, while simple, often leads to extensive travel distances for individual vehicles. Crucially, since the spatial and temporal distribution of incidents directly affects the number of vehicles needed and their operational efficiency, understanding how vehicles will serve sequences of interventions essentially, their routing based on forecasted demand becomes an important consideration for effective planning and better vehicle utilisation performance. The insights gained from this chapter underscore a clear opportunity for developing a strategic-operational tool to support the decision making and assist STM in this transition. Specifically, this research will focus on addressing three interconnected decisions:

1. **Optimal Fleet Sizing:** Determining the minimum number of electrified OS vehicles required to meet forecasted service demands.
2. **Optimal Fleet Deployment:** Strategically allocating these vehicles across the available depots to minimize response times, enhance overall network coverage, and balance workload.
3. **Efficient Vehicle Routing:** While not developing a real-time dynamic routing system, the planning framework must consider how deployed vehicles would efficiently cover sequences of forecasted incidents to accurately assess fleet requirements and deployment effectiveness.

Chapter 2

Literature Review

This chapter provides a comprehensive understanding of key concepts related to our study, beginning with a review of similar problems found in the literature, particularly in the context of emergency vehicle management, as they offer valuable insight to treat our problem. We will then explore foundational problems that are central to our research, such as facility location, fleet sizing, vehicle allocation, deployment, dispatching, and routing. These problems form the core of the operational challenges faced in fleet management and are essential to our study of the STM's electrified service vehicle optimization. Furthermore, we will examine the methods most commonly used to solve these types of problems, including classical optimization techniques and more recent advancements. This review will highlight existing approaches and their limitations, particularly in relation to the specific context of electrified operational support vehicles, to frame the contribution of our research.

2.1 The Challenge of Combinatorial Optimization and NP-Hardness

Many real-world optimization problems, especially in logistics, scheduling, and network design, involve discrete decision variables (e.g., whether to use a vehicle, which route to select, whether to open a facility). These fall under the umbrella of combinatorial optimization, a subfield of mathematical optimization concerned with finding an optimal object from a finite, or countably infinite, set of objects, where the set of feasible solutions is discrete. Classic examples include the Traveling Salesman Problem (TSP), the minimum spanning tree problem, and the knapsack problem. The primary challenge in combinatorial optimization lies in its computational complexity. As the size of a problem instance increases (e.g., the number of cities or incidents), the number of possible solutions can grow exponentially or even factorially. This "combinatorial explosion" makes an exhaustive search for the optimal solution computationally infeasible for all but the smallest instances.

A critical aspect of studying these problems is understanding their formal computational complexity, which relates to the resources, particularly time and memory, required by an algorithm to solve a problem instance. While some problems are considered "tractable" and can be solved by algorithms whose runtime grows polynomially with the input size (these belong to the complexity class P), many important combinatorial optimization problems belong to a class known as NP-hard problems [23, 24]. For NP-hard problems, no known algorithm can find an optimal solution in polynomial time for all instances (unless P=NP, a major open question in computer science). When linear programs are restricted by requiring some or all variables to take on integer values, they become Integer Linear Programs (ILPs) or Mixed-Integer Linear Programs (MILPs), which are, in their general form, NP-hard.

The NP-hard nature of problems like vehicle routing and facility location, which are central to this thesis, has profound implications for developing solution methods. It means that there is often little hope of finding a complete polyhedral characterization that would allow them to be solved as easily and efficiently as standard Linear Programs (LPs) [23]. This inherent difficulty necessitates the development of specialized approaches. Consequently, the field has developed two main streams of solution methodologies. The first involves exact algorithms, such as branch-and-bound and cutting plane methods, which guarantee finding the optimal solution but may still have exponential worst-case runtimes. The second stream involves approximate methods, such as heuristics and metaheuristics, which forgo the guarantee of optimality in exchange for finding high-quality solutions within a reasonable computational timeframe. Linear Programming relaxations, where integer constraints are temporarily ignored, have been a foundational technique since the 1960s, often serving as a basis for both exact and approximate algorithms [23].

This inherent complexity underscores the fine line often observed between "very easy" (polynomially solvable) problems and "very hard" (NP-hard) problems, even when they appear structurally similar [24]. Understanding this landscape of computational complexity is therefore crucial for selecting, designing, and evaluating the solution methodologies for the optimization challenges addressed in this thesis.

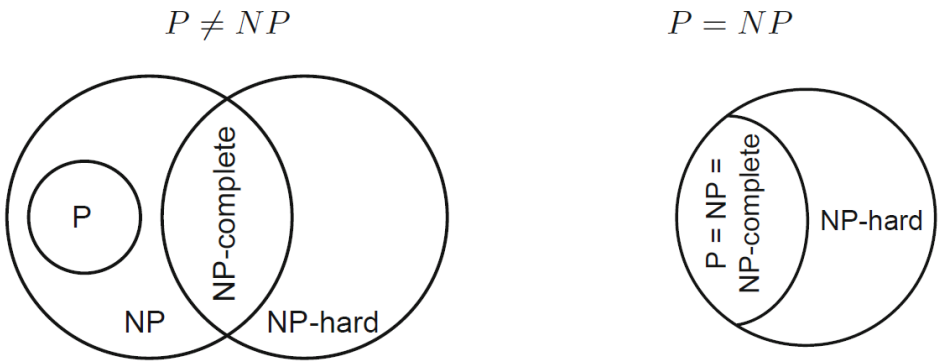

Figure 2.1: NP Problems [4]

2.2 Introduction to Fleet Management

Fleet management refers to the coordinated administration and optimization of a group of vehicles used for transporting goods or passengers. It encompasses a wide range of activities such as vehicle acquisition, maintenance, fuel management, routing, driver supervision, and compliance with regulations. The primary objective of fleet management is to ensure that transport operations are carried out efficiently, cost-effectively, and reliably while meeting service quality and safety standards.

In the context of both private logistics companies and public transportation agencies, fleet management plays a vital role at tactical and operational levels. It is especially critical for aligning available resources with fluctuating customer demand, ensuring punctuality, and minimizing environmental and financial costs.

To address these complex objectives, various mathematical models and computational methods have been developed to support the planning and execution of fleet operations. These methods often aim to optimize vehicle routing, scheduling, and allocation problems that typically fall into the category of combinatorial optimization. Such problems, including the well-known VRP and Vehicle Scheduling Problem (VSP), are notoriously difficult to solve due to their large solution spaces and operational constraints. Moreover, dynamic fleet management introduces additional challenges, as it requires real-time decision-making to respond to unexpected events such as traffic congestion, equipment failures, or emergency incidents. In this setting, systems must adapt rapidly while preserving service reliability and efficiency [25].

As transport systems evolve particularly through digitalization, automation, and electrification, fleet management is becoming increasingly sophisticated. These technological advancements offer new opportunities for improving sustainability, reducing emissions, and enhancing operational resilience. Consequently, modern fleet management is not only a logistical necessity but also a strategic lever for achieving broader environmental and societal objectives.

2.3 Fleet Management of Emergency Service Vehicles

Effective fleet management is a critical challenge within emergency service systems, which operate under demanding, time-sensitive, and often uncertain conditions. These systems encompass a variety of crucial services, including ambulance dispatch, fire department operations, traffic incident response, and disaster relief efforts. Efficiently managing these diverse fleets involves grappling with complex decisions across multiple domains. Key problem areas in this field address facility location, vehicle deployment to bases, assignment of service providers to demands, real-time dispatching of units, and proactive relocation of available resources [26, 27].

Figure 2.2: Examples of emergency vehicles [5]

Successfully tackling these issues is paramount for achieving primary system objectives, which consistently include minimizing response times, maximizing coverage of the service area, and optimizing the utilization of valuable resources [28, 29, 30]. Research often considers these problems within a multi-level planning hierarchy, distinguishing between strategic (long-term), tactical (medium-term), and operational (short-term) decision horizons [31, 32, 33]. The specific operational context can vary significantly depending on the type of service; for instance, the operational environment and typical stationing points for freeway emergency vehicles differ from those for ambulances or fire trucks based at static stations [34]. Furthermore, decision-making in this domain often relates to or builds upon established frameworks from areas such as vehicle routing problems and coverage problems [35], sometimes specifically addressing the routing of vehicles that may handle sequential tasks [36]. Given the inherent complexity and dynamic nature of emergencies, sophisticated modeling approaches, including mathematical programming, queueing theory, and various simulation techniques (such as discrete event simulation), are widely employed to analyze system performance and derive optimal strategies [37, 38, 39, 26]. To provide a structured understanding of this domain, the fundamental problems treated in this context are detailed in the subsequent subsections.

2.3.1 Facility Location Problem

One of the fundamental decisions in the design and operation of emergency service systems, directly impacting efficiency and response times, is the strategic placement of facilities or bases from which vehicles are dispatched. The formal study of facility location has a rich history, often traced back to the early 20th century with Alfred Weber's work in

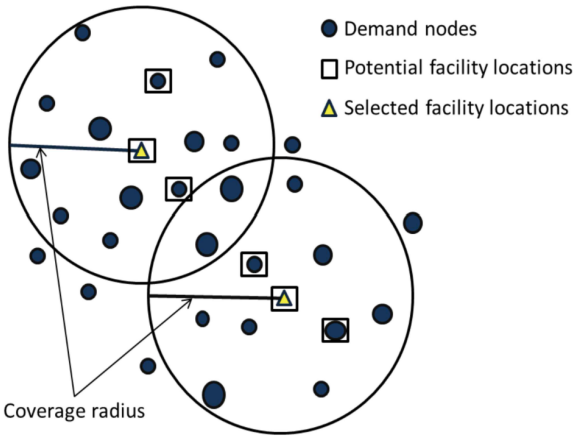

Figure 2.3: Maximal covering location problem [6]

1909, which addressed the problem of determining the optimal site for a single warehouse to minimize transportation costs to customers [40]. This foundational concept spurred interest, which was significantly revitalized in the 1960s by researchers like Hakimi (1964). Hakimi extended location theory to network-based problems, motivated by practical applications such as positioning switching centers in communication networks or police stations along highways. His work explored the general problem of locating one or more facilities on a network to minimize either the total distance to the closest facility or the maximum such distance to any point on the network.

Since these pioneering efforts, the field of location theory has flourished, becoming a well-established area of operations research with over a century of development [41]. While initial formulations were often static and deterministic, focusing on single objectives like minimizing distance or cost, the field has evolved to address more complex scenarios involving uncertainty, dynamics, and multiple conflicting objectives, particularly relevant for public service and emergency contexts [40]. Within this domain, several core problem formulations have been extensively studied and applied, serving as the basis for determining optimal facility sites in various service systems. These fundamental models include:

- **Covering models:** These models are focused on ensuring service accessibility within a predefined standard, typically a maximum allowable response time. Two prominent variants are the Set Covering Problem, which identifies the minimum number of facilities needed to ensure that all demand points are within the service time standard, and the Maximal Covering Problem, which aims to locate a fixed number of facilities (P) to maximize the total demand covered within that standard. These models are essential for establishing and maintaining minimum service guarantees and equitable coverage.

- **The P-median problem:** This classic problem seeks to locate a fixed number of facilities (P) such that the sum of the distances (or weighted distances, representing travel time or cost) between each demand point (e.g., potential incident location) and its nearest assigned facility is minimized. In the context of emergency services, the P-median objective often translates to minimizing the total system-wide travel time to incidents, aiming for overall operational efficiency.

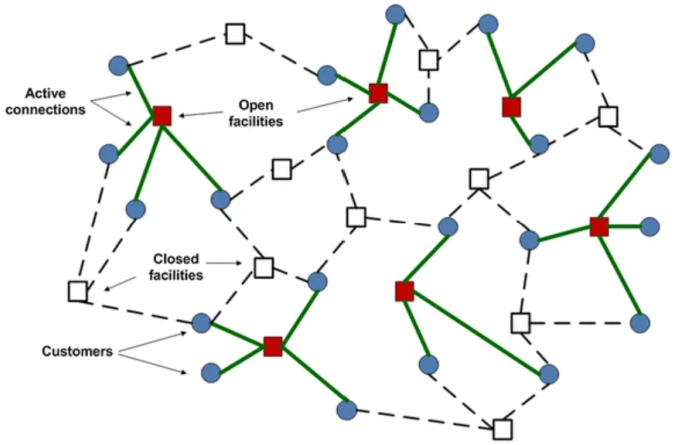

Figure 2.4: Illustrative example of the facility location problem [7]

- **The P-center problem:** This problem addresses the worst-case response by selecting the location of P facilities to minimize the maximum distance (or travel time) from any demand point to its closest facility. The P-center objective is particularly important in emergency services for minimizing the longest possible response time across the entire service region, thereby focusing on equity and guaranteeing a minimum level of accessibility for all areas.

The P-median, P-center, and various covering problems (such as Set Covering and Maximal Covering) represent foundational models in facility location theory, each addressing distinct strategic objectives. These classical formulations often serve as a starting point for understanding location decisions. However, real-world applications frequently necessitate extensions and variations of these basic models to incorporate more specific operational requirements and constraints. For instance, if the assumption in the standard P-median problem that facilities have unlimited capacity is not realistic, and facilities (e.g., depots) are indeed capacity-constrained, the problem evolves into a capacitated variant. One of the most well-known and widely studied extensions addressing this is the Capacitated Facility Location Problem (CFLP), which explicitly considers both the fixed costs of opening facilities and their finite service capacities.

The Capacitated Facility Location Problem can be introduced with the following notations:let I be the set of customers and J the set of potential facility locations. Each customer $i \in I$ has a demand h_i, and each facility $j \in J$ has a capacity S_j and a fixed opening cost f_j. The cost of serving customer i from facility j is denoted by c_{ij}. Additionally, at most p facilities can be opened.

The decision variables are: $Y_j = 1$ if a facility is opened at site j, and $X_{ij} = 1$ if customer i is assigned to facility j.

$$\text{Minimize } Z = \sum_{j \in J} f_j Y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} X_{ij} \tag{2.1}$$

$$\text{subject to: } \sum_{j \in J} Y_j \leq p \tag{2.2}$$

$$\sum_{j \in J} X_{ij} = 1 \quad \forall i \in I \tag{2.3}$$

$$\sum_{i \in I} h_i X_{ij} \leq S_j Y_j \quad \forall j \in J \tag{2.4}$$

$$Y_j \in \{0, 1\} \quad \forall j \in J \tag{2.5}$$

$$X_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J \tag{2.6}$$

This model aims to minimize the total cost, which includes the fixed costs of opening facilities and the variable costs of assigning customers. The constraints ensure that the number of opened facilities does not exceed p, each customer is assigned to exactly one facility, and the total demand assigned to any open facility does not exceed its capacity.

2.3.2 Fleet Deployment Problem

The fleet deployment problem is a critical operational and tactical challenge addressed extensively across various domains in academic literature. It fundamentally concerns the optimal allocation and assignment of a finite set of resources (the fleet) to meet a given set of demands or tasks, typically under specific constraints and objectives. The perspectives on this problem vary depending on the sector, the specific decisions being made, and the primary goals of the organization.

In the maritime industry, particularly liner shipping, fleet deployment is a well-studied area. For instance, [42] describe the problem as determining an optimal strategy for assigning a shipping company's vessels to a defined set of voyages for an upcoming planning horizon. This involves not only assigning vessels to voyages to ensure all are served at minimum cost but also determining the sequence of voyages each vessel will undertake. The possibility of chartering additional vessels to cover capacity shortfalls is also often considered. Reinforcing this, [43] state that in liner shipping, fleet deployment aims to assign ships to port rotations in a way that either maximizes profits or minimizes operational costs. From a broader merchant shipping perspective, [44] note that fleet deployment encompasses a wide array of issues, including fleet operations, scheduling, routing, and even aspects of fleet design, all typically guided by economic criteria such as profitability, income, or cost reduction. Even earlier, addressing bulk carrier management, [45] highlighted the challenge of managing excess transport capacity, where decisions include which ships to operate versus keep idle, or even sell or charter out, alongside strategies like slow steaming to optimize profitability while meeting customer demands.

The fleet deployment problem is not confined to maritime logistics. In the context of modern mobility solutions like EV sharing systems, [46] identify fleet deployment as a crucial tactical planning issue, distinct from daily operational concerns like fleet rebalancing. Here, planning decisions include determining station locations, overall fleet size,

and the strategic deployment of vehicles across these stations. This perspective, which emphasizes the number and initial placement of vehicles, is also echoed in the domain of emergency services. For example, [26] frame the deployment problem for emergency medical services as one that optimizes the number of ambulances hosted at each designated station to ensure efficient response capabilities.

Across these diverse applications, the fleet deployment problem often involves a core set of decisions: determining the appropriate number of vehicles or vessels (fleet sizing), allocating these units to specific bases or stations, and assigning them to tasks, routes, or service areas. The overarching goal is typically to enhance efficiency, minimize costs, maximize service levels, or improve profitability, making it a cornerstone of operational research and management science.

2.3.3 Assignment Problem

The assignment problem stands as a cornerstone in the field of optimization and operations research. As noted by [47], it was among the first linear programming problems to be extensively studied. Its enduring relevance stems from its frequent occurrence in practical applications and its fundamental role within network flow theory, underpinning a variety of other significant problems such as the shortest path, weighted matching, transportation, and minimum cost flow problems [47]. Broadly, the assignment problem addresses the challenge of allocating a set of resources to a set of tasks in the most efficient manner. [12] describe it as the minimization of the cost associated with assigning N tasks to M machines or agents, where each task is assigned to precisely one machine, subject to the capacity constraints of the machines. The versatility of this problem is evident in its wide range of applications across diverse domains, including facility location, transportation networks, communication systems, machine scheduling, and vehicle routing problems [12]. A specific instance of this can be seen in emergency services, where, as [26] point out, the assignment problem can involve determining which serving stations are assigned to respond to specific demands. [48] characterizes the assignment problem as a classic example of a combinatorial problem for which efficient algorithms exist. A common illustrative scenario involves assigning the best person for each task. In this setup, there are n persons available to perform n distinct tasks, and a cost, denoted as , is associated with assigning person i to task j. The objective is to find an assignment that minimizes the total cost. [48] present the following mathematical formulation for this problem:

$$\min_x \; Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} \tag{2.7}$$

$$\text{s.c.} \quad \sum_{j=1}^{n} x_{ij} = 1, \forall i = 1, \ldots, n \tag{2.8}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \forall j = 1, \ldots, n \tag{2.9}$$

$$x_{ij} \in \{0, 1\}, \forall i, j = 1, \ldots, n \tag{2.10}$$

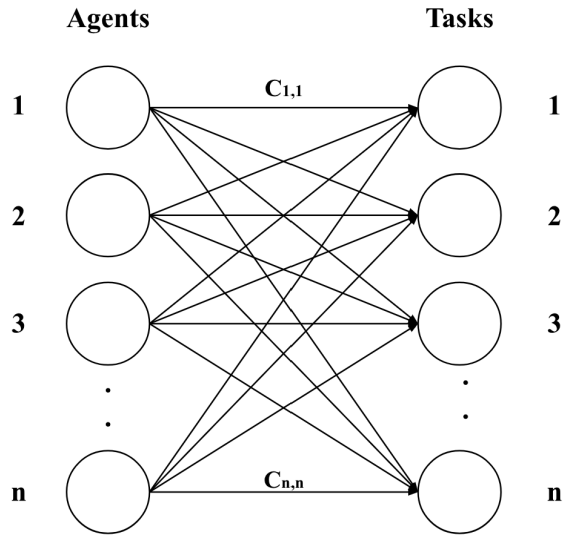

Figure 2.5: Linear Assignment Problem [8]

Here, is a binary decision variable that equals 1 if person is assigned to task , and 0 otherwise. The first set of constraints ensures that each person is assigned to exactly one task, while the second set ensures that each task is performed by exactly one person.

2.3.4 Dispatching Problem

Dispatching represents a crucial operational decision process, particularly prominent in service systems that require timely responses to incoming demands. It involves the real-time allocation of available resources (e.g., vehicles, personnel) to incoming requests or tasks. In the context of emergency medical services (EMS), [28] describe the ambulance dispatching model as one that allocates emergency calls to vehicles based on their location. Similarly, [49] define dispatching as the act of selecting which ambulance to send to an emergency call. The inherent nature of these problems often aligns with what [50] described as "online" and “real-time” optimization, where decisions must be made sequentially as new information (e.g., new service requests) arrives, often without full knowledge of future events.

The dispatching process itself, especially in emergency contexts, involves a sequence of well-defined steps. As detailed by [51], the process typically begins when a request is received and logged by a dispatcher. This initial phase involves gathering relevant data and determining the priority of the call. Once a request is processed and if a suitable idle vehicle is available, it is assigned to the task and is expected to proceed to the incident location. The overall time taken for this dispatching action includes periods for information gathering, identifying the appropriate resource, and any necessary preparation for the crew or vehicle. Following travel to the scene, the assigned unit provides the required service. Upon service completion, the unit then departs the scene, potentially proceeding to another destination such as a hospital if applicable [51].

The challenge of dispatching, while extensively studied in EMS, is not limited to this domain. It is also a key consideration in other transportation systems, such as the op-

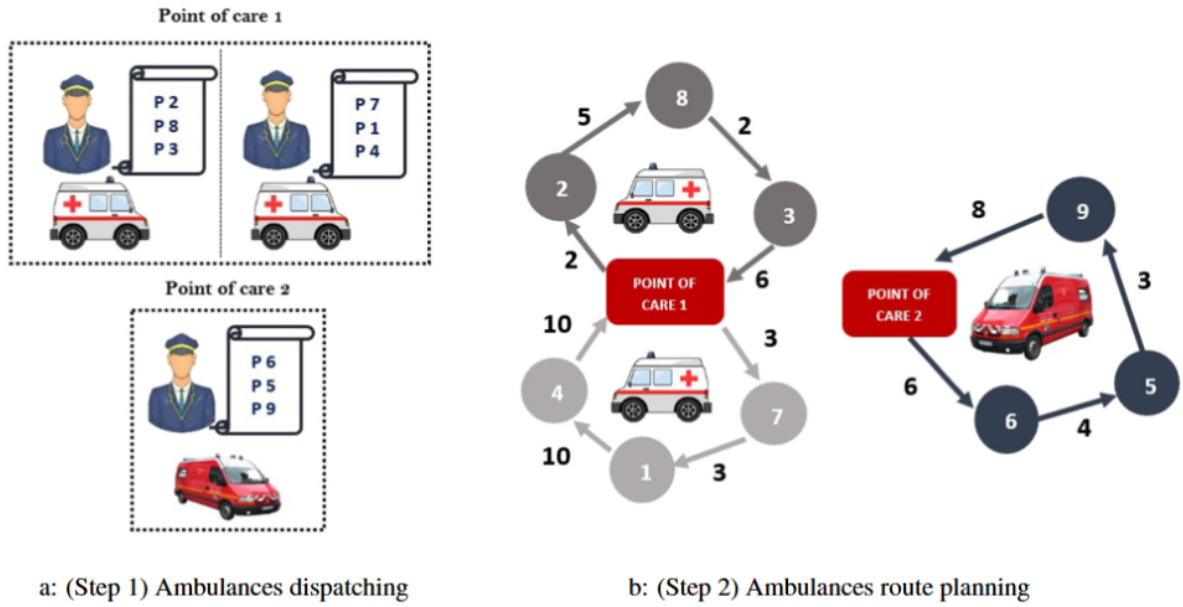

Figure 2.6: Illustration of the Ambulance Dispatching Process in Emergency Medical Services [9]

erational dispatching of buses [52] and the real-time dispatching of trains [53]. Across these applications, the core objective remains the efficient and effective assignment of operational units to demands as they arise.

2.3.5 Vehicle Routing Problem

The Vehicle Routing Problem (VRP) is a cornerstone of combinatorial optimization and logistics, with its origins tracing back to the seminal work of [54]. It addresses the challenge of designing optimal routes for a fleet of vehicles, typically originating from and returning to a central depot, to serve a set of geographically dispersed customers. The primary objective is often to minimize total travel distance or cost, while adhering to various constraints such as vehicle capacity and ensuring each customer is visited exactly once. The VRP is known to be NP-hard, meaning that finding an exact optimal solution becomes computationally intractable for large-scale instances, yet its practical applications are widespread, including goods distribution, waste collection, and parcel delivery [55].

Given its broad applicability, numerous VRP variants have been developed to address specific operational complexities encountered in real-world scenarios. A general mathematical formulation for the VRP can be described as follows: $V = \{v_0, v_1, \ldots, v_n\}$ be the set of nodes, where v_0 represents the depot and $(v_1, \ldots, v_n)$ represent the customers. Let c_{ij} be the cost of travel between node i and node j The binary variable x_{ij}^k equals 1 if vehicle k travels directly from node i and node j and 0 otherwise. The objective is to:

$$\min Z = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V, i \neq j} c_{ij} x_{ijk} \tag{2.11}$$

Subject to various constraints, including:

- Each customer is visited exactly once by one vehicle.
- All routes start and end at the depot.
- Vehicle capacity constraints are respected.
- Subtour elimination constraints.

The precise formulation and constraints will vary significantly depending on the specific VRP variant being addressed.

2.3.5.1 Vehicle Routing Problem with Time Windows (VRPTW)

The Vehicle Routing Problem with Time Windows (VRPTW) is a significant extension of the classic Vehicle Routing Problem (VRP), where each customer must be serviced within a predefined time interval, known as a time window $[e_i, l_i]$. Here, e_i represents the earliest service start time and l_i the latest service start time for customer i. This added temporal constraint, requiring adherence to specific service periods, significantly increases the problem's complexity compared to the basic VRP [55].

The primary objective in VRPTW typically remains the minimization of total travel distance or operational cost, while ensuring that all customers are served within their respective time windows and that vehicle capacities are not exceeded.

A key set of constraints in VRPTW formulations ensures that the arrival time at a customer, say a_j, respects the time window and logically follows from the departure from a preceding customer i (with service time s_i and travel time t_{ij} from i to j). This can be represented conceptually as:

$$a_j \geq a_i + s_i + t_{ij} \quad \text{(if arc } (i,j) \text{ is used by the same vehicle)} \tag{2.12}$$

$$e_j \leq a_j \leq l_j \tag{2.13}$$

The VRPTW has been extensively studied, with numerous exact and heuristic algorithms proposed for its solution, reflecting its importance in real-world logistics.

2.3.5.2 Multi-Depot Vehicle Routing Problem with Time Windows (MDVRP-TW)

Many practical logistics operations involve not just a single central depot but multiple distribution centers from which vehicles are dispatched. The Multi-Depot Vehicle Routing Problem (MDVRP) addresses such scenarios where a company operates from several depots to serve its customers [56]. In the MDVRP, each vehicle is typically assigned to

a specific origin depot, and its route must start and end at that designated depot. This introduces an additional layer of complexity, as the problem involves simultaneously assigning customers (or service areas) to depots and then constructing optimal routes for vehicles from those assigned depots.

When the constraints of customer time windows are integrated with a multi-depot operational structure, the problem evolves into the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW). This variant, illustrated in Figure 2.7 and with key time points shown in Figure 2.8, considers both multiple dispatch locations and strict service time intervals for customers simultaneously [57]. In the MDVRPTW, each vehicle originates from and must return to one of several available depots, and every customer must be served within their specified time window. This problem formulation is highly relevant for organizations with distributed logistics networks that are committed to meeting stringent service level agreements regarding delivery or service times. The primary goal in MDVRPTW is to minimize overall operational costs (such as total travel distance or the number of vehicles deployed) while satisfying all depot assignment, vehicle capacity, route continuity, and customer time window constraints. Due to its compounded complexity arising from both multi-depot operations and time window restrictions, the MDVRPTW remains an active and challenging area of research in operations research and combinatorial optimization [10].

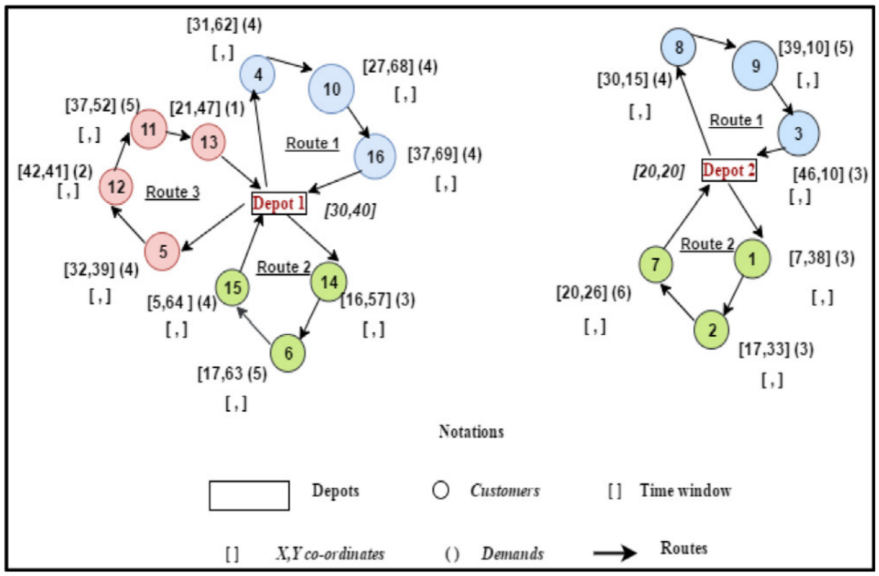

Figure 2.7: MDVRPTW Illustration [10]

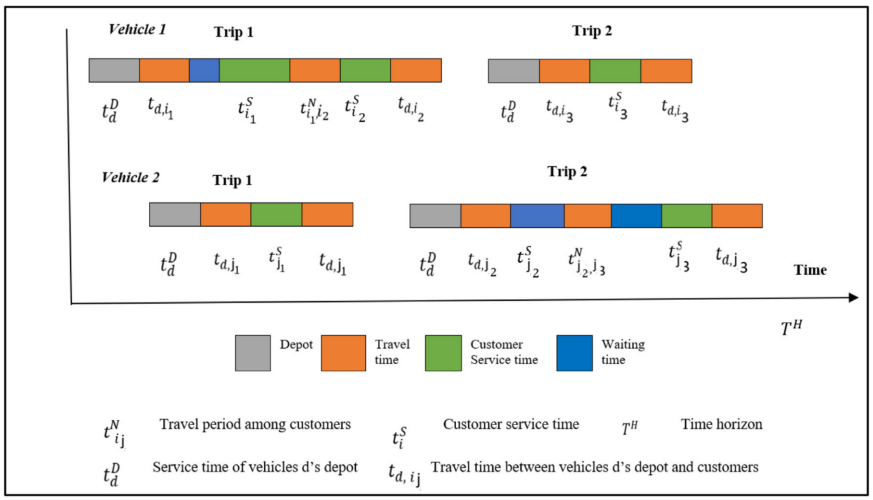

Figure 2.8: Important Time Points for MDVRPTW [11]

2.3.5.3 Electric Vehicle Routing Problem (EVRP)

The increasing adoption of electric vehicles (EVs) in logistics and transportation has given rise to the Electric Vehicle Routing Problem (EVRP) and its many variations. Unlike conventional VRPs that primarily focus on minimizing distance or time with fuel being a less restrictive constraint, the EVRP must explicitly account for the limited driving range of EVs and the operational necessity for recharging [58, 55]. This introduces new sets of decision variables and constraints related to:

- Battery capacity and energy consumption rates, which can be influenced by factors such as vehicle load, speed, terrain, and ambient temperature.
- The strategic location and operational availability of charging stations.
- The time required for recharging, which can vary significantly based on the type of charger (e.g., Level 2, DC fast charger) and the battery's state of charge.
- Policies regarding charging, such as whether partial or full charging is permitted or optimal.

The objective in EVRPs often involves minimizing total operational costs, which may include not only travel costs (related to energy consumption) but also costs associated with charging, battery degradation over time, and potentially the cost of time spent charging. EVRPs can also incorporate additional complexities such as customer time windows (leading to the E-VRPTW), multiple depots (MD-EVRP), and heterogeneous fleets. Effectively addressing the EVRP is crucial for the efficient, economically viable, and sustainable deployment of electric vehicle fleets. Recent research continues to explore sophisticated extensions, including applications in on-demand electric bus routing [59] and the integration of autonomous delivery EVs [60], reflecting the evolving landscape of electric mobility.

Table 2.1: Comparison of different papers addressing similar problems

Papers	Sector	Problem Treated			Problem Features	Approach	Modeling	Objective(s)
		D	DS	RT				
[26]	Emergency	✓	✓	-	Stochastic demand and traffic state	Scenario Generation, SAA (MC)	MIP BD	Total cost
[27]	Traffic	✓	✓	-	Stochastic demand, time and location	Scenario Generation SAA (MC)	MIP VNS	Total cost
[28]	Medical	-	✓	✓	Deterministic	-	MIP	Total cost
[29]	Fire trucks	-	✓	-	Stochastic demand and driving time	MDP	OSI, OSIA	Fraction of late arrivals
[31]	Medical	✓	✓	-	Stochastic demand	Scenario Generation	MIP	Fraction of calls with late response times
[33]	Medical	-	✓	-	Stochastic demand	Scenario Generation, SAA (MC)	MIP HDT, SBG	Total cost
[30]	Emergency	-	✓	-	Stochastic demand and response time	MAS	MIP	Travel time
[61]	Army	✓	✓	-	deterministic	-	Bi-obj MIP GA	Total cost Travel time
[36]	Medical	-	✓	✓	deterministic	-	MIP LNS	Latest service completion time
[62]	Traffic	-	✓	-	Stochastic traffic state	Queuing theory	MIP	Travel time
Our paper	**Service vehicle**	✓	✓	✓	**deterministic**	**-**	**Bi-obj MIP**	**Fleet size** **Response time**

D: Deployment, **DS**: Dispatching, **RD**: Redeployment, **RT**: Routing, **SAA**: Sample average approximation,**MC**: Monte Carlo, **MAS**: Multi-Agent System, **VNS**: Variable Neighborhood Search,**MDP**: Markov Decision Process, **BD**: Benders Decomposition, **OSI**: One-Step Improvement, **OSIA**: One-Step Improvement Approximation Heuristic, **GA**: Genetic Algorithm, **HDT**: Temporal Decomposition Heuristic, **SBG**: Lagrangian Relaxation-based Heuristic, **LNS**: Large Neighborhood Search.

To address STM's challenge, the literature on emergency vehicle management offers valuable insights. Researchers have long examined problems analogous to STM's, including those encountered in ambulance, fire truck, and other emergency service operations. In this context, the literature is generally divided into several interconnected problems: location, deployment, assignment, dispatching, and relocation. The location problem focuses on identifying the best sites for stations, while the deployment problem determines the number of vehicles that should be stationed at each location. The assignment problem determines which service areas or demand zones are covered by which stations, and the dispatching problem decides which vehicle is sent to respond to an incident [26]. Additionally, the relocation problem involves repositioning idle vehicles to reduce future response times. In the realm of traffic emergency systems, [27] defines deployment as the process of allocating vehicles to stations, and dispatching as the real-time decision of sending a vehicle to meet an incident demand. [28] further emphasizes that most dispatching models assign tasks to the nearest available resource, a strategy also highlighted by [29], who notes that recent work has increasingly focused on the joint problem of dispatching and proactive relocation to enhance overall coverage.

These operational issues are closely linked to strategic planning. According to the classification proposed by [31], decision-making can be divided into three levels: first, there is the strategic level with long-term decisions involved such as station location and fleet sizing. Second, the tactical level is covering medium-term issues like baseline deployment and shift scheduling. Finally, the operational level is focusing on short-term decisions like dispatching and dynamic relocation. This classification is considered in [32] and [33] with strategic decisions laying the groundwork for operational efficiency by determining where vehicles should be stationed, while operational decisions ensure that vehicles are dispatched and, if necessary, redeployed effectively in real time.

Recent reviews, such as that by [37], indicate a trend towards integrated approaches. These models combine multiple methodologies often sequentially or iteratively to harness the strengths of each, thereby optimizing both deployment and dispatching simultaneously. This integrated perspective is particularly relevant for STM's challenge of managing an electric fleet to improve performances.

Because of the dynamic and uncertain conditions, simulation modeling has emerged as a crucial tool. Researchers employ various types of discrete event simulations [38], continuous, hybrid, Monte Carlo, and agent based [30] to capture the uncertainty inherent in emergency scenarios. Simulations help in two main ways: by assessing the impact of parameter changes on system performance and by evaluating the robustness of mathematical programming solutions [37]. Others stochastic approaches, such as queuing theory and Markov decision processes, provide additional insights into system dynamics and response times [26, 39].

Compared to similar studies in emergency services where the primary objective is minimizing rescue time due to its direct impact on saving lives, our research takes a distinct approach. Our study focuses on optimizing the fleet size of STM's electric service vehicles. This decision is crucial from an operational perspective: for instance, if multiple incidents occur simultaneously or if there is a consistent pattern in incident frequency, the STM must avoid unnecessary capital expenditure on acquiring redundant vehicles. Therefore, our goal is to reduce investments by integrating long-term strategic decisions with short-

term operational ones. In addition to the decisions related to fleet sizing, deployment and dispatching, the routing of service vehicles plays an equally critical role at the operational level. Although the classic VRP primarily focuses on minimizing routing costs [63] and has evolved to include numerous practical constraints such as vehicle capacity, time windows, and multi-trip operations [35] its adaptations in the emergency response context have been less extensively explored. For example, [36] emphasizes that while research in disaster response typically concentrates on stationing and dispatching ambulances, only limited attention has been paid to ambulance routing, particularly when multiple pick-ups are involved. In our framework, however, the routing of STM's service vehicles is crucial since a single vehicle may respond to several incidents during the same shift. Drawing inspiration from the MDVRP-TW [64, 57], our model leverages established routing principles to efficiently manage the complex interplay between multiple depots, dispatching, and time window constraints driven by the incident's occurrence date and the need for prompt intervention while also accommodating multi-incident pickups.

Table 2.1 presents a comparative overview of related works, highlighting the nature of the challenges tackled, the methodological approaches adopted, and the distinctive aspects of our contribution most notably, the adoption of a multi-objective optimization model. Our work introduces a strategic decision-support tool for operations planning, aiming to minimize response times when feasible while simultaneously optimizing fleet size to balance cost-efficiency with demand satisfaction.

Despite the extensive focus on emergency vehicle deployment and dispatching, to the best of our knowledge, no previous work has specifically addressed the optimization of electric fleet sizing, deployment and routing under urban operational conditions for public transit operations, while simultaneously considering both strategic and operational challenges. This study aims to fill this gap. Our contribution lies in formulating a real-world problem and developing a novel mathematical model that incorporates detailed factors such as incident timing and location, travel times, distances, and the specific constraints of electric vehicles.

2.4 Demand Forecasting in Operational Planning

Effective operational planning, particularly for strategic decisions such as fleet sizing, resource allocation, and service deployment, is critically dependent on accurate forecasts of future demand. As highlighted by [65], demand forecasting is a crucial component for any organization aiming to predict and estimate future requirements to facilitate better decision-making. In the context of the STM's Operational Supervisor (OS) vehicles, "demand" translates to the anticipated number, timing, and geographical distribution of incidents necessitating OS intervention. Generating reliable forecasts, often at a granular level (e.g., hourly or by shift) over a defined planning horizon, is essential for aligning OS vehicle availability with expected service calls, thereby optimizing response times and overall operational efficiency [66].

The field of demand forecasting, a key area of predictive analytics, controls numerous downstream activities in service and supply chain management. Accurate projections

inform decisions on capacity expansion (e.g., fleet size), resource allocation (e.g., vehicle deployment to depots), and can even influence longer-term strategic planning. While advanced technology enables the dissemination of real-time data, forecasting remains vital for proactive planning to remove bottlenecks and ensure the efficient use of resources. The literature presents a spectrum of methodologies, from traditional statistical models to more contemporary machine learning and hybrid approaches [65].

2.4.1 Time Series Forecasting Methods

Many established forecasting techniques analyze historical time series data to identify underlying patterns such as trends, seasonality, and cyclical variations, which are then extrapolated to predict future values. The choice of method often depends on the characteristics of the data. [65] categorize these traditional statistical models as one of the main approaches to demand forecasting. Some commonly employed methods include:

- **Simple Averages and Moving Averages:** These fundamental techniques provide a baseline. A simple average uses the mean of all historical data. A Moving Average (MA) smooths out short-term fluctuations by calculating the average of a fixed number of recent observations, giving equal weight to each.
- **Linear Regression:** As described in [66] (implicitly, as they use regression models R1, R2, R3), regression models can be used to establish relationships between the demand (dependent variable, e.g., number of incidents) and various influencing factors (independent variables, e.g., time, day of week, special events, promotions). For instance, their R1 model considers variables like special days and promotions, R2 adds weekly partial-regressive terms, and R3 incorporates monthly and weekly dummy variables to capture seasonality [66]. This allows for the modeling of trends and the impact of external factors.
- **Exponential Smoothing Methods:** This family of methods assigns exponentially decreasing weights to past observations, giving more importance to recent data.
    - **Simple Exponential Smoothing (SES):** Suitable for data with no clear trend or seasonality, it computes a weighted average where weights decline exponentially for older data [66, 65].
    - **Holt's Linear Trend Method:** An extension of SES, Holt's method is designed for time series exhibiting a linear trend. It uses two smoothing parameters for the level and the trend of the series [66, 65].
    - **Holt-Winters Method (Triple Exponential Smoothing):** This powerful method extends Holt's approach to incorporate seasonality, in addition to level and trend [66, 65]. It is well-suited for time series with regular seasonal patterns (e.g., incidents varying by time of day, day of week, or month). The Holt-Winters equations can be adapted for additive seasonality (where seasonal variations are roughly constant) or multiplicative seasonality (where seasonal variations are proportional to the series level) [66].

- **ARIMA Models (Autoregressive Integrated Moving Average):** Also known as the Box-Jenkins method, ARIMA models are a comprehensive class of statistical models for analyzing and forecasting time series data. They combine autoregressive (AR) components (where the variable depends on its own past values) and moving average (MA) components (where the variable depends on past forecast errors), along with an integration (I) component to make the series stationary if needed [66, 65]. These models are quite robust for data that is non-stationary after differencing.

2.4.2 Advanced and Hybrid Forecasting Approaches

While classical methods are foundational, the increasing complexity of demand patterns and the availability of richer datasets have spurred the development and application of more advanced techniques.

- **Machine Learning Models:** [65] discusses various machine learning models for forecasting, including regression variants (like Poisson regression for count data, Lasso regression for high-dimensional data), Support Vector Regression (SVR), and tree-based ensemble methods like XGBoost. These models can often capture complex non-linear relationships and interactions between variables more effectively than traditional statistical models.
- **Deep Learning Models:** Models like Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, have gained popularity for time series forecasting due to their ability to learn long-range dependencies and handle volatile demand scenarios effectively [65]. Generative Adversarial Networks (GANs) have also been explored for time series generation and prediction [65].
- **Ensemble Learning and Hybrid Models:** Recognizing that individual models may not always perform optimally across all conditions, ensemble learning techniques combine the predictions of multiple diverse forecasting models to produce a more robust and often more accurate final forecast [66]. [66] specifically proposes a new heuristic ensemble approach for retail demand forecasting, which involves calculating a weighted average of MAPE (Mean Absolute Percentage Error) from different algorithms for previous weeks and adjusting weights accordingly. The idea is that different algorithms might be "champions" for different products or time periods. Hybrid models, which sequentially combine different approaches (e.g., ARIMA with a neural network to model residuals), are also a common strategy to leverage the strengths of various techniques [65].

The selection of an appropriate forecasting methodology depends on several factors, including the specific characteristics of the demand data (e.g., presence of trend, seasonality, intermittency, volatility), the nature of influencing factors [66], the forecast horizon, data availability, and the desired trade-off between model complexity, interpretability, and predictive accuracy. For the STM context, predicting the spatio-temporal occurrence of incidents will likely benefit from models that can capture seasonality, trends, and potentially the impact of external events.

2.5 Solution Approaches in the Literature

The majority of studies addressing transportation-related challenges, researchers employ tools and techniques rooted in combinatorial optimization. These solution methods are broadly classified into two main categories, as depicted in Figure 2.9. On one hand, exact methods are designed to find and rigorously prove the optimality of a solution. However, for many practical problems, particularly those of significant scale, the computation time required by these methods can increase exponentially with problem size. Consequently, on the other hand, approximate methods (often referred to as heuristics or metaheuristics) are utilized. These approaches aim to identify feasible solutions, typically of high quality, within reasonable computational timeframes, though they do not offer a guarantee of optimality.

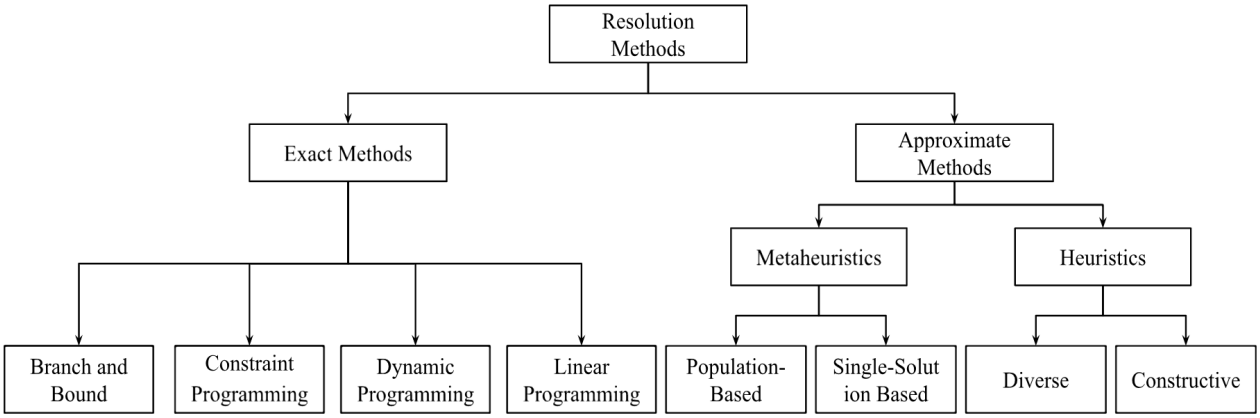

Figure 2.9: Solution Methods for Combinatorial Optimization [12]

2.5.1 Optimization Models

Mathematical optimization models provide a formal framework for finding the best possible solution to a problem given a set of constraints and one or more objectives. In the simplest case, **mono-objective optimization**, the goal is to identify a solution that optimizes a single, well defined objective function such as minimizing total operational cost, minimizing total travel distance, or maximizing service level, subject to various operational constraints. The outcome of such an optimization is typically a single "best" feasible solution according to this sole criterion. However, many real world problems, especially in complex systems like transportation and logistics, inherently involve multiple, often conflicting, objectives that decision-makers wish to address simultaneously. For example, an organization might aim to minimize operational costs while simultaneously minimizing vehicle response times and maximizing the equity of service coverage across different areas. This leads to the domain of **Multi-Objective Optimization (MOO)**. In MOO problems, it is generally not possible to find a single solution that is optimal for all objectives at the same time, as improving one objective often necessitates a compromise or degradation in another. Consequently, the focus shifts from finding a single optimal solution to identifying a set of solutions that represent the best possible trade-offs among the competing objectives.

2.5.1.1 Pareto Optimality:

In MOO, the concept of a single optimal solution is replaced by the notion of Pareto optimality (also known as Pareto efficiency or non-dominated solutions). A feasible solution is considered Pareto optimal if it is impossible to improve its performance on one objective function without worsening its performance on at least one other objective function.[13]

In MOO, the concept of a single optimal solution is replaced by the notion of *Pareto optimality* (also known as Pareto efficiency or non-dominated solutions). A feasible solution is considered Pareto optimal if it is impossible to improve its performance on one objective function without worsening its performance on at least one other objective function [67].

Formally, assuming all k objective functions $f_j(x)$ for $j = 1, \ldots, k$ are to be minimized, a feasible solution x^* is (strongly) Pareto optimal if there is no other feasible solution x such that:

$$f_j(x) \leq f_j(x^*) \quad \text{for all } j = 1, \ldots, k,$$

with at least one strict inequality:

$$f_m(x) < f_m(x^*) \quad \text{for some } m.$$

A feasible solution x^* is *weakly Pareto optimal* if there is no feasible solution x such that:

$$f_j(x) < f_j(x^*) \quad \text{for all } j = 1, \ldots, k.$$

The set of all Pareto optimal solutions forms the *Pareto front* (or efficient frontier). In problems with non-convex feasible objective and decision spaces (e.g., Mixed-Integer Programs), the set of efficient solutions can be further partitioned into *supported* and *non-supported* efficient solutions.

Supported efficient solutions are those that can be found as optimal solutions to a weighted sum of the individual objective functions for some set of non-negative weights. Non-supported efficient solutions, however, cannot be found this way and lie in the "gaps" between supported solutions, as illustrated in Figure 2.10.

Various methods exist to generate or approximate the Pareto front, or to select a preferred solution from it. These are often categorized as:

- *a-priori* (preferences defined before optimization),
- *interactive* (decision-maker iteratively refines preferences),
- *a-posteriori* (Pareto set is generated first, then the decision-maker chooses).

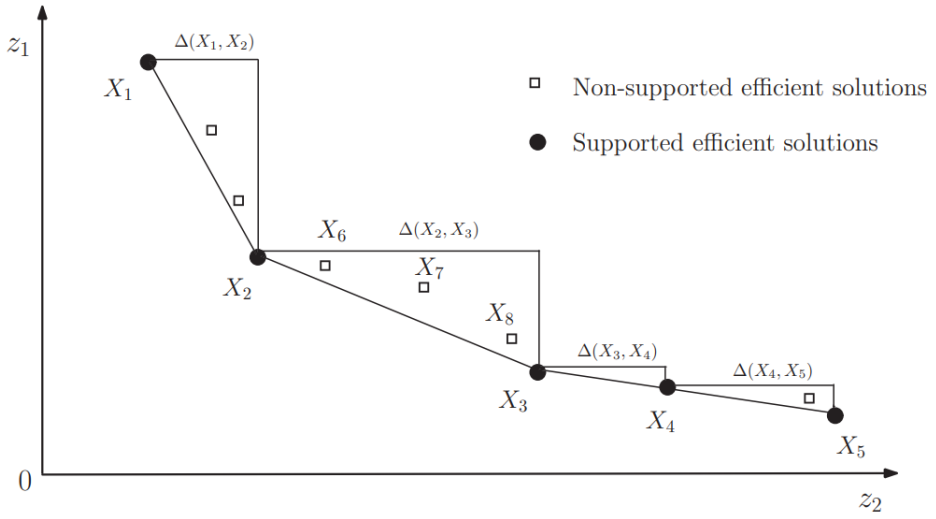

Figure 2.10: Supported and non-supported efficient solutions in the objective space Z [13]

2.5.1.2 Lexicographic Approach:

One straightforward a priori method for handling multiple objectives, particularly when a clear hierarchy of importance among them can be established by the decision-maker, is lexicographic optimization. This approach assumes that the decision-maker has a strict ordinal preference for the objectives, meaning they can rank them from most important to least important. As highlighted by [68], in lexicographic optimization, a finite number of objective functions are considered and are optimized on a feasible set in a lexicographic order. This implies that lower-priority objectives are optimized only to the extent that their optimization does not degrade the optimal values already achieved for higher-priority objectives. The relevance of lexicographic optimization can be found in multiple criteria decision-making as well as in mathematical programming [68].

A lexicographic minimization problem with k objective functions can be generally stated as:

$$\min \quad (f_1(x), f_2(x), \ldots, f_k(x)) \tag{2.14}$$
$$\text{s.t.} \quad x \in X \tag{2.15}$$

where x is the vector of decision variables, X is the feasible region, and $f_1, f_2, \ldots, f_k$ are the k objective functions ordered from most to least important. This problem can be solved by a sequence of k single-objective optimization problems as outlined in Algorithm 1:

Algorithm 1: Lexicographic Optimization Procedure

Input: A set of k objective functions $f_1(x), f_2(x), \ldots, f_k(x)$ ranked in decreasing priority; feasible region X.

1: **For** $p = 1, \ldots, k$:

2: Solve the single-objective optimization problem:

$\min f_p(x)$

subject to $x \in X$ and $f_j(x) \leq f_j^* \quad \forall j < p$, where f_j^* is the optimal value from iteration j

3: Let f_p^* be the optimal value found for $f_p(x)$

4: **End for**

Output: Lexicographically optimal solution x^* that satisfies all hierarchical preferences

The solution obtained after optimizing for the k^{th} (least priority) objective function, while respecting the optimal values achieved for all $k1$ higher-priority objectives, is termed the lexicographically optimal solution. [68] notes that for linear problems (Linear Lexicographic Programs or LLP), this procedure can be adapted, and specialized methods like the lexicographic simplex method can be employed, which consider all objective functions simultaneously but prioritize them according to the lexicographic order during pivot selection. An example of an implicit application of lexicographic optimization is the two-phase method of linear programming, where the first phase minimizes the sum of artificial variables (highest priority) before the actual objective function is optimized in the second phase (lower priority) [68]. This hierarchical approach is particularly useful when the decision-making entity, such as a company, can clearly define an ordered list of goals or targets, where achieving a higher-ranked goal takes absolute precedence over lower-ranked ones.

2.5.2 Exact Methods

Exact methods aim to find a provably optimal solution to an optimization problem. Many transportation problems can be modeled as mathematical programs, often involving integer decision variables, leading to Integer Linear Programs (ILPs). While the underlying Linear Programs (LPs) where variables are continuous can often be solved efficiently, the introduction of integrality significantly increases computational complexity. When addressing such problems, a fundamental distinction arises based on the number of objectives being optimized.

2.5.2.1 Linear, Integer, and Mixed-Integer Linear Programming

Linear Programming (LP) stands as a cornerstone of mathematical optimization, gaining prominence with G.B. Dantzig's development of the simplex method in 1947 [69]. An LP problem involves the optimization (maximization or minimization) of a linear

objective function, subject to a finite set of linear equality or inequality constraints. A standard formulation can be written as:

$$\text{Min (or Max)} \quad Z = \sum_{j=1}^{n} c_j x_j \tag{2.16}$$

$$\text{subject to:} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \forall i \in \{1, \ldots, m\} \tag{2.17}$$

$$x_j \geq 0 \quad \forall j \in \{1, \ldots, n\} \tag{2.18}$$

Here, x_j are the decision variables, c_j are the objective function coefficients, a_{ij} are the constraint coefficients, and b_i are the right-hand side values of the constraints.While problems with very few variables (typically two or three) can be solved graphically by identifying the feasible region and the optimal corner point, as illustrated in Figure 2.11, the simplex algorithm provides a systematic and efficient procedure for solving larger LP instances by iteratively moving between extreme points (vertices) of the feasible polytope [69].

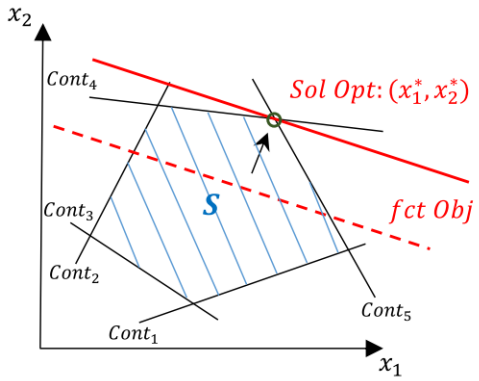

Figure 2.11: Example of a graphical solution of a linear program with 5 constraints[12]

Many real-world decision problems, however, require some or all decision variables to take on only integer values (e.g., the number of vehicles to dispatch, or binary decisions like whether to open a facility). This leads to **Integer Linear Programming (ILP)**, where all decision variables are restricted to be integers, or **Mixed-Integer Linear Programming (MILP),** which allows for a combination of integer and continuous variables [70]. A standard form for an MILP can be represented as:

$$\text{Min (or Max)} \quad Z = \sum_{j=1}^{n} c_j x_j + \sum_{k=1}^{p} d_k y_k \tag{2.19}$$

$$\text{subject to:} \quad \sum_{j=1}^{n} a_{ij} x_j + \sum_{k=1}^{p} e_{ik} y_k \leq b_i \quad \forall i \in \{1, \ldots, m\} \tag{2.20}$$

$$x_j \in \mathbb{Z}^+ \quad \forall j \in \{1, \ldots, n\} \tag{2.21}$$

$$y_k \in \mathbb{R}^+ \quad \forall k \in \{1, \ldots, p\} \tag{2.22}$$

Unlike LPs, which can generally be solved efficiently in polynomial time, ILPs and MILPs are often NP-hard. A common initial step in solving an ILP/MILP is to consider its LP

relaxation, where the integer constraints (e.g., $x_j \in \mathbb{Z}^+$)) are relaxed to allow continuous values (e.g., $x_j \in \mathbb{R}^+$). However, the optimal solution to this LP relaxation frequently does not satisfy the original integer requirements and may provide a bound that is not tight enough. The optimal integer solution often lies within the feasible region defined by the LP relaxation but not necessarily at one of its extreme points. Consequently, specialized techniques that systematically explore or decompose the integer solution space, such as branch and bound or cutting plane methods, are necessary to solve ILPs and MILPs to optimality.

2.5.2.2 Branch and Bound (B&B)

The Branch and Bound (B&B) method, with early roots in the work of [71], is a common algorithm for solving ILPs and MILPs. It's an implicit enumeration technique. The core idea is to:

- **Branch:** Systematically divide the original problem into smaller, more manageable subproblems by fixing integer variables or adding constraints. This creates a search tree.
- **Branch:** Systematically divide the original problem into smaller, more manageable subproblems by fixing integer variables or adding constraints. This creates a search tree.
- **Prune:** If a subproblem's bound indicates it cannot lead to a better solution than one already found, or if the subproblem is infeasible or yields an integer solution, that branch of the tree can be pruned (discarded), avoiding exhaustive enumeration.

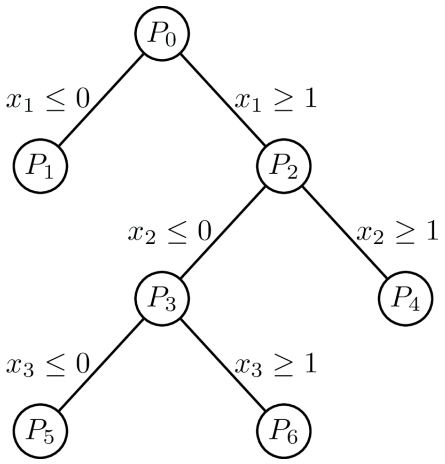

Figure 2.12: Principle of the Branch and Bound Method

The efficiency of B&B heavily depends on the quality of the bounds and the branching strategy used [70].

2.5.2.3 Other Exact Approaches

Beyond Branch and Bound, several other exact methods have been developed to tackle Integer Linear Programs (ILPs) and other combinatorial optimization problems. Dynamic Programming (DP), pioneered by R.E. Bellman in the 1950s [72], is a powerful technique particularly suited for problems that can be decomposed into a sequence of overlapping subproblems. It operates based on Bellman's "Principle of Optimality," which states that an optimal policy has the property that whatever the initial state and initial decision, the remaining decisions must constitute an optimal policy with regard to the state resulting from that first decision. DP systematically solves these simpler subproblems, stores their solutions (often in a table) to avoid redundant computations, and combines them to find the overall optimal solution, typically through a recursive formulation.

Further exact methodologies for ILPs include cutting plane methods, which iteratively add valid inequalities (cuts) to the Linear Programming (LP) relaxation to progressively tighten it and cut off fractional solutions, thereby moving closer to an integer optimum (e.g., [73]). For problems characterized by an exceptionally large number of variables, column generation techniques are often employed; these methods start with a restricted subset of variables and iteratively generate and add new variables (columns) that have the potential to improve the current solution, making them particularly useful in contexts like large-scale routing or scheduling problems (e.g., [74]). Constraint Programming (CP) offers another distinct paradigm, focusing on finding feasible solutions by defining variables with their domains and a set of constraints. CP solvers then use techniques like constraint propagation (to reduce variable domains) and systematic search to find solutions that satisfy all constraints [75]; while primarily a feasibility-finding tool, it can also be used for optimization.

While these exact methods can guarantee optimality, their computational requirements, particularly for NP-hard problems, can grow exponentially with problem size. This often renders them impractical for large-scale real-world instances, thereby underscoring the critical need for efficient approximate methods capable of finding high-quality solutions within reasonable timeframes.

2.5.3 Approximate Methods

When exact methods become computationally prohibitive due to problem size or complexity, approximate methods offer a practical alternative. The primary goal of these approaches is to efficiently find high-quality feasible solutions, often sacrificing the guarantee of proven optimality for speed and applicability to large-scale, real-world problems. A major advantage of approximate methods is their ability to generate solutions relatively quickly, making them suitable for industrial applications with tight operational constraints. These methods can be broadly categorized as follows:

2.5.3.1 Heuristics

The term "heuristic," derived from a Greek word meaning "to discover and explore" [76], typically refers to problem-specific rules or sets of rules. Their main objective, as highlighted by [76], is to construct an optimization model that is easily comprehensible and provides good solutions within a reasonable computational time. In this sense, they serve to "guide discovery" or improve problem-solving [77]. Heuristics often employ intuitive strategies based on domain knowledge or human experience, which [76] notes can be crucial in designing heuristics that approach solutions faster and are more relevant to real-life situations. While generally very fast, a defining characteristic of heuristics is that they do not necessarily converge toward an optimal solution and sometimes may not even guarantee a feasible one, though the latter is less common in practical implementations [78]. This can lead to a perception of heuristics as a "less than perfect method or a lack of solution guarantee" [77]. Despite this, they are invaluable when quick, workable solutions are needed, especially under strict time or computational resource limitations, or when more sophisticated methods are not viable. Heuristics can often be distinguished by their operational strategy:

- **Constructive Heuristics:** These algorithms build a feasible solution from scratch, typically by making a sequence of decisions to add components incrementally, often based on a greedy criterion at each step, until a complete solution is formed.
- **Improvement or Exploratory Heuristics:** These approaches often start with an existing solution (which might be generated by a constructive heuristic or randomly) and attempt to enhance it through local modifications. Some heuristics in this category might also incorporate elements to explore different parts of the solution space rather than strictly following a single improvement path, thereby introducing a diversifying aspect to avoid premature convergence to poor local optima.

2.5.3.2 Metaheuristics

Metaheuristics represent a more advanced class of approximate algorithms designed to find near-optimal solutions for complex combinatorial optimization problems that are intractable for exact methods or simple heuristics. Emerging prominently in the 1980s, they have seen significant development and application. According to [79], a metaheuristic can be described as an iterative process that guides a subordinate heuristic by combining various techniques to effectively explore the solution space. They often incorporate strategies for both intensification (focusing the search in promising regions) and diversification (exploring new, unvisited areas of the solution space) to avoid getting trapped in local optima. Learning mechanisms are often employed to structure information gathered during the search process, aiming to find solutions that are very close to, or in some cases, actually optimal. Metaheuristics can be broadly classified into two main families:

- **Single-solution based metaheuristics:** These methods iteratively improve a single candidate solution (e.g., Simulated Annealing, Tabu Search).

- **Population-based metaheuristics:** These methods maintain and evolve a set of multiple candidate solutions simultaneously (e.g., Genetic Algorithms, Particle Swarm Optimization, Ant Colony Optimization).

The generic nature of metaheuristics allows them to be adapted to a wide variety of optimization problems, making them a versatile tool in operations research.

2.6 Conclusion

This chapter has provided a comprehensive review of the foundational concepts and relevant research pertaining to the optimization of vehicle fleets, particularly in contexts similar to emergency or operational support services. We have explored the core problems of facility location, fleet sizing, vehicle deployment, dispatching, and routing, examining various established and emerging solution methodologies, from exact optimization techniques to heuristics and metaheuristics. The review also highlighted the specific challenges and considerations introduced by vehicle electrification, such as range limitations and charging infrastructure, which are increasingly pertinent.

While the literature offers a wealth of knowledge on individual aspects of fleet management, a notable gap exists in integrated approaches specifically addressing the strategic sizing and deployment of electrified operational support vehicle fleets, considering their unique constraints and objectives. Given that the electrification of an entire vehicle fleet, such as the one undertaken by the STM, represents a substantial financial investment, and the optimal sizing and deployment of this specialized electric fleet has not yet been comprehensively addressed in a holistic manner, this study seeks to fill that gap. The insights gained from this literature review will inform the model development, and solution approach presented in the subsequent chapters, aiming to provide a robust framework for the STM's critical decision-making process.

Chapter 3

Problem formulation and Mathematical Modeling

Following the detailed description of the problem and contextual analysis of the optimization challenge presented in Chapter 1, and the comprehensive review of the relevant literature and existing methodologies discussed in Chapter 2, this chapter focuses on the formal development of a mathematical model to address this multifaceted problem. The previous chapters have established the operational needs, identified the existing gaps in research, and provided the theoretical underpinnings for our approach. Now, we transition to translating these insights into a precise and solvable mathematical framework.

This chapter presents the formal mathematical formulation of the optimization problem faced by the STM. We begin by developing a conceptual model that captures the essential dynamics of the OS vehicle system. This conceptual understanding is then translated into a rigorous mathematical formulation. This model aims to incorporate the key decision variables related to fleet sizing, deployment, and routing. The operational constraints imposed by electric vehicle characteristics (such as range and charging) and service level expectations, and the strategic objectives of minimizing costs while maximizing operational effectiveness. The resulting formulation will serve as the foundation for the computational experiments and analysis detailed in the subsequent chapter.

3.1 Conceptual model

To effectively address the STM's challenge of optimizing its electrified OS vehicle fleet, a clear understanding of the operational needs and the interactions between different system components is essential. The core need is to strategically size the OS vehicle fleet and deploy these vehicles from various depots to respond to incidents across the bus network in a timely and efficient manner, especially considering the transition to electric vehicles with their inherent range and charging constraints. The nature of the decisions involved such as whether a vehicle is used, which depot a vehicle starts from, and the sequence of incidents a vehicle serves lends itself naturally to a **Mixed-Integer Linear Programming (MILP) approach**. This modeling paradigm enables the representation of different types of variables, including binary variables (e.g., whether a vehicle is assigned), integer variables (e.g., the number of used vehicles), and real-valued variables (e.g., travel times), all of which are critical to effective fleet deployment and routing.

The typical operational process for an OS vehicle during a shift can be summarized as follows:

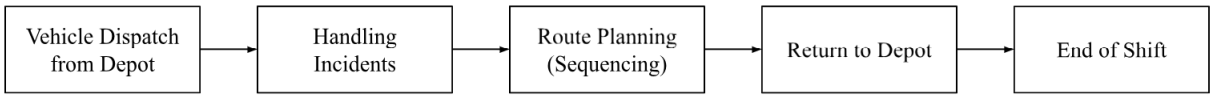

Figure 3.1: Typical Shift Process for an OS Vehicle

1. **Vehicle Dispatch from Depot:** At the commencement of each operational shift, OS vehicles are dispatched from one of the available depots. Each vehicle is typically assigned to a specific depot as its starting and ending point for the shift, based on strategic considerations related to incident distribution and network coverage.

2. **Handling incidents:** Incidents occur across the network and require an OS vehicle's intervention. Each incident is characterized by an occurrence time.

3. **Route Planning (Sequencing of Incidents):** After initiating or completing service at one incident, an OS vehicle may proceed to the next assigned incident on its planned route. The sequence in which a vehicle visits multiple assigned incidents during its shift must be optimized to ensure timely responses and efficient resource utilization.

4. **End of Route and Return to Depot:** After servicing all assigned incidents within its tour, or as dictated by other constraints (e.g., shift duration, battery level), the vehicle proceeds to its designated final destination, which is typically its originating depot for recharging and shift changeover.

5. **End of Shift:** Each OS vehicle operates within a defined shift duration (e.g., 6 hours), after which its tour must conclude.

To address the STM's optimization challenge, our goal is to develop a comprehensive decision-support tool based on the MILP framework. This tool will process key inputs,

such as a list of incidents (each with its specific time of occurrence and geographical location), depot locations, vehicle characteristics (like battery capacity), and operational rules (e.g., shift durations). The primary objective of the model will be to minimize critical performance metrics, notably the total number of OS vehicles required (fleet size) and the overall system response time to incidents. This minimization will be subject to a set of operational and routing constraints ensuring that all incidents are serviced, shift durations are not exceeded, and vehicles operate out of designated depots. The desired outputs from this optimization model will be the optimal number of OS vehicles needed and the detailed routes and schedules for each deployed vehicle, ensuring all incident demands are met efficiently. Figure 3.2 provides a overview of this input-process-output structure.

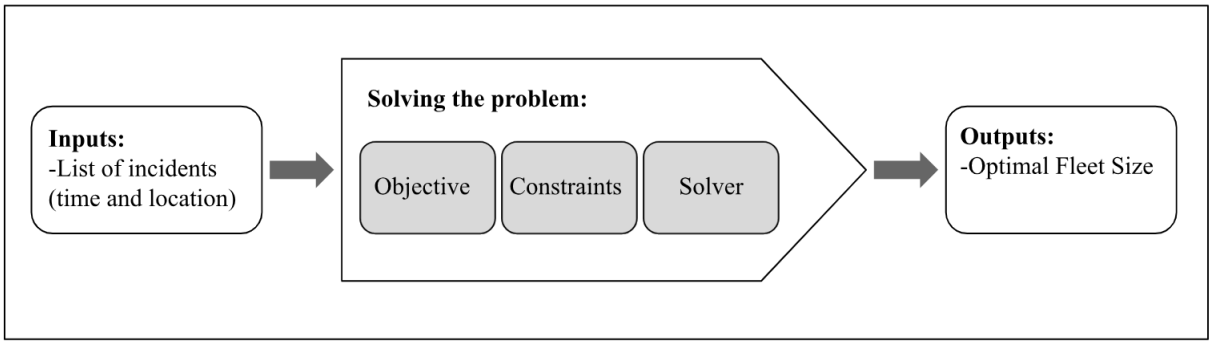

Figure 3.2: Overview of the solution method process

3.2 Modeling the Problem

The problem is modeled as a graph $G = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{A}$ the set of arcs connecting them. The node set $\mathcal{V}$ includes both the set of potential incident locations $\mathcal{N}$ and the set of depots $\mathcal{D}$, such that $\mathcal{V} = \mathcal{D} \cup \mathcal{N}$. Each arc $(i, j) \in \mathcal{A}$ represents a feasible direct travel path between nodes i and j, with an associated travel time t_{ij} or distance. Figure 3.3 provides a conceptual illustration of the spatial distribution of depots and incident locations across the service region, which is divided into operational sectors.

This study addresses two primary, potentially conflicting, objectives:

- Minimizing the total number of OS vehicles deployed (fleet size).
- Minimizing the overall response time to incidents (time taken from incident occurrence to OS vehicle arrival).

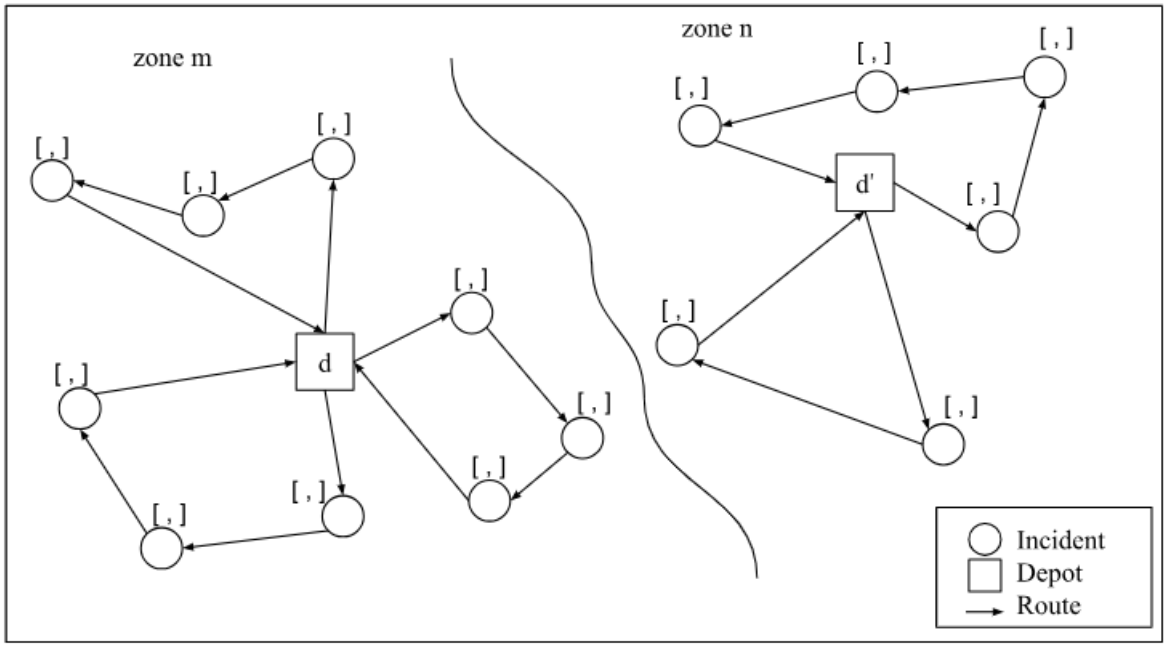

Figure 3.3: Distribution of depots and incidents across the service region

The STM has defined a clear priority between these objectives: minimizing fleet size takes precedence over minimizing response time. Consequently, our formulation adopts a lexicographic bi-objective Mixed-Integer Linear Programming (MILP) approach. This approach first optimizes the primary objective (fleet size) and then, subject to achieving that optimal fleet size, optimizes the secondary objective (response time). This model is designed to support both strategic planning (determining the number of vehicles and their depot assignments) and operational efficiency (optimizing vehicle routes during service shifts). To facilitate the mathematical modeling of this complex problem, several simplifying assumptions are made:

- All incidents or interventions are treated with equal operational priority for dispatch.
- Each incident requires the intervention of exactly one OS vehicle.
- The service time required to resolve each incident is known and deterministic, based on empirical historical data.
- The timing and location of incident occurrences for a given planning horizon are assumed to be known (e.g., derived from the incident forecasting model discussed in Chapter 2).
- The OS vehicle fleet consists of homogeneous vehicles, meaning all vehicles share identical performance characteristics, including speed, battery capacity, and service capabilities.
- The travel speed of the vehicles is assumed to be known and constant, allowing for accurate and deterministic travel time estimations between locations.

- Travel distances (and thus times) between any two nodes are pre-calculated, for instance using the Haversine formula for geodesic distances, and are considered symmetric.

3.2.1 Building the model

This section details the notations, decision variables, and constraints that constitute the MILP formulation developed. Figure 3.4 provides a visual aid to better understand some of the key variables and their relationships.

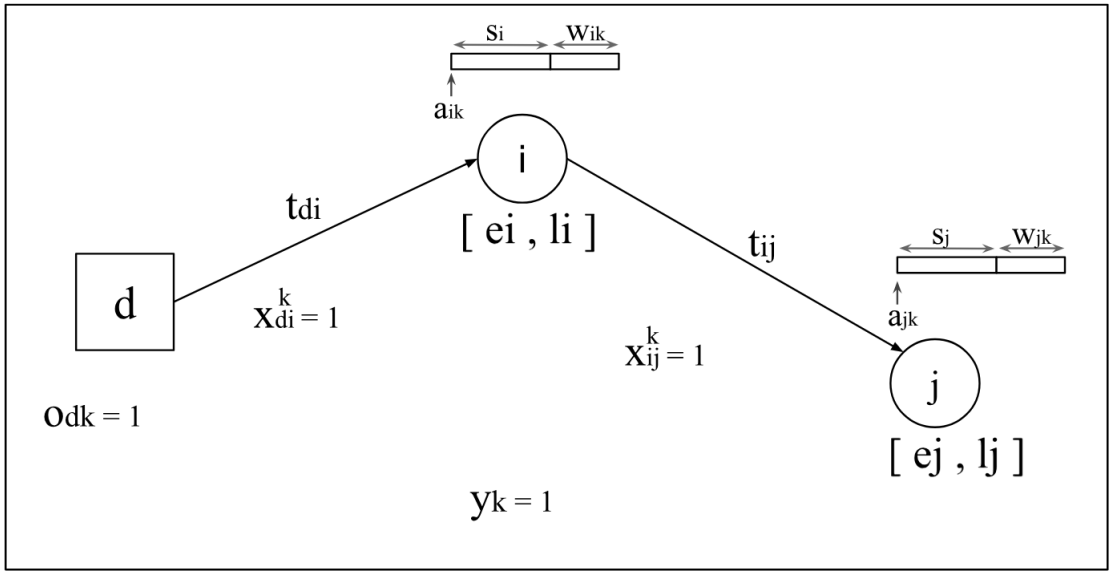

Figure 3.4: Illustration of key notations

Let $\mathcal{K}$ denote the set of OS vehicles. Each vehicle operates at a constant speed v and is equipped with a battery capacity B, expressed as the maximum travelable distance before requiring a recharge. The service region contains a set of depots $\mathcal{D}$, each with a limited parking and recharging capacity denoted by Q, and a set of incidents $\mathcal{N}$ that require intervention. The complete set of nodes in the network is then defined as $\mathcal{V} = \mathcal{D} \cup \mathcal{N}$.

For each pair of nodes $i, j \in \mathcal{V}$, t_{ij} represents the travel time between node i and node j, computed based on vehicle speed and known distances. Each incident $i \in \mathcal{N}$ is characterized by a deterministic service time s_i, corresponding to the duration needed for an OS vehicle to resolve the incident. Additionally, incidents must be addressed within specific time windows denoted by $[e_i, l_i]$, where e_i is the earliest allowable start time and l_i is the latest. Finally, all vehicles operate within a fixed shift duration denoted by S, which imposes a hard constraint on the total duration of travel and service activities that can be performed by a single vehicle during its route.

To model the optimal assignment and routing of OS vehicles, we began by identifying our main operational needs: determining whether a vehicle is used, which depot it is assigned to, how it travels between nodes, and when it arrives at each incident. These requirements naturally led us to define the following decision variables:

- $x_{ij}^k \in \{0,1\}$: indicates whether vehicle k travels directly from node i to node j;
- $y_k \in \{0,1\}$: indicates whether vehicle k is used in the solution;
- $o_{dk} \in \{0,1\}$: indicates whether vehicle k is assigned to depot d;
- $a_{ik} \in \mathbb{R}_+$: represents the arrival time of vehicle k at node i;
- $w_{ik} \in \mathbb{R}_+$: represents the waiting time of vehicle k at node i.

Based on the STM's priority structure, our model is built with two objectives, formulated in a lexicographic bi-objective approach:

1. **Primary objective – minimize the number of vehicles:**

$$\min Z_1 = \sum_{k \in \mathcal{K}} y_k \tag{3.1}$$

   This objective ensures that we only use the minimum number of vehicles necessary to satisfy all interventions.

2. **Secondary objective – minimize the response time:**

   For each incident $i \in \mathcal{N}$, let e_i denote its occurrence time. If vehicle k is assigned to respond to incident i, its *response time* is defined as the time difference $\delta_{ik} = a_{ik} - e_i$. Since e_i is a fixed known value (provided by the incident forecasting model), minimizing δ_{ik} is equivalent to minimizing a_{ik}. Therefore, the second objective can be written as:

$$\min Z_2 = \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{K}} a_{ik} \tag{3.2}$$

   This allows us to ensure that all incidents are handled as quickly as possible, after minimizing the fleet size.

Statistical Independence Analysis Between the two objectives

1. **Hypothesis Formulation**

   To assess whether the two objective functions *Fleet Size* and *Response Time* are statistically dependent, we conducted a correlation analysis. The hypotheses are formulated as follows:

   - **Null Hypothesis ($\mathbf{H_0}$):** There is no statistical correlation between fleet size and response time.
   - **Alternative Hypothesis ($\mathbf{H_1}$):** There is a statistically significant correlation (either positive or negative) between fleet size and response time.

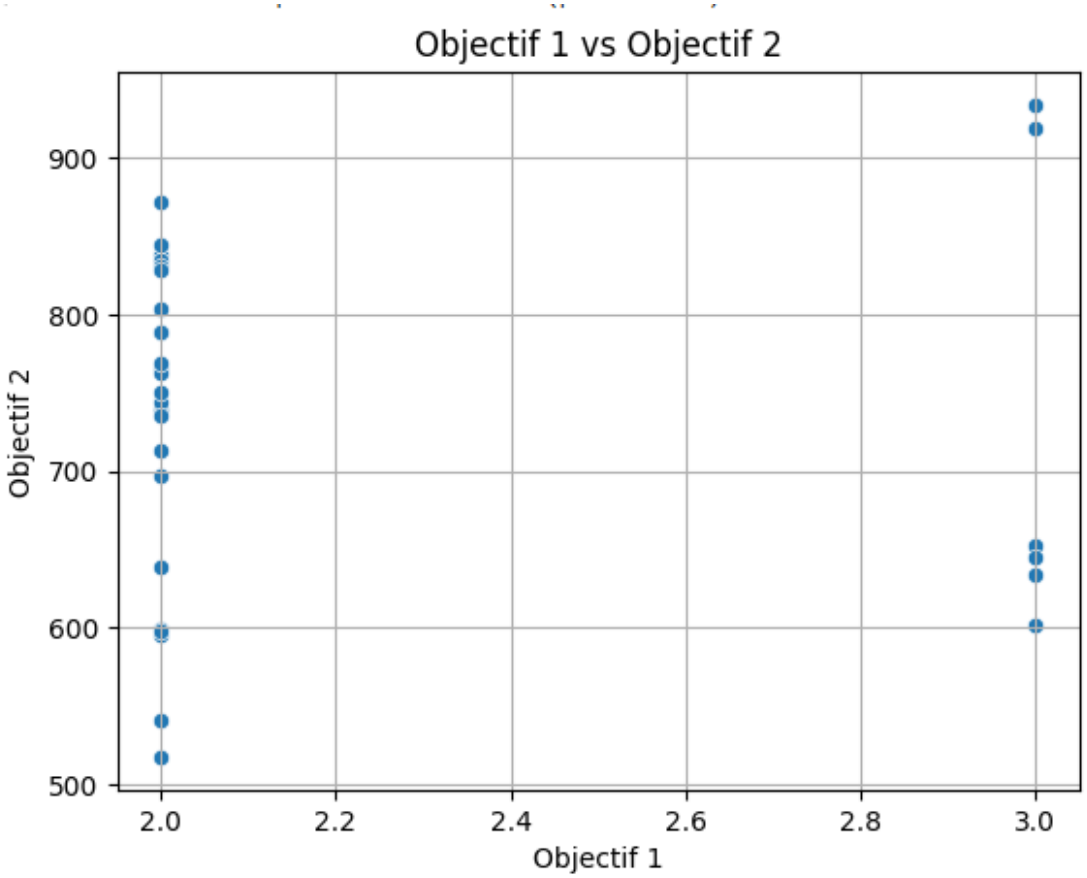

Figure 3.5: Scatter Plot of Fleet Size vs. Response Time

2. **Statistical Tests Used**

   To test the independence between the two objectives, we used two complementary correlation metrics:

   - **Pearson's correlation coefficient (r):** Measures the strength and direction of a *linear relationship* between two continuous variables.
   - **Spearman's rank correlation coefficient (ρ):** A *non-parametric test* that assesses whether the relationship between two variables is *monotonic*, regardless of its linearity.

3. **Results and Interpretation** Based on a sample of **31 solutions** generated by our multi-objective model, we obtained the following results:

Metric	**Correlation**	**p-value**
Pearson correlation (r)	-0.0130	0.9446
Spearman correlation (ρ)	-0.0274	0.8837

   Both correlation coefficients are very close to zero, and their p-values are significantly higher than 0.05, indicating no statistically significant relationship between fleet size and response time.

4. **Conclusion**

   Since both Pearson and Spearman tests failed to reject the null hypothesis, we conclude that *fleet size* and *response time* are statistically independent within the considered solution space. This independence suggests that optimizing one objective does not inherently affect the other in a linear or monotonic manner. Thus, the bi-objective formulation is meaningful and justifies the exploration of trade-offs on the Pareto front.

Model Constraints:

In order to define a mathematically sound and operationally relevant model, we started by brainstorming the essential operational rules and limitations that the system must satisfy. This led us to the following considerations:

- All incidents must be satisfied by exactly one vehicle.
- Each vehicle's route must start and end at its assigned depot.
- Vehicles must not exceed their shift duration.
- Vehicles must not exceed their maximum travelable distance (battery constraint).
- Each vehicle should have a fairly balanced workload to avoid overloading or under-using any vehicle.

We will now formulate mathematical expressions to satisfy these constraints. For clarity and organization, the constraints are grouped into the following categories:

1. **Classical VRP constraints:** To ensure the validity and efficiency of vehicle routes in our model, we start by introducing the classical constraints typically encountered in VRP formulations. These constraints ensure that each incident is served exactly once and that the routes are continuous.

   - **Demand satisfaction (each incident is visited exactly once)** This constraint guarantees that every incident (or customer node) is visited once and only once by one vehicle. In our formulation, we use a binary decision variable x_{ij}^k, which equals 1 if vehicle k travels from node i to node j, and 0 otherwise. As shown in Figure 3.6, we model the network as a directed graph where each node represents an incident, and arcs represent possible travel paths between them.

     To ensure that every incident node is entered exactly once, we impose the following constraint:

$$\sum_{i \in \mathcal{V}, i \neq j} \sum_{k \in \mathcal{K}} x_{ij}^k = 1, \quad \forall j \in \mathcal{N} \tag{3.3}$$

This means that each node must have exactly one incoming arc across all vehicles. Similarly, to ensure that each node is exited exactly once, we add:

$$\sum_{j \in \mathcal{V}, i \neq j} \sum_{k \in \mathcal{K}} x_{ij}^k = 1, \quad \forall i \in \mathcal{N} \tag{3.4}$$

Together, these two constraints guarantee that: no incident is skipped, no incident is served multiple times and each vehicle visits a node only once, respecting service feasibility.

- **Flow conservation (continuity of vehicle paths)** This set of constraints ensures the continuity of vehicle routes. That is, for any node visited by a vehicle, the same vehicle must also leave that node to go to another. This avoids routes where a vehicle arrives at an incident and doesn't continue further.
  Formally, we write:

$$\sum_{i \in \mathcal{V}, i \neq j} x_{ij}^k = \sum_{l \in \mathcal{V}, l \neq j} x_{jl}^k, \quad \forall j \in \mathcal{V}, \forall k \in \mathcal{K} \tag{3.5}$$

This constraint applies individually to each vehicle k and ensures that inflow equals outflow at every node : if a vehicle arrives at a node, it must also leave it, ensuring a valid and complete path for each vehicle.

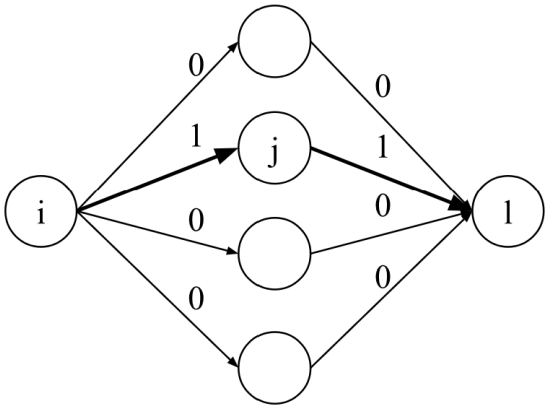

Figure 3.6: Graphical Representation of Node Visit and Flow Balance Constraints

2. **Time-related constraints:** This group of constraints ensures that all vehicles adhere to temporal requirements along their assigned routes, including arrival within time windows and proper chronological sequencing of visited locations.

   - **Respect of the time windows:** Each incident $i \in \mathcal{N}$ is characterized by an occurrence period, defined by an earliest start time e_i and a latest start time l_i. This effectively creates a time window within which an OS vehicle $k \in \mathcal{K}$ must arrive to begin service. If a vehicle k serves incident i (indicated by $\sum_{j \in \mathcal{V}, i \neq j} x_{ji}^k = 1$), its arrival time a_{ik} at incident i must fall within this specified interval. This is formally expressed as:

$$e_i \cdot \sum_{j \in \mathcal{V}, i \neq j} x_{ji}^k \leq a_{ik} \leq l_i \cdot \sum_{j \in \mathcal{V}, i \neq j} x_{ji}^k, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{3.6}$$

The summation term ensures these bounds are only active if incident i is indeed visited by vehicle k.

- **Chronological Routing Phases:**To accurately model the progression of a vehicle along its route, we must establish the chronological relationship between arrival times at successive nodes. This involves considering the travel time between nodes, service time at an incident (if applicable), and any potential waiting time incurred by the vehicle. We can conceptualize a vehicle's route in segments, as illustrated in Figure 3.7:

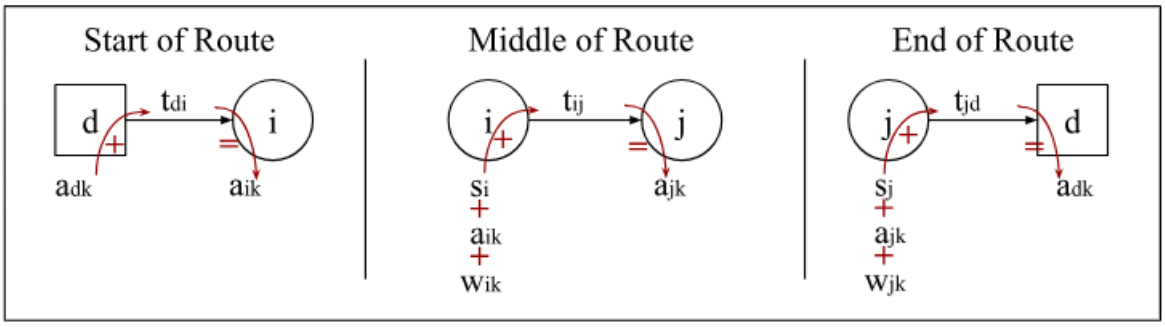

Figure 3.7: Graphical Representation of Chronological Routing Phases

To enforce these chronological relationships conditionally based on whether an arc is actually traversed by a vehicle (i.e., $x_{ij}^k = 1$), one might intuitively think of a direct multiplication. For example, if vehicle k travels from depot d to incident i ($x_{di}^k = 1$), its arrival time a_{ik} would be $t_{di} + w_{dk}$. If it doesn't make this trip ($x_{di}^k = 0$), then a_{ik} related to this specific path initiation should effectively be zero or unconstrained by this particular relationship. This could be represented non-linearly as:

$$a_{ik} = (t_{di} + w_{dk}) \cdot x_{di}^k, \quad \forall i \in \mathcal{N}, \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.7}$$

Similarly, for travel between an incident i and a subsequent node j:

$$a_{jk} = (a_{ik} + s_i + w_{ik} + t_{ij}) \cdot x_{ij}^k, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{V}, i \neq j, \forall k \in \mathcal{K} \tag{3.8}$$

However, this formulations involve the multiplication of a decision variable by a binary variable (x_{di}^k or x_{ij}^k), which results in a non-linear constraint. Since we are developing a Mixed-Integer **Linear** Program (MILP), these non-linearities must be transformed into equivalent linear forms.

A standard technique in mathematical programming to linearize such products is **the "big-M" method**. This method uses a sufficiently large constant M to activate or deactivate parts of a constraint based on the value of the binary variable. The value of M must be chosen carefully: it needs to be large enough so that it doesn't incorrectly restrict feasible solutions when the binary variable deactivates the core relationship, yet not so large as to cause numerical instability in the solver. In our context, a suitable value for M could be the maximum shift duration (e.g., 6 hours or 360 minutes), as no single activity or arrival time should logically exceed this overall operational limit.

Thus, applying the big-M linearization to the *start of the route* (from a depot d to an incident i by vehicle k), the non-linear relationship is replaced by the

following pair of linear inequalities:

$$a_{ik} \leq t_{di} + w_{dk} + M \cdot (1 - x_{di}^k), \qquad \forall i \in \mathcal{N}, \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.9}$$

$$a_{ik} \geq t_{di} + w_{dk} - M \cdot (1 - x_{di}^k), \qquad \forall i \in \mathcal{N}, \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.10}$$

These constraints effectively enforce $a_{ik} \approx t_{di} + w_{dk}$ if vehicle k travels from depot d to incident i (i.e., $x_{di}^k = 1$). In this case, $1 - x_{di}^k = 0$, and the M terms vanish. If $x_{di}^k = 0$, then $1 - x_{di}^k = 1$, and the constraints become $a_{ik} \leq t_{di} + w_{dk} + M$ and $a_{ik} \geq t_{di} + w_{dk} - M$. Given that M is large, these effectively become $a_{ik} \leq$ large value, and $a_{ik} \geq$ small or negative value, rendering them non-restrictive for a_{ik} in this specific path context (other constraints, like time windows, will bind a_{ik}). Note that if depots have fixed departure times, w_{dk} might be zero, or it could represent waiting until the earliest service time e_i of the incident. The arrival time a_{ik} is also bounded by the incident's time window constraints.

Similarly, for the *sequencing between nodes* (from an incident i to any subsequent node j, which could be another incident or a depot, by vehicle k), the non-linear relationship is linearized as:

$$a_{jk} \leq a_{ik} + s_i + w_{ik} + t_{ij} + M \cdot (1 - x_{ij}^k), \qquad \forall i \in \mathcal{N}, \forall j \in \mathcal{V}, i \neq j, \forall k \in \mathcal{K} \tag{3.11}$$

$$a_{jk} \geq a_{ik} + s_i + w_{ik} + t_{ij} - M \cdot (1 - x_{ij}^k), \qquad \forall i \in \mathcal{N}, \forall j \in \mathcal{V}, i \neq j, \forall k \in \mathcal{K} \tag{3.12}$$

These constraints enforce $a_{jk} \approx a_{ik} + s_i + w_{ik} + t_{ij}$ if vehicle k travels from incident i to node j (i.e., $x_{ij}^k = 1$). The decision variables for waiting times (w_{dk} and w_{ik}) and arrival times (a_{ik}, a_{jk}) are determined by the model during optimization to ensure feasibility and contribute to the overall objective function (e.g., minimizing total response time or operational costs). The use of inequalities (upper and lower bounds with big-M) instead of attempting to force a strict equality when an arc is active is a robust modeling practice, particularly when dealing with decision variables like waiting times that provide flexibility for the optimizer. The "End of Route and Return to Depot" phase is implicitly covered by these sequencing constraints when node j is a depot.

3. **Depot assignment constraints:** These constraints govern how vehicles are assigned to depots and how their routes must originate and terminate, ensuring logical fleet operations and adherence to depot assignments.

   - **Unique Depot Assignment per Vehicle:** A fundamental requirement is that if a vehicle is utilized, it must be assigned to exactly one home depot from which it operates for its entire shift. This ensures clear accountability and operational structure. This is enforced by ensuring that each vehicle k is assigned to at most one depot, or exactly one if it is used. If a vehicle is not used, it will not be assigned to any depot.

$$\sum_{d \in \mathcal{D}} o_{dk} = y_k, \quad \forall k \in \mathcal{K} \tag{3.13}$$

- **Consistency between Depot Assignment and Route Origination:** If a vehicle k starts its route by traveling from a depot d to any incident $j \in \mathcal{N}$ (i.e., $\sum_{j \in \mathcal{N}} x_{dj}^k = 1$), then that vehicle k must indeed be assigned to operate out of that specific depot d (i.e., $o_{dk} = 1$). This links the operational routing decision (leaving a depot) with the strategic depot assignment.

$$o_{dk} \geq \sum_{j \in \mathcal{N}} x_{dj}^k, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.14}$$

  This inequality ensures that if the sum on the right is 1 (meaning vehicle k leaves depot d for an incident), then o_{dk} must be at least 1 (and thus 1, since it's binary). If the vehicle does not leave depot d for an incident, the sum is 0, and o_{dk} can be 0 or 1 (its value being determined by the firstprevious constraint.

- **Route Start and End at the Same Assigned Depot:** A key operational rule is that each vehicle's tour must be a closed loop, starting and ending at its assigned home depot. This means if a vehicle k departs from a depot d (i.e., $\sum_{j \in \mathcal{V}} x_{dj}^k = 1$, indicating it leaves depot d for any node j), it must also return to that same depot d (i.e., $\sum_{i \in \mathcal{V}} x_{id}^k = 1$, indicating it arrives at depot d from any node i). Conversely, if it doesn't start from depot d, it cannot end there. This is enforced by:

$$\sum_{i \in \mathcal{V}} x_{id}^k \leq \sum_{j \in \mathcal{V}} x_{dj}^k, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.15}$$

- **At Most One Departure and Arrival per Depot per Vehicle:** To further ensure that a vehicle's route is well-defined and associated with a single depot, we must explicitly state that a vehicle k can depart from at most one depot, and arrive at at most one depot. If the vehicle is not used, it will not depart from or arrive at any depot.

$$\sum_{i \in \mathcal{V}} x_{id}^k \leq 1, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.16}$$

$$\sum_{j \in \mathcal{V}} x_{dj}^k \leq 1, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.17}$$

4. **Capacity Constraints:** These constraints impose limits on various operational aspects, including vehicle operational duration (shift length), travel range (battery capacity for EVs), and depot capacity for hosting vehicles.

   - **Shift Duration Constraint:** Each OS vehicle $k \in \mathcal{K}$ operates within a maximum allowed shift duration, denoted by S (e.g., 360 minutes for a 6-hour shift). To ensure that no vehicle exceeds its shift limit, we must constrain the timing of all its activities. A straightforward way to enforce this linearly is to ensure that the arrival time a_{ik} of vehicle k at any node $i \in \mathcal{V}$ (which includes incidents and its return to the depot) does not exceed the maximum shift duration S. If a vehicle returns to its depot, that return arrival time must be within S. If it serves an incident, the arrival at that incident (and implicitly the subsequent service and travel) must allow for completion within S. This is expressed as:

$$a_{ik} \leq S, \quad \forall i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{3.18}$$

- **Vehicle Range Constraint (Battery Capacity for EVs):** For electric vehicles, a critical constraint is their limited driving range, often defined by battery capacity, which can be translated into a maximum travel distance B (e.g., 150 km). To ensure that the total distance traveled by each EV $k \in \mathcal{K}$ on its route does not exceed this limit, we sum the distances of all traversed segments. Assuming a constant average vehicle speed v, the travel time t_{ij} between nodes i and j can be related to distance d_{ij} by $d_{ij} = v \cdot t_{ij}$. Therefore, the constraint on total travel distance can be expressed using the travel times t_{ij} and the decision variables x_{ij}^k:

$$v \cdot \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} x_{ij}^k \cdot t_{ij} \leq B, \quad \forall k \in \mathcal{K} \tag{3.19}$$

  This constraint sums the travel times t_{ij} for all arcs (i, j) that are actually part of vehicle k's route (where $x_{ij}^k = 1$). Multiplying this total travel time by the average speed v_{speed} gives the total distance covered by vehicle k, which must not exceed the maximum battery range B_{max}.

- **Depot Capacity Constraint:** As depots are often multi-purpose facilities, they may have a limited capacity Q for hosting or basing OS service vehicles. This means that the total number of OS vehicles assigned to operate out of a specific depot $d \in \mathcal{D}$ cannot exceed its designated capacity. Using the binary variable o_{dk}, this constraint is formulated as:

$$\sum_{k \in \mathcal{K}} o_{dk} \leq Q, \quad \forall d \in \mathcal{D} \tag{3.20}$$

  This ensures that for each depot d, the sum of vehicles k assigned to it does not surpass its capacity Q.

5. **Load balancing constraints:** While the primary objectives often revolve around minimizing fleet size and response times, organizations may also consider aspects of equity and workload distribution among their employees for socio-economic reasons or to maintain operational sustainability. Although not the primary focus for the STM at this juncture for this specific OS vehicle problem, we explored how load balancing could be incorporated as a constraint. The aim would be to ensure that all deployed OS personnel (represented by their vehicles) handle a relatively similar amount of work.

   Workload for an OS vehicle $k \in \mathcal{K}$ can be conceptualized in at least two ways:

   a. Based on the number of interventions: Ensuring that each active OS vehicle handles a roughly equal number of incidents during its shift.

   b. Based on the total work duration or travel time: Ensuring that the total time spent actively working (traveling to incidents, servicing incidents) is distributed equitably among active OS vehicles.

   The two workload balancing approaches differ in focus and complexity. Balancing by number of interventions ensures a simple, equal task count among personnel but ignores variation in effort or time per task. Balancing by total duration or travel time offers a more accurate reflection of actual workload by incorporating service

and travel times, but it requires more detailed data. The choice depends on data availability, equity objectives, and desired model complexity.

An initial approach to formulate load balancing, for example, based on total travel time, might involve ensuring that the total travel time for each active vehicle k ($\sum_{i\in\mathcal{V}}\sum_{\substack{j\in\mathcal{V}\\ i\neq j}} t_{ij}\cdot x_{ij}^k$) stays within a certain percentage (e.g., $\pm 20\%$) of the average travel time per active vehicle. This could be expressed as:

$$(1-\alpha)\cdot\left(\frac{\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}}\sum_{\substack{j\in\mathcal{V}\\ i\neq j}} t_{ij}\cdot x_{ij}^k}{\sum_{k\in\mathcal{K}} y_k}\right) \leq \sum_{i\in\mathcal{V}}\sum_{\substack{j\in\mathcal{V}\\ i\neq j}} t_{ij}\cdot x_{ij}^k \leq (1+\alpha)\cdot\left(\frac{\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}}\sum_{\substack{j\in\mathcal{V}\\ i\neq j}} t_{ij}\cdot x_{ij}^k}{\sum_{k\in\mathcal{K}} y_k}\right) \quad \forall k\in\mathcal{K} \tag{3.21}$$

where $y_k = 1$ if vehicle k is used, and α is the allowable deviation (e.g., 0.2 for 20%). A similar formulation could be conceptualized for balancing the number of incidents:

$$(1-\alpha)\cdot\left(\frac{\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}}\sum_{\substack{j\in\mathcal{V}\\ i\neq j}} x_{ij}^k}{\sum_{k\in\mathcal{K}} y_k}\right) \leq \sum_{i\in\mathcal{V}}\sum_{\substack{j\in\mathcal{V}\\ i\neq j}} x_{ij}^k \leq (1+\alpha)\cdot\left(\frac{\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}}\sum_{\substack{j\in\mathcal{V}\\ i\neq j}} x_{ij}^k}{\sum_{k\in\mathcal{K}} y_k}\right) \quad \forall k\in\mathcal{K} \tag{3.22}$$

However, the formulation above involving division by a sum of decision variables ($\sum_{k\in\mathcal{K}} y_k$, the total number of active vehicles) introduces non-linearity, making it unsuitable for a standard MILP solver.

To linearize the concept of load balancing, a common strategy is to compare the workload of every pair of active vehicles. Let W_k represent the workload of vehicle k (this could be its total travel time, number of incidents served, or total service duration). We want to ensure that for any two active vehicles k and k', the absolute difference in their workloads, $|W_k - W_{k'}|$, does not exceed a predefined allowable difference, Δ_{max}. This can be linearized using the big-M method to activate the comparison only when both vehicles k and k' are in service (i.e., $y_k = 1$ and $y_{k'} = 1$). The constraints would be formulated as:

$$W_k - W_{k'} \leq \Delta_{\max} + M(1-y_k) + M(1-y_{k'}) \tag{3.23}$$
$$W_{k'} - W_k \leq \Delta_{\max} + M(1-y_k) + M(1-y_{k'}) \tag{3.24}$$

for all pairs k, k' where $k < k'$, and W_k would be substituted by the linear expression for workload, e.g., $\sum_i\sum_j t_{ij}x_{ij}^k$ for travel time, or $\sum_i\sum_j x_{ij}^k$ (if j is an incident) for number of incidents.)

These pairwise constraints ensure that no two active vehicles have workloads that differ by more than Δ_{max}. While this increases the number of constraints, it maintains the linearity required for MILP solvers. The specific definition of W_k (as number of incidents or total time) would then be substituted into these linearized constraints.

6. **Variable Definition and Domain Constraints:** A Mixed-Integer Linear Program involves different types of decision variables, each with a specific domain. It is crucial to define these domains correctly. In our model, we utilize binary variables

to represent yes/no decisions, and continuous (real-valued) variables to represent quantities such as time.

For instance, binary variables like x_{ij}^k indicate whether vehicle k travels directly from node i to node j. Similarly, o_{dk} is a binary variable indicating if vehicle k is assigned to depot d. Another important binary variable, let's denote it as y_k, could be defined to be 1 if and only if vehicle k is actually used (i.e., assigned to serve at least one incident or makes at least one trip from a depot). This y_k variable is often linked to other decision variables; for example, if $\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{V}} x_{ij}^k > 0$, then y_k must be 1.

Continuous variables in our model primarily relate to timing aspects. For example, a_{ik} represents the arrival time of vehicle k at node i, and w_{ik} represents the waiting time of vehicle k at node i. These can take any non-negative real value, determined by the optimization process.

The domains for the primary decision variables used in this model are therefore defined as follows:

$$x_{ij}^k, y_k, o_{dk} \in \{0, 1\}, \qquad \forall i, j \in \mathcal{V}, i \neq j, \forall k \in \mathcal{K}, \forall d \in \mathcal{D} \tag{3.25}$$

$$a_{ik}, w_{ik} \geq 0 \text{ (and } \in \mathbb{R}), \qquad \forall i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{3.26}$$

The binary variables enforce discrete choices, while the real-valued variables allow for continuous adjustments in timing, all within the constraints of the overall system.

3.2.2 The mathematical model

Based on the previously defined notations, decision variables, objectives, and constraints, we now present the complete MILP formulation of the problem. This model integrates all operational considerations, including vehicle routing, time windows, and resource limitations.

We begin this section with a summary table of the main notations used throughout the model, followed by the complete mathematical formulation.

It is important to note that, although workload balancing was initially considered among the relevant constraints, we decided not to include it in the final version of the model. This is due to the additional complexity it introduces. In fact, even without this constraint, we observed significant computational challenges in solving large-scale instances.

The problem can be described using the following notations:

Table 3.1: Notation used in the MILP model

Sets and indices:	
$\mathcal{N} = \{1, 2, \ldots, n\}$	Set of incidents.
$\mathcal{D} = \{n+1, \ldots, n+d\}$	Set of depots, indexed by d.
$\mathcal{V} = \mathcal{N} \cup \mathcal{D}$	Set of all nodes.
$\mathcal{K}$	Set of vehicles, indexed by k.
ni**Parameters:**	
t_{ij}	Traveling time between two nodes i and j.
$[e_i, l_i]$	Time window for each incident i.
s_i	Service time for incident i.
v	Vehicle speed.
B	Vehicle battery capacity.
S	Vehicle shift.
Q	Depot capacity.
M	Positive large number.
Variables:	
x_{ij}^k	1 if vehicle k travels from i to j, 0 otherwise.
y_k	1 if vehicle k is used, 0 otherwise.
o_{dk}	1 if vehicle k is assigned to the depot d, 0 otherwise.
a_{ik}	Arrival time of the vehicle k at i.
w_{ik}	Waiting time for vehicle k at node i.

Based on these elements and the previously discussed modeling choices, the optimization problem is formulated as follows:

$$\min Z_1 = \sum_{k \in \mathcal{K}} y_k \tag{3.27}$$

$$\min Z_2 = \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{K}} a_{ik} \tag{3.28}$$

Subject to:

$$\sum_{i \in \mathcal{V}, i \neq j} \sum_{k \in \mathcal{K}} x_{ij}^k = 1, \quad \forall j \in \mathcal{N} \tag{3.29}$$

$$\sum_{j \in \mathcal{V}, i \neq j} \sum_{k \in \mathcal{K}} x_{ij}^k = 1, \quad \forall i \in \mathcal{N} \tag{3.30}$$

$$\sum_{i\in\mathcal{V},i\neq j} x_{ij}^k = \sum_{i\in\mathcal{V},i\neq j} x_{ji}^k, \quad \forall j \in \mathcal{V}, \forall k \in \mathcal{K} \tag{3.31}$$

$$e_i \cdot \sum_{j\in\mathcal{V},i\neq j} x_{ji}^k \leq a_{ik} \leq l_i \cdot \sum_{j\in\mathcal{V},i\neq j} x_{ji}^k, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{3.32}$$

$$a_{ik} \leq t_{di} + w_{dk} + M \cdot (1 - x_{di}^k), \quad \forall i \in \mathcal{N}, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.33}$$

$$a_{ik} \geq t_{di} + w_{dk} - M \cdot (1 - x_{di}^k), \quad \forall i \in \mathcal{N}, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.34}$$

$$a_{jk} \leq a_{ik} + s_i + w_{ik} + t_{ij} + M \cdot (1 - x_{ij}^k), \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{V}, i \neq j, \forall k \in \mathcal{K} \tag{3.35}$$

$$a_{jk} \geq a_{ik} + s_i + w_{ik} + t_{ij} - M \cdot (1 - x_{ij}^k), \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{V}, \quad i \neq j, \quad \forall k \in \mathcal{K} \tag{3.36}$$

$$a_{ik} \leq S, \quad \forall i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{3.37}$$

$$v \cdot \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{V}} x_{ij}^k \cdot t_{ij} \leq B, \quad \forall k \in \mathcal{K} \tag{3.38}$$

$$\sum_{d\in\mathcal{D}} o_{dk} = 1, \quad \forall k \in \mathcal{K} \tag{3.39}$$

$$o_{dk} \geq \sum_{j\in\mathcal{N}} x_{dj}^k, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.40}$$

$$\sum_{i\in\mathcal{V}} x_{id}^k \leq \sum_{j\in\mathcal{V}} x_{dj}^k, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.41}$$

$$\sum_{i\in\mathcal{V}} x_{id}^k \leq 1, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.42}$$

$$\sum_{j\in\mathcal{V}} x_{dj}^k \leq 1, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \tag{3.43}$$

$$M \cdot y_k \geq \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{V}} x_{ij}^k, \quad \forall k \in \mathcal{K} \tag{3.44}$$

$$\sum_{k\in\mathcal{K}} o_{dk} \leq Q, \quad \forall d \in \mathcal{D} \tag{3.45}$$

$$x_{ij}^k, y_k, o_{dk}, \in \{0, 1\}, \quad \forall i, j \in \mathcal{V}, \forall k \in \mathcal{K}, \forall d \in \mathcal{D} \tag{3.46}$$

$$a_{ik}, w_{ik}, \in \mathbb{R}, \quad \forall i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{3.47}$$

Where (3.27–3.28) are the two objective functions, representing the minimization of fleet size and total response time, respectively. Constraints (3.29-3.30) ensure that each incident is served by exactly one vehicle, thus ensuring full coverage of all incidents. Constraints (3.31) guarantee flow conservation, which means that every vehicle that arrives at an incident node will leave it. Time-related constraints, including time windows, waiting time, and service time, are defined in constraints (3.32-3.36). Constraints (3.37) ensure that no vehicle is assigned to an incident occurring after the end of its designated shift. Constraints (3.38) impose battery capacity limitations on each vehicle. Constraints (3.39-3.43) require that each vehicle is assigned to a specific depot and that each route both starts and ends at the same depot. Constraints (3.44) define the variable y_k, indicating whether vehicle k is used. Finally, constraints (3.45) impose the capacity limits of the depots. while constraints(3.46-3.47) represent the integrality constraints.

3.2.3 Model Functioning Example

To illustrate the conceptual functioning of the optimization approach, which can be seen as a form of lexicographic optimization where objectives are prioritized, let us consider the example depicted in Figure 3.8. This example involves a network with 10 nodes: 7 incidents requiring service (represented by circles and labeled 0 through 6) and 3 potential depots from which vehicles can operate (represented by squares and labeled 7, 8, and 9).

The optimization process can be envisioned in two main steps, prioritizing different objectives sequentially:

1. **Step 1 – Minimizing Fleet Size (Objective 1)** In the first step, the primary goal is to determine the minimum number of OS vehicles (and consequently, active depots) required to service all incidents while satisfying all fundamental operational constraints (such as shift durations, vehicle capacities, and ensuring every incident is attended). At this stage, the model does not primarily focus on minimizing the response time to each individual incident but rather on achieving coverage with the leanest possible fleet.

   The left panel of Figure 3.8 (Objective 1) illustrates a potential outcome of this first optimization phase.

   - Depot 7 is active and serves incidents 0, 1, 3, 5 and 6.
   - Depot 8 is active and serves incident 2 and 4.
   - Depot 9 is not active.

   In this configuration, Just two depots are utilized, implying that a minimum of two vehicles (assuming one route per active depot for simplicity in this example) are necessary to cover all seven incidents. because of the conditions and time constraints nd the data of the time window two vehicles are not suffitient to satify all the demand respecting the model constraints, that why we need more vehicles which lead us to

the optimal solution found by the model 4 vehicles, every vehicle got a route and order of incidents to visit.

2. **Step 2 – Minimizing Response Time (Objective 2)** In the second step, the optimal number of vehicles found in the first step (in our example, 4 vehicles) is now taken as a fixed constraint or an upper bound. The model's objective then shifts to minimizing a measure of total response time (e.g., the sum of arrival times at all incidents or total travel time), given that no more than 4 vehicles can be used.

   The right panel of Figure 3.8 (Objective 2) shows how the routes and depot assignments might be reorganized to achieve this second objective.

   - Depot 7 is now shown as inactive.
   - Depot 8 becomes more central, now serving incidents 1, 2, 3, 4, and 5 with more compact routes.
   - Depot 9 now serves incidents 0 and 6, also with seemingly more direct paths.

Visually, the routes in "Objective 2" appear shorter and more efficient in terms of travel to individual incidents. This reorganization is achieved while adhering to the constraint that the number of vehicles used does not exceed the optimal number determined in Step 1. In this visual example, it appears that the 4 vehicles (or their workload equivalent) might have been re-assigned to operate more efficiently from just 2 depots (8 and 9) to reduce overall response times. The crucial aspect is that the primary objective of minimizing vehicles (to 4 in this example) was satisfied first, and then, within that constraint, response times were improved.

This two-step process effectively emulates a lexicographic approach where fleet size is the highest priority objective, and response time is the secondary priority objective, optimized only after the first objective's optimal value has been achieved and fixed.

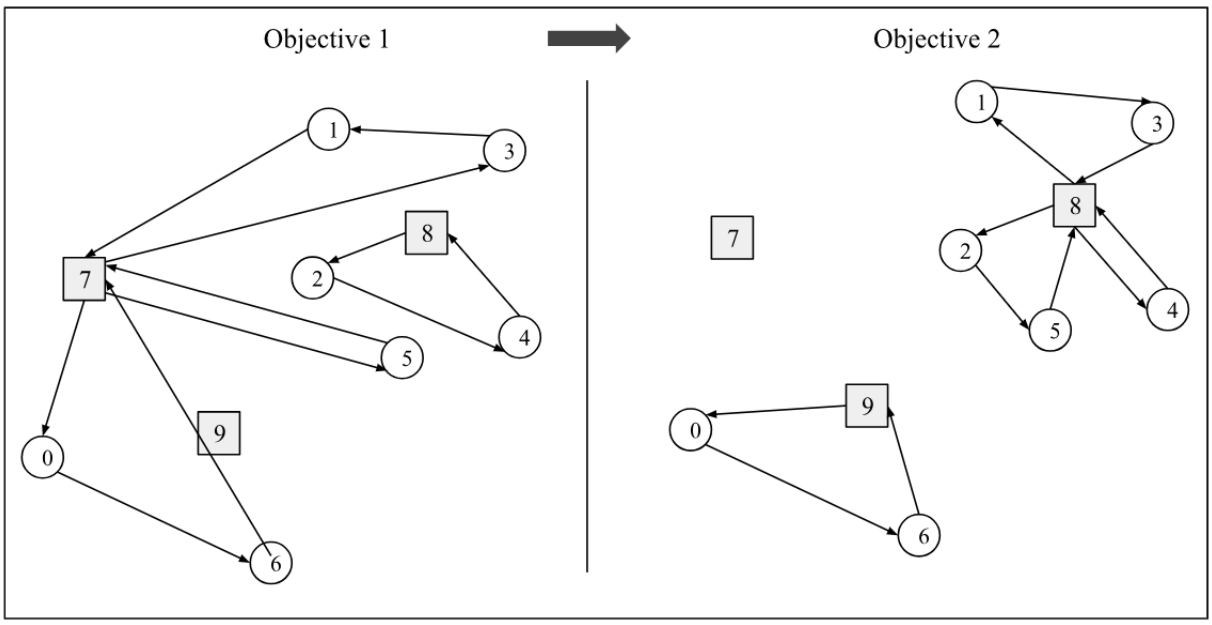

Figure 3.8: Illustration of the Model's Operation

3.3 Heuristic Algorithm: A Decomposition-Based Approach

While the MILP model provides a rigorous framework for finding optimal solutions, its computational demands can become significant for very large-scale instances, such as those representing multiple days of STM operations or highly granular incident data. To address these potential scalability challenges and provide a more agile solution approach, we propose a heuristic algorithm based on problem decomposition. The STM problem exhibits natural decomposability due to its inherent temporal (shifts) and spatial (depots, sectors) structure, offering a promising strategy to tackle larger problem instances efficiently.

3.3.1 Multi-Level Problem Decomposition

The proposed decomposition heuristic operates on three hierarchical levels:

Level 1 : Decomposition by Shift The first level of decomposition addresses the temporal dimension by dividing the overall planning horizon into distinct operational shifts (e.g., 6-hour or 8-hour periods). Historical data indicates that a typical day for the STM involves approximately 200 incidents requiring OS vehicle intervention, distributed across these shifts, with incident density varying between peak and off-peak periods. By considering each shift independently, we significantly reduce the number of concurrent incidents and decision variables that need to be handled at once. For each shift:

- Isolate all incidents occurring within that specific shift's timeframe.
- If the total number of incidents within a single shift is still large (e.g., exceeds a predefined threshold, say 10-15 incidents, which is a common manageable size for exact solvers on subproblems), further decomposition is triggered.

Level 2 : Decomposition by Depot (or Depot Service Area) If a single shift still contains a large number of incidents, the second level of decomposition leverages the spatial distribution of depots. The STM operates 8 depots, and company policy often dictates which depots are primarily responsible for servicing incidents within certain geographical areas or sectors. For a given shift:

- Incidents are provisionally assigned or associated with their closest or designated primary service depot.
- If the number of incidents associated with a single depot within that shift still exceeds the manageable threshold (e.g., 10-15 incidents), a finer-grained spatial decomposition is applied.

Level 3 : Decomposition by Sector The Island of Montreal is partitioned into 15 distinct operational sectors. If, after shift and depot-level considerations, the incident load for a particular depot (within a shift) remains too high, this third level further

divides the problem. For the subset of incidents associated with an overloaded depot during a specific shift:

- These incidents are further grouped based on the sector in which they occur.
- This aims to create smaller, more localized subproblems, each ideally containing a manageable number of incidents (e.g., targeting approximately 10 incidents per subproblem, as suggested by preliminary findings).

In scenarios where, even after the three-level decomposition (by shift, depot, and sector), a particular subproblem still includes an excessive number of incidents (e.g., during high-demand peak periods), additional refinement strategies can be employed to maintain computational tractability. These include:

- **Intra-sector clustering**: Incidents within an overly dense sector-depot-shift subproblem can be further partitioned into smaller spatial clusters based on geographic proximity. This produces more granular subproblems, each with fewer incidents, allowing for more efficient resolution.
- **Demand aggregation**: Incidents that are both temporally and spatially close can be temporarily aggregated into a single *meta-incident*. The aggregated service time corresponds to the sum of the individual service times, and the location is approximated by a centroid or a representative point. After solving, the meta-incident is disaggregated back into the original incidents to preserve demand fidelity.
- **Time-window segmentation**: Incidents within a subproblem are divided into smaller subsets according to predefined or adaptive time intervals. These time-segmented subproblems are then solved sequentially or in parallel, improving MILP tractability while respecting temporal demand dynamics.

The overarching goal of these techniques is to ensure that all subproblems remain within a manageable size threshold, thereby improving the efficiency and robustness of the decomposition heuristic under varying operational intensities. Future work will address the refinement of the decomposition logic, subproblem definitions (particularly vehicle sharing), and the integration of these advanced strategies.

Algorithm 2: Three-Level Decomposition Heuristic

Input: Set of incidents $\mathcal{N}$, partitioned by days into q shifts $\mathcal{T} = \{T_1, T_2, \ldots, T_q\}$. Each shift is associated with depots $\mathcal{D} = \{D_1, D_2, \ldots, D_8\}$, and each depot covers a subset of sectors $\mathcal{S} = \{S_1, S_2, \ldots, S_{13}\}$.

For each shift $T \in \mathcal{T}$:
  If $|\mathcal{N}_T| \leq 10$:
    Solve MILP on $\mathcal{N}_T$ directly
  Else
    For each depot $D \in \mathcal{D}$ associated with shift T:
      If $|\mathcal{N}_{T,D}| \leq 10$:
        Solve MILP on $\mathcal{N}_{T,D}$
      Else
        For each sector $S \in \mathcal{S}_D$:
          If $|\mathcal{N}_{T,D,S}| \leq 10$:
            Solve MILP on $\mathcal{N}_{T,D,S}$
          Else
            Apply a **refinement strategy** on $\mathcal{N}_{T,D,S}$
            Solve the resulting MILP on the refined subproblem
          End if
        End for
      End if
    End for
End if
End for

Aggregate all local MILP solutions to build the global solution (fleet size, response time, etc.)

Output: Aggregated operational plan combining all subproblem solutions

3.3.2 Solution Consolidation

Once the problem is decomposed into these smaller, more manageable subproblems (e.g., incidents within a specific sector, served by a specific depot, during a specific shift), the core MILP formulation (as described in Section 3.2) can be applied to solve each subproblem independently. The objective within each subproblem would be to optimize fleet usage and response times for the incidents contained within it, using vehicles notionally assigned to the relevant depot.

After solving all subproblems, a post-processing and aggregation step is required. This involves:

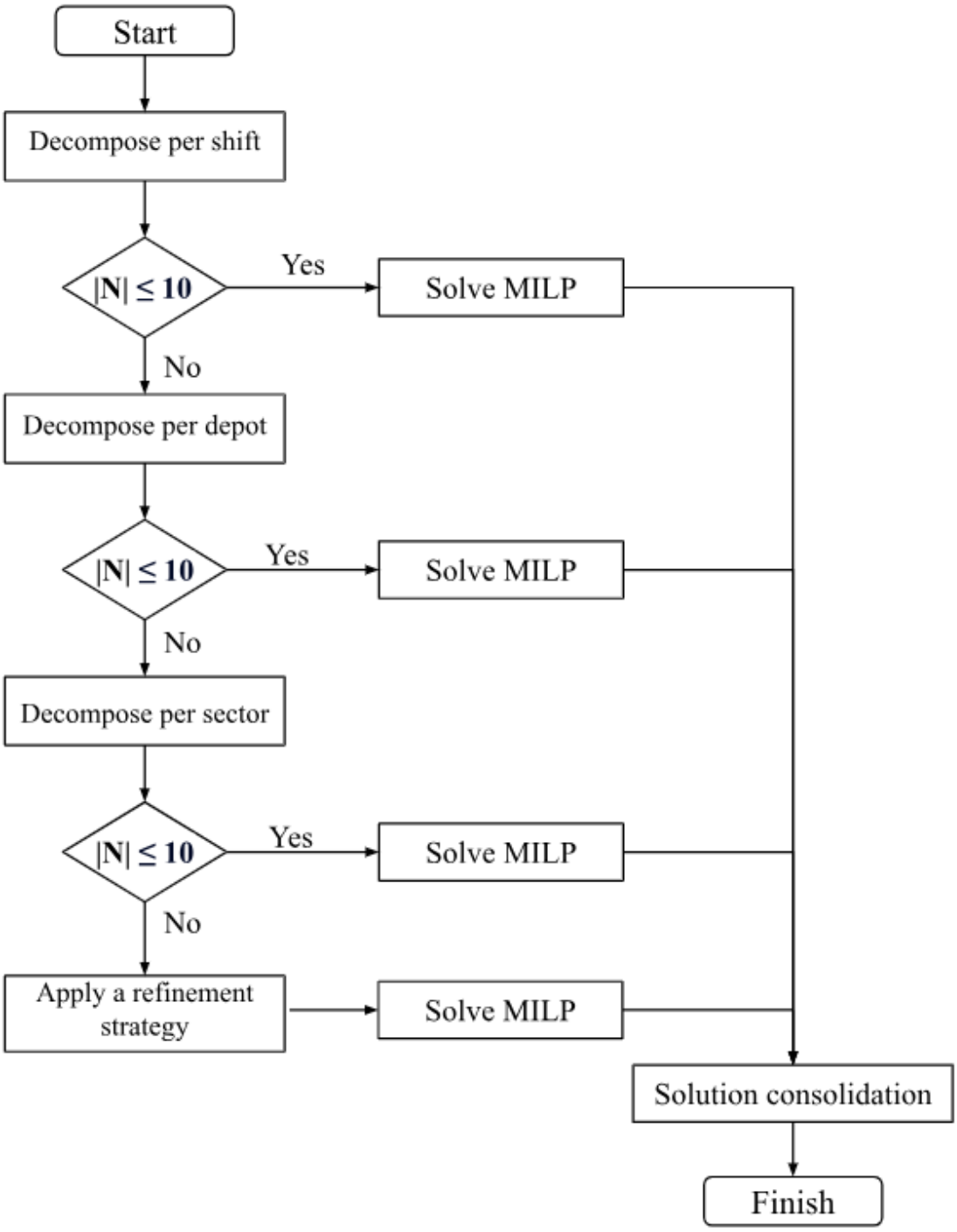

Figure 3.9: Structure of the Heuristic Decomposition Algorithm

1. **Consolidating Vehicle Requirements:** Summing the number of vehicles required by each subproblem can provide an initial estimate of the total fleet size. However, careful consideration is needed as a vehicle might potentially serve incidents across subproblem boundaries if shifts or geographical areas are very close, or if a more sophisticated global aggregation is performed. For this heuristic, a simpler approach might be to sum depot-level vehicle needs per shift.
2. **Aggregating Performance Metrics:** Response times, travel distances, and other relevant metrics from each subproblem are collected and aggregated to estimate the overall system performance.
3. **Potential Refinements (Future Work):** More advanced aggregation might involve re-optimizing vehicle assignments at the boundaries of decomposed units or using the heuristic solution as a high-quality starting point for a global MILP solve with a very short time limit.

The general logic of the decomposition-based technique is illustrated in Figure 3.9.

3.4 Conclusion

This chapter has comprehensively detailed the formulation of the mathematical optimization model designed to address STM's challenge, alongside the development of a complementary heuristic approach. We began by establishing a conceptual model of the operational process, which was then systematically translated into a rigorous Mixed-Integer Linear Program (MILP). This involved defining the necessary sets, parameters, decision variables, the objective function aimed at minimizing fleet size and operational response times, and a detailed set of constraints capturing the intricate operational rules. An illustrative example was also provided to demonstrate the fundamental mechanics of the proposed mathematical model.

Recognizing the potential computational challenges associated with solving large-scale instances of this complex problem with an exact MILP, we also proposed and detailed a multi-level decomposition-based heuristic. This heuristic leverages the natural temporal and spatial structure of the STM's operations to break down the overarching problem into smaller, more manageable subproblems. The methodology for this multi-level problem decomposition and the subsequent consolidation of solutions from these subproblems were outlined.

With these models now formally defined and their underlying logic explained, the subsequent chapter will focus on their practical implementation, computational testing using generated instances, and a thorough analysis of the results obtained.

Chapter 4

Experimental Phase: Two Approaches to Solve the Problem

After having presented the mathematical formulation of the integrated model encompassing both the strategic deployment and operational routing aspects of the STM electrified OS vehicle fleet in the preceding chapter, we now transition to the resolution phase. This phase initially involved leveraging the capabilities of the commercial integer programming solver, IBM ILOG CPLEX, to obtain exact solutions for our model. Subsequently, recognizing the potential computational demands of solving large-scale instances to optimality, we also explored the development of heuristic approaches.

This chapter will first detail the crucial instance generation phase, which provides the necessary data for testing our models. We will then separately analyze the implementation and resolution using the exact solver (CPLEX) and discuss the heuristic development.

4.1 Instance Generation

The generation of test instances is a critical preliminary phase in computational optimization studies. It involves creating specific problem examples used to rigorously test and evaluate the performance of mathematical models and solution algorithms. The validity of this generation process is fundamental to ensuring the consistency and reliability of the entire validation and resolution process.

In the context of this research, while the STM provided valuable real-world operational data, relying solely on this single dataset would limit our ability to rigorously assess the scalability, robustness, and general performance characteristics of the proposed mathematical model under varying conditions. Publicly available benchmark instances that precisely match the specific problem characteristics including multiple depots, incident time windows reflecting their occurrence patterns, diverse service times, and the unique operational aspects of OS vehicles are not readily available.

Therefore, to comprehensively evaluate the capacity of our optimization model across a spectrum of scenarios and to ensure statistically sound performance analysis, a tailored instance generator was developed. This generator allows for the creation of multiple instance families, where each family is defined by a specific combination of key structural parameters (e.g., the number of incidents N, the number of available vehicles K, and the number of depots D). For each defined family, we systematically generate a set of 25 distinct instances. This number (25 instances per family) was chosen to provide a sufficient sample size for performance evaluation, allowing for more reliable statistical inferences about the model's behavior for a given problem size and structure. While the structural parameters define a family, the specific data within each of the 25 instances (such as incident locations, exact occurrence times within windows, and service durations) are generated in a uniform manner, drawing inspiration from the characteristics and distributions observed in the real operational data provided by the STM. This approach ensures that our test instances are not only diverse but also grounded in realistic operational contexts.

Instance Format: Each test instance is stored in a plain text file with the following format:

- N: The total number of incidents to be serviced.
- D: The total number of depots from which OS vehicles can be dispatched.
- K: The total number of available OS vehicles (this parameter is subject to optimization).
- V: The total number of nodes in the network, where $V = N + D$ (representing all incidents and all depots).
- e_i: A vector of length N, where e_i is the occurrence time for incident i.
- l_i: A vector of length N, where l_i is the maximum allowable duration for an incident to remain unresolved (e.g., a bus cannot remain out of service for more than a specified time limit after its occurrence).

- s_i: A vector of length N, where s_i is the service time required to resolve the incident i.

- t_{ij}: An V x V matrix representing the travel times between all pairs of nodes (depots and incidents). This matrix is derived by first calculating the geographical distances between nodes and then converting these distances to travel times, assuming a constant vehicle speed.

Figure 4.1 illustrates an example of an instance file for a small problem with $N = 5$ incidents, $D = 3$ depots, and $K = 4$ vehicles, where the yellow rectangle corresponds to N, the blue to D, the orange to K, the green to V, the gray to the time windows with the first line e_i and the second line l_i, the purple to the vector s_i, and the red to the matrix of t_{ij}.

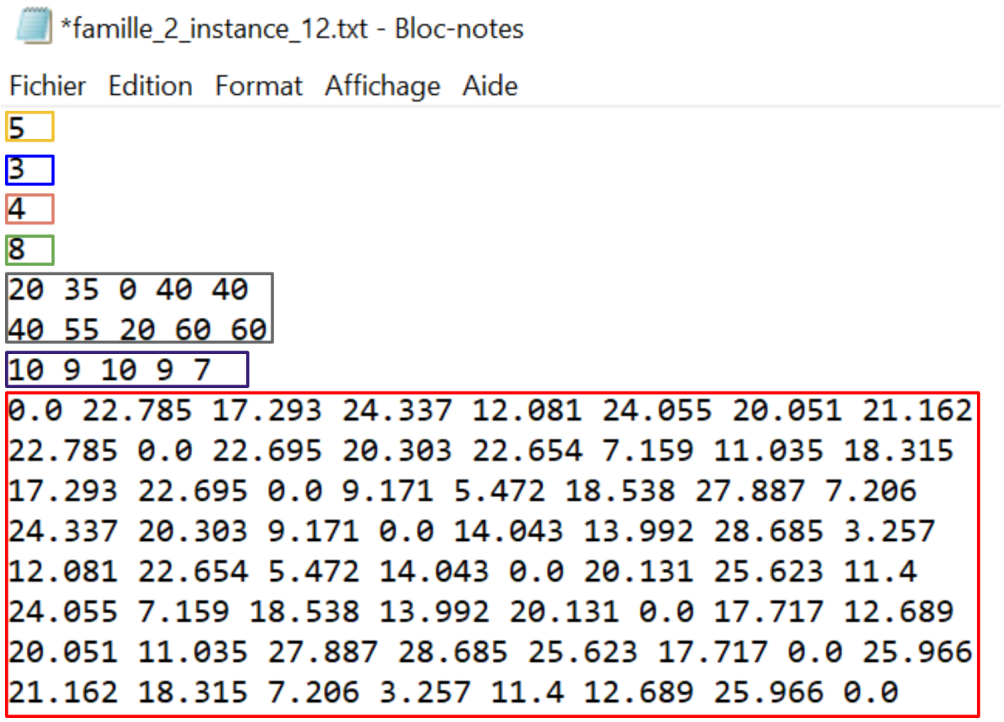

Figure 4.1: Example structure of an instance file (N=5, D=3, K=4)

Instance Generation Process: A dedicated instance generation algorithm was developed to automate this process. Each family is generated by varying the structural parameters N, D, and K incrementally most often using a step size of 1 to assess how the model performs as problem complexity increases.

The geographical coordinates of incidents are randomly generated within the Island of Montreal to reflect the uncertainty of demand locations, as incidents can occur anywhere on the island. To calculate travel times more realistically, considering the curvature of the Earth, the Haversine formula is used instead of the Euclidean distance. This approach provides more accurate distances between nodes on the spherical surface of the Earth.

Service times s_i are sampled from empirical distributions based on historical intervention durations, which typically average around 30 minutes. Regarding time windows $[e_i, l_i]$, their generation takes into account operational constraints observed in real data. In particular, time windows are set within the first five hours of the shift, with an upper bound of one hour for incident occurrence. This prevents generating incident times near the end of the shift, which could lead to infeasible schedules.

For example, if a 6h-shift starts at 2:00 pm and an incident occurs at 7:50 pm and the shift ends at 8:00 pm, with a service time of 30 minutes plus travel times to and from the depot, it would be impossible for the model to find a feasible solution since the incident cannot be serviced within the shift. By restricting incident occurrence times to the early part of the shift, the model ensures all incidents can be feasibly served, and it respects the operational rule that a bus cannot remain broken down for more than one hour.

The data generator for the instances was developed using Python. The underlying algorithm is detailed in Algorithm 3 below:

Algorithm 3: Instance Data Generator

Input: Parameters of the instance family (e.g. family index, number of instances)

1: Initialize fixed parameters: N, D, K

2: Generate earliest time windows $e[i]$ randomly within the shift

3: Set latest time windows $l[i] = e[i]+$ fixed margin defined by the company

4: Generate random service times $s[i]$

5: Generate random GPS coordinates for all $V = N + D$ points

6: **For each** pair of points (i, j):

7: Compute distance using Haversine formula

8: Multiply by a factor (1/speed) to obtain travel time

9: **End for**

10: Write all data into a text file

11: Save the file into the dedicated output folder

Output: A folder containing all generated instances for the same family

4.2 Model Implementation Using CPLEX

To evaluate the computational performance of the proposed Mixed-Integer Linear Program (MILP) formulation for optimizing the STM's OS vehicle fleet, we implemented the model and conducted numerical experiments. This section details the development environment, programming language, optimization solver, and the steps involved in translating our mathematical model into a solvable format.

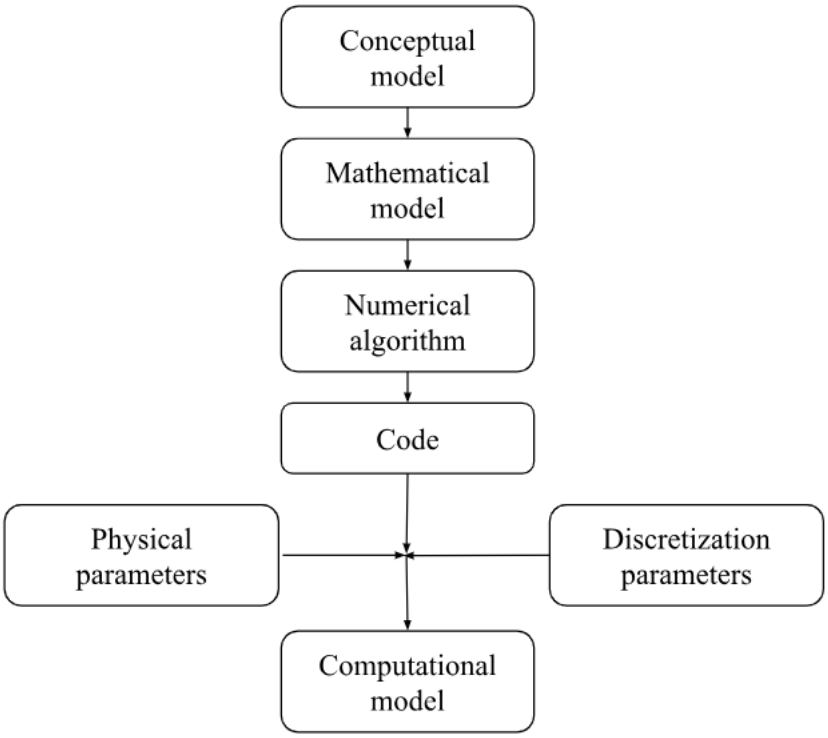

Figure 4.2: Process of Model Development [14]

4.2.1 Development Environment

Mathematical optimization models can be implemented using a variety of programming languages such as C++, Java, and Python. These languages offer different trade-offs in terms of performance, ease of use, and integration capabilities with optimization solvers. To solve Mixed Integer Linear Programming (MILP) models, a wide range of solvers is available, including commercial solutions like IBM ILOG CPLEX and Gurobi, as well as open-source alternatives such as SCIP and CBC. These solvers are designed to efficiently handle complex optimization problems involving thousands of variables and constraints.

For this work, we chose to use IBM ILOG CPLEX Optimization Studio, one of the most powerful and widely recognized solvers for mathematical programming and among the supported programming languages, we selected Java for the implementation. Java is an object-oriented language known for its platform independence, robustness, and rich standard libraries. In our project, Java was used to define the MILP model structure, load input instance data, interact with the CPLEX API to solve the model, and retrieve the results for analysis.

Numerical experiments were conducted on a machine equipped with an Intel Core i5-6300U (2 cores, 4 threads, 2.4 GHz base frequency) and 8 GB of RAM. A computational time limit of 600 seconds (10 minutes) was imposed for each optimization run to ensure practical solution times.

4.2.2 Programming Language (Java)

Java is an object-oriented programming language known for its platform independence, robustness, and extensive standard libraries. It facilitates the development of structured and maintainable code, making it particularly suitable for optimization-based applications. In this work, Java was chosen as the primary language for implementing the MILP model and interfacing with the CPLEX solver. Its main advantages include high portability across operating systems, strong security features, and broad library support. These characteristics make Java well-suited for developing complex optimization systems that demand both computational efficiency and long-term maintainability. We used Java to define the model structure, load instance data, invoke the CPLEX solver, and process the resulting solutions.

4.2.3 Solver (IBM ILOG CPLEX Optimization Studio)

IBM ILOG CPLEX Optimization Studio (referred to as CPLEX) is a powerful software suite dedicated to mathematical optimization. Originally developed by ILOG and later acquired by IBM, CPLEX is designed to solve a wide range of optimization problems, including linear programs (LP), quadratic programs (QP), and notably for our purposes, mixed-integer linear programs (MILP). These MILPs are particularly relevant as they can model complex decision-making problems with both continuous and discrete variables, aiming to find the best possible solution from a potentially vast set of feasible options. CPLEX's functionalities include advanced algorithms for solving these problem types and the capability to handle intricate constraints and objective functions. It provides Application Programming Interfaces (APIs) for several popular programming languages, including C++, Java, and Python, facilitating the integration of its optimization capabilities into custom applications. We opted for CPLEX due to several compelling reasons:

- **Popularity and Trust:** It is one of the most widely used and respected solvers among operations research practitioners and academics.
- **Academic Availability:** It is offered free of charge for academic research, which is crucial for university-led projects.
- **Comprehensive Documentation:** While some suggest updates have been less frequent post-IBM acquisition, the existing documentation remains extensive and highly valuable for users.
- **Solver Performance:** Its advanced branch-and-bound and cutting plane algorithms ensure efficient solution times for large-scale MILPs.

4.2.4 Modeling Steps in Java using CPLEX Concert Technology

To implement the proposed MILP model within the Java environment and utilize the CPLEX solver, we leveraged CPLEX's Concert Technology. Concert Technology pro-

vides an object-oriented API that allows users to define optimization models, variables, objective functions, and constraints in a way that is independent of the underlying solver algorithm.

This section outlines the general steps involved in modeling our problem using this library:

1. **Import the required libraries**

```
import ilog.concert.*;
import ilog.cplex.*;
```

   This step gives access to the CPLEX optimization and Concert API classes.

2. **Create the environment**

```
IloCplex cplex = new IloCplex();
```

   The `IloCplex` object represents both the model and the environment in Java.

3. **Declare and add decision variables to the model**

```
IloNumVar[] y = new IloNumVar[K];
for (int k = 0; k < K; k++) {
    y[k] = cplex.boolVar();
}
```

   Here, we define a binary decision variable y_k for each vehicle $k \in \{1, \ldots, K\}$. The method `cplex.boolVar()` creates variables that can only take the values 0 or 1. These are typically used to model on/off decisions for example, indicating whether a vehicle k is used in the solution ($y_k = 1$) or not ($y_k = 0$).

4. **Define the objective function**

```
IloLinearNumExpr obj = cplex.linearNumExpr();
for (int k = 0; k < K; k++) {
    obj.addTerm(1, y[k]);
}
cplex.addMinimize(obj);
```

   In this formulation, the objective is to minimize the total number of vehicles used. The expression sums all binary variables y_k, where $y_k = 1$ indicates that vehicle k is activated. Thus, $\sum_{k=1}^{K} y_k$ represents the total number of active vehicles in the solution. By minimizing this sum, the model encourages using as few vehicles as necessary to satisfy the problem constraints.

5. **Define constraints**

```
for (int j = 0; j < N; j++) {
    IloLinearNumExpr expr = cplex.linearNumExpr();
    for (int k = 0; k < K; k++) {
        for (int i = 0; i < V; i++) {
            if (j != i) {
                expr.addTerm(1, x[i][j][k]);
```

```
                }
            }
        }
        cplex.addEq(expr, 1);
    }
```

This constraint ensures that each incident node j is visited exactly once by one vehicle. The decision variable $x[i][j][k]$ equals 1 if vehicle k travels from node i to node j, and 0 otherwise. The inner loops sum all such possible incoming routes to node j across all vehicles k and origin nodes i (excluding $i = j$ to avoid loops). The equality constraint $\sum_{k=1}^{K} \sum_{\substack{i=1 \\ i \neq j}}^{V} x_{ijk} = 1$ enforces that exactly one vehicle enters each node j, satisfying the single-visit requirement.

6. **Solve the model**

```
if (cplex.solve()) {
    // Proceed if solved successfully
}
```

7. **Retrieve and display results**

```
System.out.println("Status = " + cplex.getStatus());
System.out.println("Objective Value = " + cplex.getObjValue());
System.out.println("Gap = " + cplex.getMIPRelativeGap());

Display values of a[i][k]
for (int i = 0; i < N; i++) {
    for (int k = 0; k < K; k++) {
        if (cplex.getValue(a[i][k]) > 0.5) {
            System.out.println("a[" + i + "][" + k + "] = 1");
        }
    }
}
```

After solving the model, this code block prints the solver status (e.g., 'Optimal', 'Feasible'), the value of the objective function, and the MIP relative gap, which quantifies the optimality gap for mixed-integer programs. The loop iterates through the binary assignment variables a_{ik}, printing the pairs where the variable is activated (i.e., equals 1), indicating that technician k is assigned to incident i. Execution time can also be displayed if you manually record timestamps before and after the solve process.

8. **Release memory**

```
cplex.end();
```

It's essential to close the model to avoid memory leaks.

4.3 Results and Discussion

This section presents the computational results obtained by solving the proposed Mixed-Integer Linear Programming (MILP) model. We evaluate its performance across various families of instances and discuss the implications of the results.

To systematically investigate the scalability of the model, we gradually increased the size and complexity of the instances by expanding the parameters N, D, and K within each family. We then applied our solution method to these progressively larger instances, continuing the process until the model could no longer find a solution within the predefined computational time limit.

This approach enabled us to pinpoint the practical limits of the model's solvability and to identify where performance bottlenecks or infeasibilities begin to emerge. These insights are essential for evaluating the model's real-world applicability, particularly in time-sensitive operational settings.

4.3.1 MILP Results

To evaluate the computational efficacy of the MILP model, we conducted a series of experiments across varying instance families, primarily differing in the number of incidents, depots, and available vehicles. Table 4.1 summarizes the key performance metrics obtained from these experiments. For each instance family, we report the structural characteristics, namely the number of decision variables and constraints generated by the model. Subsequently, for each of the two objective functions within our lexicographical approach, two rows are displayed: the first corresponds to the results obtained when optimizing Objective 1 (minimizing fleet size, 3.27), and the second to the results when optimizing Objective 2 (minimizing total arrival time, 3.28), given the fleet size fixed by the first objective.

For each objective, we indicate the number of instances (out of 25 per family) that were solved to proven optimality (i.e., with an optimality gap of 0%) within the allocated time limit. For instances not solved to optimality, we report the average and standard deviation of the final optimality gap. We also report the average execution time in seconds, along with its standard deviation, for each objective function. It is important to note that the overall 10-minute (600 seconds) time limit was applied cumulatively for solving both objectives within the lexicographic approach for each instance.

4.3.2 Discussion of MILP Results

The computational results presented in Table 4.1 provide valuable insights into the performance and scalability of the exact solution approach. Overall, the results indicate that the first objective, minimizing fleet size (Objective 1), is generally solved more rapidly and to optimality more frequently than the second objective, which aims to minimize the total arrival time (Objective 2). The model demonstrates its ability to consistently

Table 4.1: MILP Computational Outcomes: Time and Gap Metrics

Class $(\|\mathcal{N}\|, \|\mathcal{K}\|, \|\mathcal{D}\|)$	**Nb Var**	**Nb Cons**	**Solved Opt**	**Gap %**		**CPLEX time (s)**	
				Average	**Standard deviation**	**Average**	**Standard deviation**
(5, 2, 3)	198	356	25	-	-	0.09	0.03
			25	-	-	0.24	0.10
(6, 2, 4)	332	601	25	-	-	0.14	0.07
			25	-	-	0.37	0.16
(7, 3, 5)	620	1100	25	-	-	0.22	0.08
			25	-	-	1.17	0.56
(8, 3, 5)	735	1322	25	-	-	0,55	0,36
			25	-	-	5,28	6,05
(9, 3, 6)	1032	1873	25	-	-	1.35	1.02
			25	-	-	13.94	16.20
(10, 4, 7)	1603	2870	25	-	-	51.44	73.79
			14	15.03%	7.60%	336.06	209.23
(11, 4, 7)	1820	3292	24	33.33%	0.00%	135.40	355.42
			7	23.29%	8.71%	164.75	198.40
(12, 4, 7)	2051	3742	21	33.33%	0.00%	275,93	252,14
			1	28,71%	8,42%	476,62	134,59

solve instances with up to 9 incidents to optimality for both objectives within the 10-minute time limit. However, as the problem size increases, particularly to 10 incidents, the model's ability to find and prove optimal solutions for the second objective within the time limit diminishes. For instance, while Objective 1 is solved to optimality for all 25 instances in the 10-incident family, there are 11 instances where Objective 2 yields an average optimality gap of approximately 15%, indicating that the exact approach begins to struggle with these larger, more complex scenarios. Furthermore, the performance varies notably even across instances of similar nominal size (e.g., same number of incidents, depots, and vehicles). This suggests that specific characteristics of an instance, such as the spatial distribution of incidents relative to depots or the tightness of time windows, significantly impact the underlying problem complexity and, consequently, the solver's efficiency. These observations highlight the inherent limitations of the current exact method in consistently handling complex or particularly challenging scenarios within practical time limits, pointing towards the need for more scalable and potentially adaptive approaches, such as the heuristic or decomposition-based techniques discussed earlier. To further understand the behavior of the solution times and optimality gaps, box plots were generated for these metrics across different instance families, comparing the performance for Objective 1 and Objective 2.

A **box plot** is a standardized way of displaying the distribution of data based on a five-number summary: minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum. It can also highlight outliers.

For the box plot of solution times, a logarithmic scale was used on the y-axis. This choice was made because the values for small instances are very low and then increase significantly for larger instances. Using a logarithmic scale allows for better visualization and comparison across the full range of values. A logarithmic scale is a nonlinear scale that represents data by orders of magnitude rather than a fixed unit interval; it is particularly useful when the data spans several orders of magnitude. In contrast, for the box plot of the optimality gaps, a logarithmic scale was not necessary, as many of the values are equal to zero and the range is relatively limited.

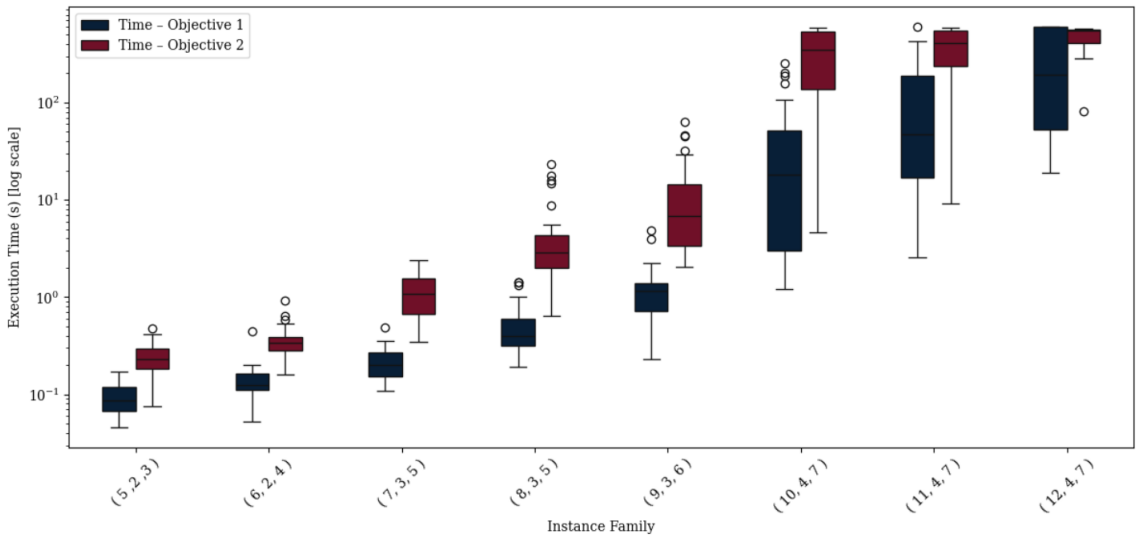

Figure 4.3: Boxplot of Execution Time (log scale) per Family

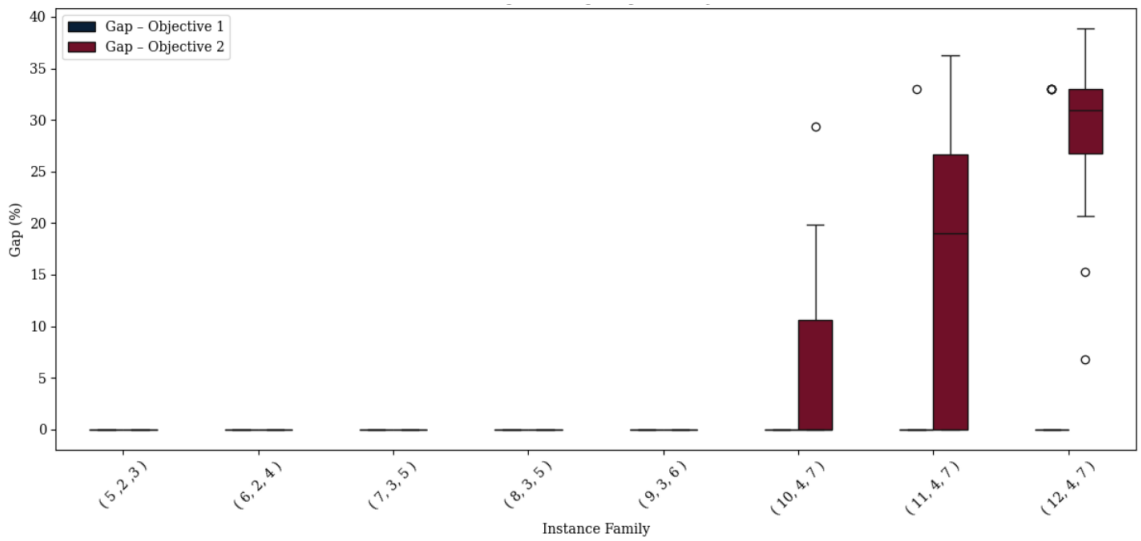

Figure 4.4: Boxplot of Gap(%) per Family

Analyzing Figures 4.3 and 4.4, several key observations emerge:

1. **Execution Time Discrepancy:** Figure 4.3 clearly shows that the execution time for Objective 1 is consistently lower than that for Objective 2 across all instance families. The model typically finds the optimal number of vehicles relatively quickly. This is likely because determining the minimum fleet size is a structurally simpler problem compared to the subsequent detailed routing and scheduling required to minimize total arrival times, which also incorporates the fixed fleet size constraint from the first objective as imposed by the lexicographic approach.

2. **Combinatorial Explosion and Time Increase:** As the problem instances grow in size (e.g., more incidents), the solution time particularly for Objective 2 tends to increase significantly, sometimes exponentially. This is characteristic of NP-hard combinatorial optimization problems, reflecting the "combinatorial explosion" where the number of possible solutions grows immensely with problem size.

3. **Presence of Outliers:** Both box plots exhibit outliers, particularly for larger instance sizes. These outliers represent instances within a family that are significantly harder to solve (requiring more time or resulting in larger gaps) than their peers. This variability underscores the point that instance-specific characteristics, beyond just the raw number of incidents or vehicles, heavily influence computational difficulty. Testing across a representative family of instances (25 in our case) helps capture this variability and ensures more robust conclusions about model performance.

4. **Time Limit Impact:** Given the cumulative 10-minute time limit for both objectives, there are scenarios—especially with larger instances where a substantial portion of the allocated time is consumed by solving (or attempting to solve) Objective 1. In some extreme outlier cases, the solver might exhaust the time limit on the first objective, leaving little to no time for the second, which directly impacts the reported time and gap for Objective 2.

5. **Interpretation of Gaps for Objective 1:** The optimality gaps reported for Objective 1 (minimizing fleet size) can sometimes appear large (e.g., 25% or 33%) even if the absolute difference in the number of vehicles is small. Since the objective value (number of vehicles) is a small integer (e.g., 2, 3, 4), if the optimal solution is, for example, 3 vehicles and the solver finds a solution with 4 vehicles before terminating, the gap would be $(4-3)/3 \approx 33\%$. This is a consequence of the discrete and small-valued nature of this particular objective function and often occurs in outlier instances where finding that last optimal vehicle assignment is particularly difficult.

6. **Larger Gaps for Objective 2:** Figure 4.4 generally shows larger and more variable optimality gaps for Objective 2 compared to Objective 1. This is expected, as Objective 2 is solved after Objective 1, often with less remaining time from the cumulative limit. Additionally, it is a more complex objective involving intricate routing and scheduling decisions. The performance on Objective 1 both time taken and quality of the fixed fleet size passed to Objective 2 directly influences the resources available and the starting point for optimizing Objective 2.

In conclusion, the analysis of the MILP model's performance demonstrates its capability to effectively solve smaller to moderately sized instances of the STM OS vehicle optimization problem. However, as instance size and complexity increase, the exact approach faces significant computational challenges, manifested in longer solution times and non-zero optimality gaps within the practical time limits imposed. This is characteristic of NP-hard combinatorial optimization problems where the solution space explodes with increasing size. Relying solely on an exact solver like CPLEX to find proven optimal solutions for large-scale, real-world instances within operational timeframes (which might be minutes, not hours or days) is often not feasible. This underscores the necessity for developing efficient methods, which can provide high-quality solutions for larger instances in a tolerable amount of computation time, making them more suitable for practical deployment.

4.3.3 Heuristic Results

As demonstrated in the preceding analysis, while the MILP model provides optimal solutions for smaller instances, its computational performance degrades significantly when applied to larger, more complex scenarios representative of a full operational day for the STM. The exact model was unable to consistently solve instances beyond a certain size (e.g., around 10 incidents) to optimality within the imposed time limits. This scalability challenge necessitates the development of alternative approaches capable of handling larger problem instances efficiently.

To this end, we proposed a decomposition-based heuristic technique, as detailed in Chapter 3, Section 3. This heuristic is inspired by the natural hierarchical organization of the STM's operational work, which involves shifts, depot service areas, and distinct geographical sectors. The core idea is to strategically break down the overall large-scale problem into a series of smaller, more manageable subproblems. The key advantage of this approach is its ability to leverage the proven capability of the developed MILP model to solve these smaller subproblems (notably those with a maximum of around 10 incidents).

While the full automation and comprehensive testing of this decomposition heuristic are still ongoing, preliminary experiments have been conducted to validate its potential and gather initial performance insights. For this initial validation, we considered a historical data sample representing a significant portion of daily operations (approximately one-third of a day). This sample focused on a region encompassing 5 specific sectors where, in reality, approximately 80 incidents are typically handled over a day (divided into 4 operational shifts). These incidents are serviced by vehicles operating from 3 designated depots, with the assignment of sectors to their primary servicing depots defined by existing STM operational policy, as outlined in Table 4.2.

Depot	Sectors
depot1	S6, S7
depot2	S1, S2, S3
depot3	S5, S15
depot4	S8, S9
depot5	S4, S14
depot6	S5, S10
depot7	S11, S12, S13
depot8	S6, S7

Table 4.2: Assignment of Sectors to Depots

By applying the decomposition technique to this data sample, the problem was broken down into 12 distinct subproblems. Each of these subproblems was then solved using the exact MILP formulation described in Section 3.2.

The results obtained from solving these subproblems are summarized in Table 4.3. All 12 subproblems were solved to proven optimality (optimality gap = 0%) for both lexicographical objectives. Notably, the total cumulative computation time to solve all 12 subproblems for both objectives was remarkably low, under 6 seconds. Analyzing the performance for each objective within these subproblems:

- For Objective 1 (minimizing fleet size per subproblem), the average execution time was approximately 0.18 seconds, with a standard deviation of 0.3 seconds.
- For Objective 2 (minimizing total arrival time per subproblem, given the fleet size from Objective 1), the average execution time was approximately 0.83 seconds, with a standard deviation of 1.41 seconds.

Consistent with the findings from the direct MILP application on varying instance sizes, the time required to solve the second objective (routing and scheduling) was generally higher than that for the first objective (fleet sizing) within each subproblem.

These preliminary findings are highly encouraging. They suggest that the decomposition-based technique is an effective strategy for tackling the STM OS vehicle optimization

Shift	Depot	Sector	Incidents	Time 1 (s)	Time 2 (s)
Shift 1	D6	S5, S10	6	0.071	0.095
	D4	S8, S9	6	0.059	0.229
Shift 2	D6	S10	7	0.121	0.698
	D7	S11	4	0.023	0.04
	D6	S5	8	0.104	0.742
Shift 3	D6	S10	6	0.042	0.411
	D7	S11	9	0.199	0.119
	D6	S5	6	0.043	0.378
	D4	S8, S9	9	0.157	0.592
Shift 4	D7	S11	6	0.064	0.199
	D6	S5, S10	5	0.055	0.246
	D4	S8, S9	7	0.2	0.647

Table 4.3: Results of the Execution of the Heuristic on a Test Instance

problem. By breaking the larger problem into smaller pieces that the exact MILP can handle efficiently, we can achieve optimal solutions for these sub-units rapidly. While the complete automation of the decomposition process and the development of sophisticated aggregation methods for the subproblem solutions are still under development, these initial results strongly indicate the viability and significant potential of this heuristic approach to provide high-quality, scalable solutions for the STM.

4.4 Action Plan for Model Integration and Use

Having developed and experimentally validated the mathematical models in the preceding sections, this final technical part of this thesis outlines a practical action plan for integrating these optimization tools into the STM's operational planning processes. The ultimate value of a mathematical model is realized through its consistent application in a real-world environment. Therefore, this section proposes the development of a comprehensive Decision Support Platform designed to transform the MILP and heuristic models into a tangible, user-friendly tool for STM planners and managers. This platform would not only solve the optimization problems but also provide crucial functionalities for data visualization, scenario analysis, and strategic planning.

The proposed platform would be built around four core functionalities, supported by a robust and scalable system architecture.

4.4.1 Core Platform Functionalities

1. **Real-Time Data Monitoring and Visualization:**
   The platform's foundation would be a dynamic dashboard, potentially developed

using a business intelligence service like Power BI. This component would connect to STM's operational databases to provide a real-time or near-real-time overview of incident data. Users could visualize incident locations on a map of Montreal, filter by type, time of day, or sector, and track key performance indicators (KPIs), providing essential situational awareness for daily operations and a rich data source for strategic analysis.

2. **Scenario Generation and Management:**
   A key feature for strategic planning would be the ability to generate and analyze "what-if" scenarios. The platform would provide an interface for users to:

   - **Define Scenario Parameters:** Planners could easily set parameters to create new instances, such as the number of expected incidents (N), the set of available depots (D), potential fleet sizes (K), and other operational constraints (e.g., shift durations, EV range).
   - **Visualize Scenarios:** Upon generation, the system would display the instance graphically on a map, showing the locations of incidents and depots. This visual feedback is crucial for ensuring the generated scenario is realistic and for intuitive understanding before launching the optimization.

3. **Optimization Model Resolution (MILP & Heuristic):**
   The core of the platform would be its optimization engine. The user interface would allow a planner to:

   - **Select an Instance:** Choose a generated scenario or a historical dataset to solve.
   - **Choose a Solution Method:** Select either the Exact MILP Solver for smaller, critical problems where optimality is required, or the Decomposition Heuristic for larger-scale instances where a fast, high-quality solution is preferred.
   - **Execute and Display Results:** The system would dispatch the problem to a specialized computational unit for resolution. Upon completion, the results would be displayed clearly, including:
     - The optimal fleet size and vehicle-to-depot assignments.
     - Detailed vehicle routes and schedules visualized on the map.
     - A summary of key performance indicators (e.g., total response time, average vehicle utilization, total distance traveled).

4. **Heuristic Resolution:**
   As the decomposition heuristic relies on solving smaller MILP subproblems, this functionality would be integrated within the optimization engine. When a user selects the heuristic approach, the platform would automatically execute the multi-level problem decomposition (by shift, depot, and sector), solve each subproblem using the MILP solver, and then run the solution aggregation logic to present a cohesive global solution.

4.4.2 Proposed System Architecture

To support these functionalities, a flexible and scalable software architecture is necessary. A microservices-based architecture is proposed, as it allows each core functionality (e.g., instance generation, optimization, data storage) to be developed, deployed, and updated independently. This modularity is ideal for complex applications, enhancing maintainability and resilience.

These independent services would communicate via a standardized REST API (Representational State Transfer Application Programming Interface). The REST API acts as a universal gateway, allowing the user-facing application to interact with the backend services using standard HTTP requests. For example, a user action in the interface would translate to an API call: a GET request to retrieve historical incident data, a POST request to create and save a new scenario, or another POST request to launch an optimization job. This decouples the user interface from the complex logic of the backend, allowing for greater flexibility in development.

The systemic needs to support this architecture include:

- **User Interface:** A web-based application providing access to all platform functionalities.
- **API Gateway:** A central point for managing and routing all REST API requests.
- **Backend Services:**
    - A Storage Service connected to a dedicated database for storing instances, historical data, and solution results.
    - An Optimization Service that manages the queue of solving jobs and interfaces with the CPLEX solver.
- **Specialized Compute Units:** Powerful servers dedicated to running the computationally intensive MILP and heuristic algorithms.
- **Data Visualization Service:** A service to process solution data and render it on maps and charts for the user interface.

4.4.3 User Workflow and Activity Diagram

The typical user workflow on the proposed platform is illustrated in the activity diagram below (Figure 4.5). This diagram shows the sequence of actions a user, such as an STM planner, would take to analyze a problem and obtain an optimized solution. The process begins with either analyzing existing data or creating a new scenario, proceeds through the selection of a solution method, and concludes with the visualization and storage of the results.

This structured action plan provides a clear roadmap for translating the academic contributions of this thesis into a powerful, practical tool that can directly support the strategic and operational goals of the STM.

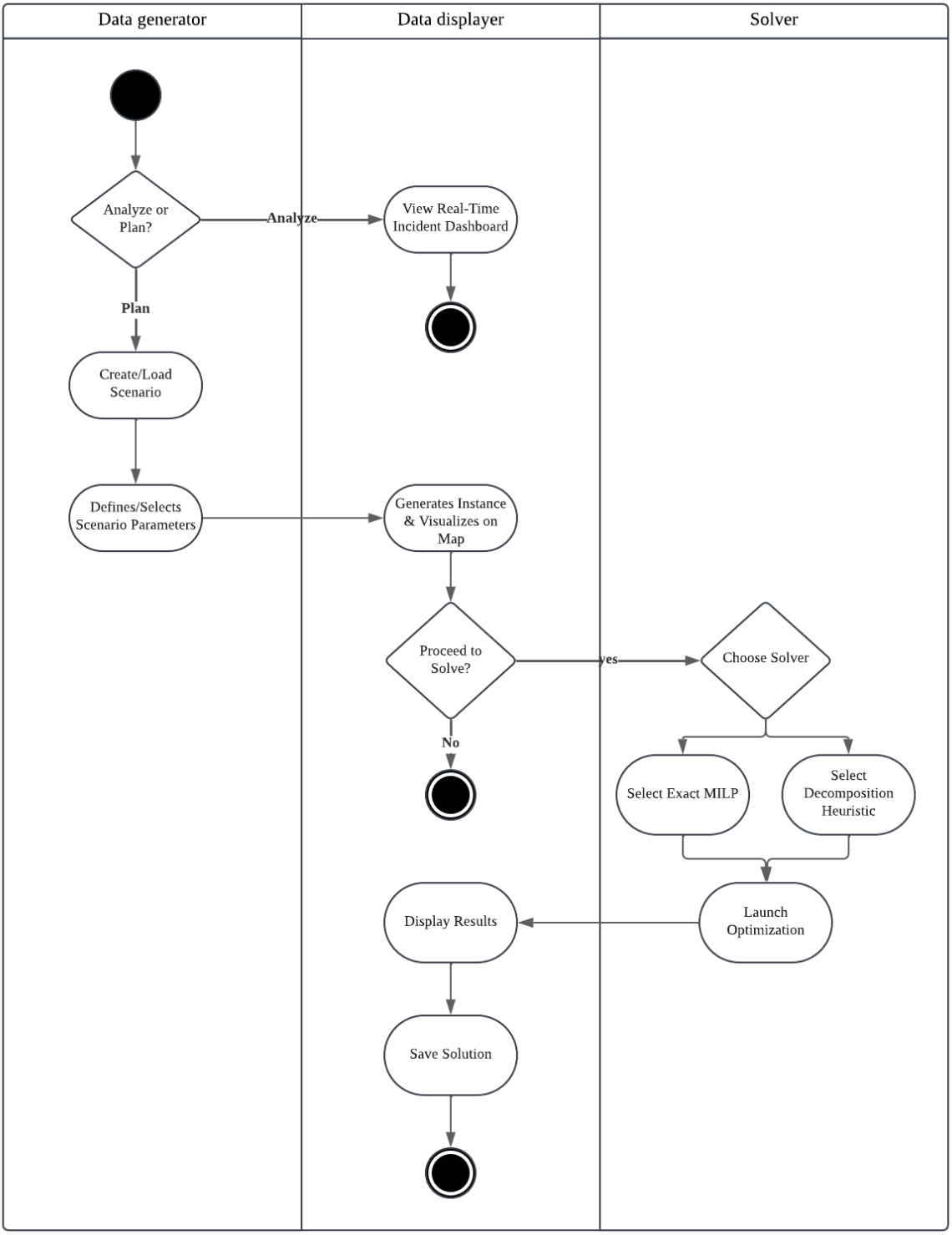

Figure 4.5: Platform Activity Diagram

4.5 Conclusion

This chapter has presented a comprehensive evaluation of the solution approaches proposed in the previous chapter through a carefully designed experimental framework. Beginning with the generation of diverse and representative instances and the implementation of the model.

The computational results have confirmed the strengths of the exact MILP approach in solving small to moderately sized instances, particularly in minimizing fleet size, the primary objective. However, as the problem size increased, the MILP model revealed its limitations especially in addressing the secondary objective of minimizing arrival times due to the computational complexity of NP-hard problems. The difficulty in obtaining optimal solutions within a reasonable time frame for larger instances highlighted the need for scalable and more flexible alternatives.

In response to these limitations, a decomposition-based heuristic was introduced and preliminarily validated. By breaking down the problem into smaller, more manageable subproblems aligned with the STM's operational structure, this approach leverages the strengths of exact optimization on a reduced scale. Initial experiments using this heuristic on representative data yielded promising results, demonstrating its potential for addressing real-world problem sizes more efficiently.

In conclusion, this chapter has not only illustrated the practical boundaries of exact MILP-based optimization for STM problems but also demonstrated the limitations of exact approaches in addressing NP-hard problems. Additionally, it laid the groundwork for a promising heuristic alternative, which is still under development particularly regarding automation and the aggregation of subproblem solutions.

General Conclusion

This thesis has addressed the complex challenge of optimizing the electrified Operational Supervisor (OS) vehicle fleet for the Société de transport de Montréal (STM). Driven by the STM's commitment to sustainability and operational efficiency, our research aimed to develop a robust decision-support framework for determining the optimal fleet size and deployment strategy for these critical service vehicles, particularly considering the unique constraints introduced by electrification.

The study commenced with a detailed problem description, contextualizing the operational environment of the STM and the pivotal role of OS vehicles in maintaining network reliability. This was followed by a comprehensive literature review, which surveyed foundational concepts in optimization, relevant problems such as facility location and vehicle routing with its many variants (including those with time windows, multiple depots, and electric vehicle considerations), and established solution methodologies. This review confirmed that while individual components of fleet management are well-studied, an integrated approach for the strategic sizing and deployment of an electrified operational support fleet, like the STM's OS vehicles, represented a notable gap in existing research.

Subsequently, we developed a conceptual model capturing the operational dynamics of the OS fleet, which was then translated into a rigorous Mixed-Integer Linear Program (MILP). This mathematical formulation was designed to integrate key decision variables related to fleet sizing and deployment, operational constraints (including shift durations, vehicle range, depot capacities, and incident time windows), and the lexicographical objectives of first minimizing the total number of vehicles and then minimizing the total incident arrival time.

The experimental phase involved the generation of diverse test instances, inspired by STM's operational data, and the implementation of the MILP model using Java and the IBM ILOG CPLEX solver. The results demonstrated the MILP's efficacy in solving smaller to moderately sized instances to optimality. However, as anticipated for an NP-hard combinatorial optimization problem, scalability limitations became evident with larger instances, where finding proven optimal solutions within practical time limits proved challenging. This led to the exploration and preliminary validation of a decomposition-based heuristic. This heuristic, inspired by the STM's operational structure (shifts, depots, sectors), aims to break down the larger problem into manageable subproblems that can be efficiently solved by the core MILP. Initial results for this decomposition approach were highly promising, showcasing its potential to provide high-quality solutions for larger-scale scenarios in significantly reduced computation times.

Summary of Findings

The primary contributions and findings of this research can be summarized as follows:

1. **Problem Formulation for a Real-World Challenge:** We addressed a pertinent and practical problem faced by a major public transit agency (STM) concerning the optimization of its specialized electrified OS vehicle fleet. To the best of our knowledge, this specific integrated problem of sizing and deploying such a fleet, with its unique operational context, has not been extensively treated in the existing academic literature.
2. **Development of a Novel Mathematical Model:** A comprehensive Mixed-Integer Linear Program (MILP) was developed. This model integrates crucial aspects of fleet sizing, multi-depot assignment, vehicle routing with time windows, and considerations relevant to electric vehicles, all within a lexicographical optimization framework.
3. **Validation of MILP Performance:** The computational capabilities of the exact MILP model were rigorously evaluated. We identified its strengths in solving smaller instances and clearly demarcated its scalability limits, providing a benchmark for the performance of exact methods on this class of problem.
4. **Proposal and Preliminary Validation of a Decomposition-Based Heuristic:** Recognizing the limitations of the exact model for large-scale instances, a multi-level decomposition heuristic was conceptualized and its initial feasibility and effectiveness were demonstrated. This approach shows significant promise for practical application by leveraging the strengths of the MILP on smaller, more tractable subproblems.
5. **Academic Dissemination:** The significance and novelty of this research have been recognized through the acceptance of a research paper based on this work for publication and presentation at the [Name of IEEE/IFAC Conference, e.g., "IEEE Conference on X" or "IFAC Symposium Y"], scheduled for July 2025. This serves as an external validation of the research quality and its contribution to the field.

Limitations and Challenges

Despite the promising results, this study is subject to certain limitations and encountered several challenges:

1. **Computational Resources:** The computational experiments for the MILP model were conducted on a standard desktop computer (Intel Core i5, 8 GB RAM). While sufficient for initial validation, it is acknowledged that many academic and industrial optimization studies utilize significantly more powerful computing resources (e.g., servers with 16+ cores and larger memory). Access to such resources could potentially extend the solvable range of the exact MILP or reduce solution times.

2. **Heuristic Development Stage:** The proposed decomposition-based heuristic is currently in its preliminary stages of development. While initial results are encouraging, the full automation of the decomposition logic, the sophisticated aggregation of subproblem solutions (especially managing interdependencies or vehicle sharing across subproblem boundaries), and comprehensive testing across a wider range of large-scale instances are yet to be completed.

3. **Data Assumptions and Simplifications:** Like any modeling effort, certain assumptions were made to render the problem tractable, such as constant vehicle speeds, deterministic service times, and treating incidents with equal priority. Real-world operations often involve greater variability and dynamic events that are not fully captured in the current static model.

4. **MILP Scalability:** The inherent NP-hard nature of the underlying combinatorial optimization problem means that even with advanced solvers like CPLEX, finding proven optimal solutions for very large, real-world instances within tight operational deadlines will always remain a significant challenge for direct MILP application.

Future Directions for Research

The findings and limitations of this thesis open up several exciting avenues for future research and development:

1. **Full Development and Refinement of the Decomposition Heuristic:** A primary focus will be to complete the development of the decomposition heuristic, including robust methods for subproblem definition, efficient solution aggregation, and comprehensive benchmarking against the MILP and potentially other heuristics on large-scale instances.

2. **Exploration of Metaheuristic Approaches:** Given the problem's complexity, exploring established metaheuristics could yield effective solutions. Genetic Algorithms (GAs), for instance, are known for their efficacy in solving complex routing problems and could be adapted for this integrated sizing, deployment, and routing challenge. Other metaheuristics like Tabu Search, Simulated Annealing, or Adaptive Large Neighborhood Search (ALNS) also warrant investigation.

3. **Advanced Mathematical Modeling Techniques:** To enhance the solvability of the exact model for larger instances, advanced MILP techniques could be explored. This includes applying Benders decomposition, which is well-suited for problems with a particular structure, or column generation. Reformulating parts of the problem, perhaps drawing inspiration from how some routing problems have been successfully modeled as scheduling problems where efficient solution methods exist, could also be a fruitful direction.

4. **Incorporating Stochasticity and Dynamics:** Real-world incident occurrences, travel times, and service times are often stochastic. Future work could focus on incorporating this uncertainty. This might involve:

- *Simulation-Optimization:* Coupling the deterministic optimization model with a simulation model to evaluate solution robustness under uncertainty and to refine parameters.
- *Stochastic Programming or Robust Optimization:* Formulating models that explicitly account for uncertainty in input parameters.
- *Dynamic Approaches:* Developing dynamic dispatching or re-optimization strategies using techniques like rolling horizon planning or Markov Decision Processes to adapt to real-time information as incidents unfold.

5. **Evaluation of Alternative Solvers:** While CPLEX is a powerful solver, comparing its performance with other leading commercial solvers like Gurobi, or innovative solvers like Hexaly (which utilizes a list-based system for variable definition that can be efficient for certain problem structures), could provide insights into the best tools for this specific problem class.

6. **Integration of More Detailed EV Constraints:** Further refining the model to include more detailed aspects of electric vehicle operation, such as non-linear charging functions, battery degradation, or the impact of ambient temperature on range, would enhance its real-world applicability.

In conclusion, this thesis has laid a solid foundation for optimizing the STM's electrified OS vehicle fleet. The developed models and the promising initial results from the decomposition heuristic provide valuable tools and insights. The identified limitations and future research directions offer a clear roadmap for continued work towards even more sophisticated and robust solutions for this critical operational challenge.

Bibliography

[1] P Mala, M Palanivel, S Priyan, N Anbazhagan, Srijana Acharya, Gyanendra Prasad Joshi, and Joohan Ryoo. Sustainable decision-making approach for dual-channel manufacturing systems under space constraints. *Sustainability*, 13(20):11456, 2021.

[2] Société de transport de Montréal. Société de transport de montréal, 2025. Accessed on May 18, 2025 at 11:00 AM (UTC+01:00).

[3] Alexandra Dujonc. Fast charging of electric cars: Battery degradation myths and realities revealed, May 2025. Accessed on May 18, 2025 at 11:15 AM (UTC+01:00).

[4] Everton Gomede. The famous np problems, March 2023. Accessed on May 15, 2025 at 11:45 AM (UTC+01:00).

[5] Richard Perham. Driving efficiency for emergency services fleets, 2022. Accessed on May 10, 2025 at 12:00 PM (UTC+01:00).

[6] Virgilio C Guzmán, David A Pelta, and José L Verdegay. An approach for solving maximal covering location problems with fuzzy constraints. *International Journal of Computational Intelligence Systems*, 9(4):734–744, 2016.

[7] Jesica De Armas, Angel A Juan, Joan M Marquès, and João Pedro Pedroso. Solving the deterministic and stochastic uncapacitated facility location problem: from a heuristic to a simheuristic. *Journal of the Operational Research Society*, 68(10):1161–1176, 2017.

[8] Dipankar Dasgupta, German Hernandez, Deon Garrett, Pavan Kalyan Vejandla, Aishwarya Kaushal, Ramjee Yerneni, and James Simien. A comparison of multiobjective evolutionary algorithms with informed initialization and kuhn-munkres algorithm for the sailor assignment problem. In *Proceedings of the 10th annual conference companion on Genetic and evolutionary computation*, pages 2129–2134, 2008.

[9] Takwa Tlili, Sofiene Abidi, and Saoussen Krichen. A mathematical model for efficient emergency transportation in a disaster situation. *The American journal of emergency medicine*, 36(9):1585–1590, 2018.

[10] R. Yesodha and T. Amudha. A bio-inspired approach: Firefly algorithm for Multi-Depot Vehicle Routing Problem with Time Windows. *Computer Communications*, 190:48–56, 6 2022.

[11] Lu Zhen, Chengle Ma, Kai Wang, Liyang Xiao, and Wei Zhang. Multi-depot multi-trip vehicle routing problem with time windows and release dates. *Transportation Research Part E: Logistics and Transportation Review*, 135:101866, 2020.

[12] Walid Behiri. *Une méthodologie pour modéliser et optimiser la mutualisation du transport ferroviaire urbain de marchandises et de passagers*. PhD thesis, Université Paris-Est, 2017.

[13] Chendong Li, Chulwoo Park, Krishna R Pattipati, and David L Kleinman. Distributed algorithms for biobjective assignment problems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 5893–5898. IEEE, 2011.

[14] Shuvodeep De. Verification and validation in computational mechanics. *Mechanical Engineering, Preprints*, 2022020121, 2022.

[15] Mengpin Ge, Johannes Friedrich, and Leandro Vigna. Where do emissions come from? 4 charts explain greenhouse gas emissions by sector, December 2024. World Resources Institute. Accessed: 2025-05-14.

[16] Aleksandar Šobot and Sergej Gričar. An example of the transition to sustainable mobility in the austrian city of graz. *Sustainability*, 17(10):4324, 2025.

[17] United Nations Economic Commission for Europe. At cop28, unece and partners highlight need to decarbonize inland transport and how un tools and legal instruments can help, December 2023. Accessed: 2025-05-14.

[18] Xiaoqing Qu, Lei Zhong, Zhiqi Zeng, Hao Tu, and Xiaopeng Li. Automation and connectivity of electric vehicles: Energy boon or bane? *Cell Reports Physical Science*, 3:Article 101002, 2022.

[19] Alan Jenn. Emissions benefits of electric vehicles in uber and lyft ride-hailing services. *Nature Energy*, 5:520–525, 2020.

[20] Menia Mylonakou, Athanasios Chassiakos, Stylianos Karatzas, and Garyfallia Liappi. System dynamics analysis of the relationship between urban transportation and overall citizen satisfaction: A case study of patras city, greece. *Systems*, 11(3), 2023.

[21] MS Hossain, Laveet Kumar, MM Islam, and Jeyraj Selvaraj. A comprehensive review on the integration of electric vehicles for sustainable development. *Journal of Advanced Transportation*, 2022(1):3868388, 2022.

[22] Société de transport de Montréal. Rapport annuel 2024, 2024. Consulté le 2 juillet 2025.

[23] Research trends in combinatorial optimization: Bonn 2008. page 562, 2008.

[24] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[25] Maurizio Bielli, Alessandro Bielli, and Riccardo Rossi. Trends in models and algorithms for fleet management. *Procedia-Social and Behavioral Sciences*, 20:4–18, 2011.

[26] Wei Wang, Shuaian Wang, Lu Zhen, and Xiaobo Qu. EMS location-allocation problem under uncertainties. *Transportation Research Part E: Logistics and Transportation Review*, 168, 12 2022.

[27] Lu Zhen, Jingwen Wu, Fengli Chen, and Shuaian Wang. Traffic emergency vehicle deployment and dispatch under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 183, 3 2024.

[28] Ziling Zeng, Wen Yi, Shuaian Wang, and Xiaobo Qu. Emergency Vehicle Routing in Urban Road Networks with Multistakeholder Cooperation. *Journal of Transportation Engineering, Part A: Systems*, 147(10), 10 2021.

[29] D. Usanov, P. M.van de Ven, and R. D.van der Mei. Dispatching fire trucks under stochastic driving times. *Computers and Operations Research*, 114, 2 2020.

[30] Sarah Ibri, Mustapha Nourelfath, and Habiba Drias. A multi-agent approach for integrated emergency vehicle dispatching and covering problem. *Engineering Applications of Artificial Intelligence*, 25(3):554–565, 4 2012.

[31] Dimitris Bertsimas and Yeesian Ng. Robust and stochastic formulations for ambulance deployment and dispatch. *European Journal of Operational Research*, 279(2):557–571, 12 2019.

[32] Guangli Zhang, Rui Ma, Yunfeng Kong, Chenchen Lian, Hao Guo, and Shiyan Zhai. A multi-period capacitated facility location problem with maximum travel time and backup service for locating and sizing EMS stations. *Computational Urban Science*, 4(1), 12 2024.

[33] Rania Boujemaa, Aida Jebali, Sondes Hammami, and Angel Ruiz. Multi-period stochastic programming models for two-tiered emergency medical service system. *Computers and Operations Research*, 123, 11 2020.

[34] Hyoshin Park, Deion Waddell, and Ali Haghani. Online optimization with look-ahead for freeway emergency vehicle dispatching considering availability. *Transportation Research Part C: Emerging Technologies*, 109:95–116, 12 2019.

[35] Joseph Tassone and Salimur Choudhury. A Comprehensive Survey on the Ambulance Routing and Location Problems. 1 2020.

[36] Luca Talarico, Frank Meisel, and Kenneth Sörensen. Ambulance routing for disaster response with patient groups. *Computers and Operations Research*, 56:120–133, 2015.

[37] Y Frichi and F Jawab. A review and classification of ambulance deployment and redeployment models in emergencies 'A review and classification of ambulance deployment and redeployment models in emergencies'. Technical Report 1, 2024.

[38] Sebastian A. Rodriguez, Rodrigo A. De la Fuente, and Maichel M. Aguayo. A simulation-optimization approach for the facility location and vehicle assignment problem for firefighters using a loosely coupled spatio-temporal arrival process. *Computers and Industrial Engineering*, 157, 7 2021.

[39] Valérie Bélanger, Angel Ruiz, and Patrick Soriano and. Déploiement et redéploiement des véhicules ambulanciers dans la gestion d'un service préhospitalier d'urgence. *INFOR: Information Systems and Operational Research*, 50(1):1–30, 2012.

[40] Susan Hesse Owen and Mark S Daskin. Strategic facility location: A review. *European journal of operational research*, 111(3):423–447, 1998.

[41] Reza Zanjirani Farahani, Maryam SteadieSeifi, and Nasrin Asgari. Multiple criteria facility location problems: A survey. *Applied mathematical modelling*, 34(7):1689–1709, 2010.

[42] Kjetil Fagerholt, Trond AV Johnsen, and Haakon Lindstad. Fleet deployment in liner shipping: a case study. *Maritime Policy & Management*, 36(5):397–409, 2009.

[43] Qiang Meng, Shuaian Wang, Henrik Andersson, and Kristian Thun. Containership routing and scheduling in liner shipping: overview and future research directions. *Transportation Science*, 48(2):265–280, 2014.

[44] Anastassions N Perakis and Nikiforos Papadakis. Fleet deployment optimization models. part 1. *Maritime Policy & Management*, 14(2):127–144, 1987.

[45] Harry Benford. A simple approach to fleet deployment. *Maritime Policy and management*, 8(4):223–228, 1981.

[46] Chung-Cheng Lu, Shangyao Yan, Hui-Chieh Li, Ali Diabat, and Hsiao-Tung Wang. Optimal fleet deployment for electric vehicle sharing systems with the consideration of demand uncertainty. *Computers & Operations Research*, 135:105437, 2021.

[47] Dimitri P Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1):152–171, 1981.

[48] Bernard Fortz. Recherche opérationnelle et applications. Support de cours 53 pages, Université Libre de Bruxelles, Département d'informatique, 2012.

[49] Valérie Bélanger, Ettore Lanzarone, Angel Ruiz, and Patrick Soriano. *The ambulance relocation and dispatching problem.* CIRRELT Montréal, QC, Canada, 2015.

[50] Thomas Winter. *Online and real-time dispatching problems.* Citeseer, 2000.

[51] Verena Schmid. Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European journal of operational research*, 219(3):611–621, 2012.

[52] K Gkiotsalitis and EC Van Berkum. An exact method for the bus dispatching problem in rolling horizons. *Transportation Research Part C: Emerging Technologies*, 110:143–165, 2020.

[53] Leonardo Lamorgese and Carlo Mannino. An exact decomposition approach for the real-time train dispatching problem. *Operations Research*, 63(1):48–64, 2015.

[54] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[55] Burak Gülmez, Michael Emmerich, and Yingjie Fan. Multi-objective optimization for green delivery routing problems with flexible time windows. *Applied Artificial Intelligence*, 38(1):2325302, 2024.

[56] Jairo R Montoya-Torres, Julián López Franco, Santiago Nieto Isaza, Heriberto Felizzola Jiménez, and Nilson Herazo-Padilla. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129, 2015.

[57] Masoud Rabbani, Mina Akbarpour, Mahla Hosseini, and Hamed Farrokhi-Asl. A multi-depot vehicle routing problem with time windows and load balancing: A real world application. *International Journal of Supply and Operations Management*, 8(3):347–369, 9 2021.

[58] Chenn-Jung Huang, Kai-Wen Hu, Heng-Ming Chen, Hsiu-Hui Liao, Han Wen Tsai, and Sheng-Yuan Chien. An intelligent energy management mechanism for electric vehicles. *Applied Artificial Intelligence*, 30(2):125–152, 2016.

[59] Ying Lian, Flavien Lucas, and Kenneth Sörensen. The electric on-demand bus routing problem with partial charging and nonlinear function. *Transportation Research Part C: Emerging Technologies*, 157:104368, 2023.

[60] Nima Moradi, İhsan Sadati, and Bülent Çatay. Last mile delivery routing problem using autonomous electric vehicles. *Computers & Industrial Engineering*, 184:109552, 2023.

[61] Haoran He, Xiaoxiong Zhang, Jun Yang, Xiaolei Zhou, and Hao Yan. Optimization and Research on Army Vehicle Deployment in Emergency Situations. In *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pages 5314–5318. Institute of Electrical and Electronics Engineers Inc., 2024.

[62] Gan Chai, Jinde Cao, Wei Huang, and Jianhua Guo. Optimized traffic emergency resource scheduling using time varying rescue route travel time. *Neurocomputing*, 275:1567–1575, 1 2018.

[63] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.

[64] Arjun Paul, Ravi Shankar Kumar, Chayanika Rout, and Adrijit Goswami. Designing a multi-depot multi-period vehicle routing problem with time window: hybridization of tabu search and variable neighbourhood search algorithm. 2021.

[65] Chaitanya Ingle, Dev Bakliwal, Jayesh Jain, Preeyesh Singh, Preeti Kale, and Vaibhav Chhajed. Demand forecasting: Literature review on various methodologies. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2021.

[66] A Okay Akyuz, Mitat Uysal, Berna Atak Bulbul, and M Ozan Uysal. Ensemble approach for time series analysis in demand forecasting: Ensemble learning. In *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 7–12. IEEE, 2017.

[67] Jacob A Abernethy, Robert Schapire, and Umar Syed. Lexicographic optimization: Algorithms and stability. In *International Conference on Artificial Intelligence and Statistics*, pages 2503–2511. PMLR, 2024.

[68] H Isermann. Linear lexicographic optimization. *Operations-Research-Spektrum*, 4(4):223–228, 1982.

[69] George B Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity analysis of production and allocation*, 13:339–347, 1951.

[70] Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*. John Wiley & Sons, 1999.

[71] Ailsa H Land and Alison G Doig. An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pages 105–132. Springer, 2009.

[72] Richard Bellman. Terminal control, time lags, and dynamic programming. *Proceedings of the National Academy of Sciences*, 43(10):927–930, 1957.

[73] Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100, 1991.

[74] François Vanderbeck and Laurence A Wolsey. An exact algorithm for ip column generation. *Operations research letters*, 19(4):151–159, 1996.

[75] Alan K Mackworth. Consistency in networks of relations. *Artificial intelligence*, 8(1):99–118, 1977.

[76] Kaushik Kumar, Divya Zindani, and J Paulo Davim. *Optimizing Engineering Problems through Heuristic Techniques*. CRC Press, 2019.

[77] Rudolf Groner, Marina Groner, and Walter F Bischof. *Methods of heuristics*. Routledge, 2014.

[78] Heiner Müller-Merbach. Heuristics and their design: a survey. *European Journal of Operational Research*, 8(1):1–23, 1981.

[79] Ibrahim H Osman and Gilbert Laporte. Metaheuristics: A bibliography. *Annals of Operations research*, 63:511–623, 1996.

Appendix

Table 4: Detailed Results for Instance Family 1

(N,D,K)	Instances	Gap1 (%)	Gap2 (%)	Time1 (s)	Time2 (s)	Total time
(5 ,2 ,3)	instance 1	0,00%	0,00%	0,138	0,474	0,61
(5 ,2 ,3)	instance 2	0,00%	0,00%	0,061	0,177	0,24
(5 ,2 ,3)	instance 3	0,00%	0,00%	0,046	0,075	0,12
(5 ,2 ,3)	instance 4	0,00%	0,00%	0,171	0,354	0,53
(5 ,2 ,3)	instance 5	0,00%	0,00%	0,087	0,086	0,17
(5 ,2 ,3)	instance 6	0,00%	0,00%	0,065	0,238	0,30
(5 ,2 ,3)	instance 7	0,00%	0,00%	0,141	0,213	0,35
(5 ,2 ,3)	instance 8	0,00%	0,00%	0,047	0,189	0,24
(5 ,2 ,3)	instance 9	0,00%	0,00%	0,111	0,185	0,30
(5 ,2 ,3)	instance 10	0,00%	0,00%	0,067	0,324	0,39
(5 ,2 ,3)	instance 11	0,00%	0,00%	0,076	0,268	0,34
(5 ,2 ,3)	instance 12	0,00%	0,00%	0,086	0,212	0,30
(5 ,2 ,3)	instance 13	0,00%	0,00%	0,126	0,419	0,55
(5 ,2 ,3)	instance 14	0,00%	0,00%	0,116	0,143	0,26
(5 ,2 ,3)	instance 15	0,00%	0,00%	0,145	0,255	0,40
(5 ,2 ,3)	instance 16	0,00%	0,00%	0,118	0,415	0,53
(5 ,2 ,3)	instance 17	0,00%	0,00%	0,114	0,233	0,35
(5 ,2 ,3)	instance 18	0,00%	0,00%	0,097	0,231	0,33
(5 ,2 ,3)	instance 19	0,00%	0,00%	0,066	0,293	0,36
(5 ,2 ,3)	instance 20	0,00%	0,00%	0,067	0,156	0,22
(5 ,2 ,3)	instance 21	0,00%	0,00%	0,067	0,226	0,29
(5 ,2 ,3)	instance 22	0,00%	0,00%	0,096	0,37	0,47
(5 ,2 ,3)	instance 23	0,00%	0,00%	0,124	0,271	0,40
(5 ,2 ,3)	instance 24	0,00%	0,00%	0,072	0,201	0,27
(5 ,2 ,3)	instance 25	0,00%	0,00%	0,065	0,167	0,23
(5 ,2 ,3)	Average	0,00%	0,00%	0,0948	0,2470	0,3418
	Ecart type	-%	-%	0,0339	0,1008	-

Table 5: Detailed Results for Instance Family 2

(N,D,K)	Instances	Gap1 (%)	Gap2 (%)	Time1 (s)	Time2 (s)	Total time
(6, 2, 4)	instance1	0,00%	0,00%	0,1630	0,2810	0,44
(6, 2, 4)	instance2	0,00%	0,00%	0,1560	0,3580	0,51
(6, 2, 4)	instance3	0,00%	0,00%	0,1450	0,5300	0,68
(6, 2, 4)	instance4	0,00%	0,00%	0,0920	0,5820	0,67
(6, 2, 4)	instance5	0,00%	0,00%	0,1250	0,5110	0,64
(6, 2, 4)	instance6	0,00%	0,00%	0,1160	0,3570	0,47
(6, 2, 4)	instance7	0,00%	0,00%	0,2010	0,6360	0,84
(6, 2, 4)	instance8	0,00%	0,00%	0,4400	0,9150	1,36
(6, 2, 4)	instance9	0,00%	0,00%	0,0770	0,2870	0,36
(6, 2, 4)	instance10	0,00%	0,00%	0,1240	0,2660	0,39
(6, 2, 4)	instance11	0,00%	0,00%	0,1620	0,3700	0,53
(6, 2, 4)	instance12	0,00%	0,00%	0,1250	0,2720	0,40
(6, 2, 4)	instance13	0,00%	0,00%	0,1950	0,4870	0,68
(6, 2, 4)	instance14	0,00%	0,00%	0,1160	0,2960	0,41
(6, 2, 4)	instance15	0,00%	0,00%	0,0830	0,3410	0,42
(6, 2, 4)	instance16	0,00%	0,00%	0,1110	0,3180	0,43
(6, 2, 4)	instance17	0,00%	0,00%	0,0520	0,1580	0,21
(6, 2, 4)	instance18	0,00%	0,00%	0,1920	0,3340	0,53
(6, 2, 4)	instance19	0,00%	0,00%	0,1790	0,2930	0,47
(6, 2, 4)	instance20	0,00%	0,00%	0,1480	0,3710	0,52
(6, 2, 4)	instance21	0,00%	0,00%	0,0920	0,2150	0,31
(6, 2, 4)	instance22	0,00%	0,00%	0,1120	0,3870	0,50
(6, 2, 4)	instance23	0,00%	0,00%	0,1150	0,2570	0,37
(6, 2, 4)	instance24	0,00%	0,00%	0,1520	0,3680	0,52
(6, 2, 4)	instance25	0,00%	0,00%	0,0600	0,1660	0,23
(6, 2, 4)	Average	0,00%	0,00%	0,1413	0,3742	0,5156
	Ecart type	-%	-%	0,0743	0,1642	-

Table 6: Detailed Results for Instance Family 3

(N,D,K)	Instances	Gap1 (%)	Gap2 (%)	Time1 (s)	Time2 (s)	Total time
(7, 3, 5)	instance1	0,00%	0,00%	0,4850	1,2010	1,69
(7, 3, 5)	instance2	0,00%	0,00%	0,2150	1,9180	2,13
(7, 3, 5)	instance3	0,00%	0,00%	0,1690	1,6990	1,87
(7, 3, 5)	instance4	0,00%	0,00%	0,2010	0,3430	0,54
(7, 3, 5)	instance5	0,00%	0,00%	0,2220	1,8580	2,08
(7, 3, 5)	instance6	0,00%	0,00%	0,1540	0,6740	0,83
(7, 3, 5)	instance7	0,00%	0,00%	0,3240	1,5650	1,89
(7, 3, 5)	instance8	0,00%	0,00%	0,3540	1,9370	2,29
(7, 3, 5)	instance9	0,00%	0,00%	0,3460	2,1690	2,52
(7, 3, 5)	instance10	0,00%	0,00%	0,1870	1,0240	1,21
(7, 3, 5)	instance11	0,00%	0,00%	0,2880	1,4510	1,74
(7, 3, 5)	instance12	0,00%	0,00%	0,1450	0,5870	0,73
(7, 3, 5)	instance13	0,00%	0,00%	0,2990	0,4590	0,76
(7, 3, 5)	instance14	0,00%	0,00%	0,2700	0,6760	0,95
(7, 3, 5)	instance15	0,00%	0,00%	0,1400	0,5510	0,69
(7, 3, 5)	instance16	0,00%	0,00%	0,2250	1,1970	1,42
(7, 3, 5)	instance17	0,00%	0,00%	0,2020	1,0820	1,28
(7, 3, 5)	instance18	0,00%	0,00%	0,1800	0,9330	1,11
(7, 3, 5)	instance19	0,00%	0,00%	0,2090	0,5030	0,71
(7, 3, 5)	instance20	0,00%	0,00%	0,1350	0,9760	1,11
(7, 3, 5)	instance21	0,00%	0,00%	0,1090	1,1230	1,23
(7, 3, 5)	instance22	0,00%	0,00%	0,1880	0,9980	1,19
(7, 3, 5)	instance23	0,00%	0,00%	0,1440	1,0780	1,22
(7, 3, 5)	instance24	0,00%	0,00%	0,1240	0,9540	1,08
(7, 3, 5)	instance25	0,00%	0,00%	0,2600	2,3940	2,65
(7, 3, 5)	Average	0,00%	0,00%	0,2230	1,1740	1,3970
	Ecart type	-%	-%	0,0884	0,5661	-

Table 7: Detailed Results for Instance Family 4

(N,D,K)	Instances	Gap1 (%)	Gap2 (%)	Time1 (s)	Time2 (s)	Total time
(8, 3, 5)	instance1	0,00%	0,00%	0,392	2,177	2,57
(8, 3, 5)	instance2	0,00%	0,00%	0,344	2,514	2,86
(8, 3, 5)	instance3	0,00%	0,00%	0,6	3,861	4,46
(8, 3, 5)	instance4	0,00%	0,00%	0,277	0,857	1,13
(8, 3, 5)	instance5	0,00%	0,00%	0,396	3,263	3,66
(8, 3, 5)	instance6	0,00%	0,00%	1,003	15,657	16,66
(8, 3, 5)	instance7	0,00%	0,00%	0,19	2,886	3,08
(8, 3, 5)	instance8	0,00%	0,00%	0,313	3,424	3,74
(8, 3, 5)	instance9	0,00%	0,00%	0,27	0,642	0,91
(8, 3, 5)	instance10	0,00%	0,00%	1,318	14,934	16,25
(8, 3, 5)	instance11	0,00%	0,00%	0,27	0,879	1,15
(8, 3, 5)	instance12	0,00%	0,00%	0,506	4,372	4,88
(8, 3, 5)	instance13	0,00%	0,00%	0,748	5,534	6,28
(8, 3, 5)	instance14	0,00%	0,00%	0,314	1,209	1,52
(8, 3, 5)	instance15	0,00%	0,00%	0,414	3,963	4,38
(8, 3, 5)	instance16	0,00%	0,00%	0,316	2,018	2,33
(8, 3, 5)	instance17	0,00%	0,00%	0,726	4,191	4,92
(8, 3, 5)	instance18	0,00%	0,00%	0,502	8,755	9,26
(8, 3, 5)	instance19	0,00%	0,00%	0,405	2,763	3,17
(8, 3, 5)	instance20	0,00%	0,00%	1,414	17,584	19,00
(8, 3, 5)	instance21	0,00%	0,00%	0,316	0,892	1,21
(8, 3, 5)	instance22	0,00%	0,00%	0,398	1,524	1,92
(8, 3, 5)	instance23	0,00%	0,00%	0,523	2,102	2,63
(8, 3, 5)	instance24	0,00%	0,00%	1,433	23,477	24,91
(8, 3, 5)	instance25	0,00%	0,00%	0,34	2,471	2,81
(8, 3, 5)	Average	0,00%	0,00%	0,54912	5,27796	5,8271
	Ecart type	-%	-%	0,3639	6,0455	-

Table 8: Detailed Results for Instance Family 5

(N,D,K)	Instances	Gap1 (%)	Gap2 (%)	Time1 (s)	Time2 (s)	Total time
(9, 3, 6)	instance1	0,00%	0,00%	1,38	9,981	11,36
(9, 3, 6)	instance2	0,00%	0,00%	0,817	3,222	4,04
(9, 3, 6)	instance3	0,00%	0,00%	1,144	9,877	11,02
(9, 3, 6)	instance4	0,00%	0,00%	1,828	45,745	47,57
(9, 3, 6)	instance5	0,00%	0,00%	0,539	3,406	3,95
(9, 3, 6)	instance6	0,00%	0,00%	0,519	14,446	14,97
(9, 3, 6)	instance7	0,00%	0,00%	1,03	10,944	11,97
(9, 3, 6)	instance8	0,00%	0,00%	1,361	18,082	19,44
(9, 3, 6)	instance9	0,00%	0,00%	2,222	5,744	7,97
(9, 3, 6)	instance10	0,00%	0,00%	0,713	2,638	3,35
(9, 3, 6)	instance11	0,00%	0,00%	1,331	2,033	3,36
(9, 3, 6)	instance12	0,00%	0,00%	0,57	2,558	3,13
(9, 3, 6)	instance13	0,00%	0,00%	0,656	2,784	3,44
(9, 3, 6)	instance14	0,00%	0,00%	1,598	9,428	11,03
(9, 3, 6)	instance15	0,00%	0,00%	0,845	5,008	5,85
(9, 3, 6)	instance16	0,00%	0,00%	0,933	5,65	6,58
(9, 3, 6)	instance17	0,00%	0,00%	1,695	6,862	8,56
(9, 3, 6)	instance18	0,00%	0,00%	1,228	2,28	3,51
(9, 3, 6)	instance19	0,00%	0,00%	3,912	31,836	35,75
(9, 3, 6)	instance20	0,00%	0,00%	0,229	5,519	5,75
(9, 3, 6)	instance21	0,00%	0,00%	1,348	62,727	64,08
(9, 3, 6)	instance22	0,00%	0,00%	1,348	29,348	30,70
(9, 3, 6)	instance23	0,00%	0,00%	0,696	6,154	6,85
(9, 3, 6)	instance24	0,00%	0,00%	4,812	44,819	49,63
(9, 3, 6)	instance25	0,00%	0,00%	1,023	7,606	8,63
(9, 3, 6)	Average	0,00%	0,00%	1,35108	13,94788	15,2990
	Ecart type	-%	-%	1,0253	16,2078	-

Table 9: Detailed Results for Instance Family 6

(N,D,K)	Instances	Gap1 (%)	Gap2 (%)	Time1 (s)	Time2 (s)	Total time
(10, 4, 7)	instance1	0,00%	0,00%	204,278	135,85	340,13
(10, 4, 7)	instance2	0,00%	0,00%	1,309	136,271	137,58
(10, 4, 7)	instance3	0,00%	0,00%	27,424	58,879	86,30
(10, 4, 7)	instance4	0,00%	20,02%	51,285	548,738	600,02
(10, 4, 7)	instance5	0,00%	0,00%	2,993	4,619	7,61
(10, 4, 7)	instance6	0,00%	0,00%	1,487	201,306	202,79
(10, 4, 7)	instance7	0,00%	0,00%	1,886	352,564	354,45
(10, 4, 7)	instance8	0,00%	0,00%	11,02	236,281	247,30
(10, 4, 7)	instance9	0,00%	0,00%	21,368	202,342	223,71
(10, 4, 7)	instance10	0,00%	16,68%	101,169	498,856	600,03
(10, 4, 7)	instance11	0,00%	0,00%	50,127	539,053	589,18
(10, 4, 7)	instance12	0,00%	29,39%	18,184	582,195	600,38
(10, 4, 7)	instance13	0,00%	12,21%	7,138	592,893	600,03
(10, 4, 7)	instance14	0,00%	3,29%	107,043	492,987	600,03
(10, 4, 7)	instance15	0,00%	0,00%	3,18	81,237	84,42
(10, 4, 7)	instance16	0,00%	6,30%	4,623	595,391	600,01
(10, 4, 7)	instance17	0,00%	19,41%	188,381	411,627	600,01
(10, 4, 7)	instance18	0,00%	0,00%	1,65	259,277	260,93
(10, 4, 7)	instance19	0,00%	19,15%	157,676	442,35	600,03
(10, 4, 7)	instance20	0,00%	19,83%	252,981	347,035	600,02
(10, 4, 7)	instance21	0,00%	10,60%	7,711	592,304	600,02
(10, 4, 7)	instance22	0,00%	0,00%	1,211	438,33	439,54
(10, 4, 7)	instance23	0,00%	0,00%	41,752	42,423	84,18
(10, 4, 7)	instance24	0,00%	0,00%	1,932	26,803	28,74
(10, 4, 7)	instance25	0,00%	8,50%	18,181	581,843	600,02
(10, 4, 7)	Average	0,00%	15,03%	51,43956	336,05816	387,4977
	Ecart type	-%	7,60%	73,7858	209,2313	-

Table 10: Detailed Results for Instance Family 7

(N,D,K)	Instances	Gap1 (%)	Gap2 (%)	Time1 (s)	Time2 (s)	Total time
(11, 4, 7)	instance1	0,00%	36,23%	425,738	174,273	600,01
(11, 4, 7)	instance2	0,00%	0,00%	16,028	197,135	213,16
(11, 4, 7)	instance3	0,00%	24,15%	58,623	541,405	600,03
(11, 4, 7)	instance4	0,00%	18,98%	38,394	561,63	600,02
(11, 4, 7)	instance5	0,00%	0,00%	2,584	73,364	75,95
(11, 4, 7)	instance6	0,00%	26,63%	168,714	431,317	600,03
(11, 4, 7)	instance7	0,00%	28,41%	363,804	236,214	600,02
(11, 4, 7)	instance8	0,00%	1,44%	28,642	571,363	600,01
(11, 4, 7)	instance9	0,00%	25,30%	276,988	323,024	600,01
(11, 4, 7)	instance10	0,00%	24,10%	334,664	265,344	600,01
(11, 4, 7)	instance11	0,00%	0,00%	2,718	278,125	280,84
(11, 4, 7)	instance12	0,00%	23,11%	30,986	569,021	600,01
(11, 4, 7)	instance13	0,00%	0,00%	17,12	16,759	33,88
(11, 4, 7)	instance14	0,00%	0,00%	5,5	408,08	413,58
(11, 4, 7)	instance15	0,00%	26,04%	11,543	588,466	600,01
(11, 4, 7)	instance16	0,00%	29,34%	46,905	553,121	600,03
(11, 4, 7)	instance17	0,00%	31,40%	31,324	568,698	600,02
(11, 4, 7)	instance18	0,00%	0,00%	46,593	340,905	387,50
(11, 4, 7)	instance19	0,00%	29,42%	186,821	413,199	600,02
(11, 4, 7)	instance20	0,00%	26,61%	362,931	237,085	600,02
(11, 4, 7)	instance21	0,00%	22,82%	152,308	447,709	600,02
(11, 4, 7)	instance22	0,00%	6,24%	5,873	594,14	600,01
(11, 4, 7)	instance23	0,00%	0,00%	56,547	9,071	65,62
(11, 4, 7)	instance24	33,33%	-%	600,03	-	600,03
(11, 4, 7)	instance25	0,00%	15,64%	113,807	486,211	600,02
(11, 4, 7)	Average	33,33%	23,29%	135,4074	370,2357917	505,6432
	Ecart type	0	8,71%	164,7250	188,0220	-

Table 11: Detailed Results for Instance Family 8

(N,D,K)	Instances	Gap1 (%)	Gap2 (%)	Time1 (s)	Time2 (s)	Total time
(12, 4, 7)	instance1	0,00%	37,30%	53,932	546,093	600,03
(12, 4, 7)	instance2	33,33%	-%	600,028	-	600,03
(12, 4, 7)	instance3	0,00%	36,68%	97,439	502,611	600,05
(12, 4, 7)	instance4	0,00%	32,66%	231,322	368,707	600,03
(12, 4, 7)	instance5	0,00%	-%	600,031	-	600,03
(12, 4, 7)	instance6	0,00%	29,61%	52,521	547,508	600,03
(12, 4, 7)	instance7	0,00%	31,80%	518,75	81,281	600,03
(12, 4, 7)	instance8	0,00%	30,39%	56,073	543,968	600,04
(12, 4, 7)	instance9	0,00%	20,65%	48,888	551,134	600,02
(12, 4, 7)	instance10	0,00%	34,03%	204,127	395,881	600,01
(12, 4, 7)	instance11	0,00%	6,78%	27,399	572,629	600,03
(12, 4, 7)	instance12	0,00%	15,26%	18,876	581,154	600,03
(12, 4, 7)	instance13	0,00%	29,01%	145,329	454,681	600,01
(12, 4, 7)	instance14	0,00%	31,41%	315,666	284,352	600,02
(12, 4, 7)	instance15	0,00%	24,86%	30,882	569,155	600,04
(12, 4, 7)	instance16	0,00%	0,00%	23,99	576,047	600,04
(12, 4, 7)	instance17	33,33%	-%	600,032	-	600,03
(12, 4, 7)	instance18	0,00%	32,62%	56,981	543,051	600,03
(12, 4, 7)	instance19	33,33%	-%	600,04	-	600,04
(12, 4, 7)	instance20	33,33%	-%	600,025	-	600,03
(12, 4, 7)	instance21	0,00%	27,39%	191,05	408,97	600,02
(12, 4, 7)	instance22	0,00%	38,86%	24,692	575,32	600,01
(12, 4, 7)	instance23	0,00%	-%	600,039	-	600,04
(12, 4, 7)	instance24	0,00%	-%	600,27	-	600,27
(12, 4, 7)	instance25	0,00%	-%	600,046	-	600,05
(12, 4, 7)	Average	33,33%	28,71%	275,93712	476,6201176	752,5572
	Ecart type	0,00%	8,42%	252,1420	134,5932	-