



الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
المدرسة الوطنية العليا للتكنولوجيات المتقدمة
National Higher School of Advanced Technologies



Department of Logistics and Transport Engineering

Final Year Project to Obtain the Diploma of

Engineering

- Field -

Transport Engineering

- Specialization -

Transport Engineering

- Thesis Title -

Optimization of Fresh Product Deliveries

Case Study of Celia, Lactalis Group

Presented by

Nada BOUKHALFA

Jury Members:

TOUMI Manel
BADJARA Mohamed El Amine
BOUZID M. Cherif

MCB ENSTA
MAA ENSTA
MCA ENSTA

President
Supervisor
Examiner

Algiers, September 25, 2025

Academic Year: 2024-2025

DEDICATION

*To my father,
whose wisdom and love shaped who I am today,
though you are no longer with us,
your lessons and memory
continue to guide every step I take,*

*to my mother,
whose boundless love, sacrifice, and unwavering faith
have been my anchor through life's storms,
my source of inspiration and strength,*

*to my beloved siblings
Djoumana, Hind, Imane, and Amine,
who have been my constant companions and support,*

*and to my dear friends
Thilleli, Meriem, Ines, Amina, and Zahra,
whose friendship and encouragement
have enriched my journey.*

This humble work is dedicated to you all with all my love.

Acknowledgments

First and foremost, all praises and thanks to **Allah** for the strength, the blessing, and the guidance through this journey.

I extend my sincere appreciation to my supervisor **Mr. Badjara Mohamed El Amine**, for his invaluable mentorship and support throughout this work.

My heartfelt thanks go to my supervisor at CELIA **Mr. Slimani Abdelkarim**, for accepting this theme and for his assistance and expertise.

I am deeply grateful to **Mr. Zerrouki Bilel**, Customer Service Responsible for his help, support, and availability during my internship period.

My sincere thanks go to the jury members **Mme Toumi Manel** and **Mr. Bouzid M. Cherif** for agreeing to evaluate my work and for their valuable feedback.

I would like to express my gratitude to all my professors of the **GLT Department** for their commitment and the knowledge they have shared with us during the academic journey.

Finally, I would like to thank all those who have supported me, directly or indirectly, throughout the completion of this work.

ملخص

يطور مشروع التخرج نظاماً لوجستياً مزدوج المكونات لتحسين توزيع المنتجات الغذائية الطازجة، يتناول تنبؤ الطلب وتوجيه المركبات كمشكلتين منفصلتين. أجريت الدراسة بالشراكة مع aileC، لمعالجة التحديات العملية عبر حلول قائمة على البيانات. تشمل المنهجية: تنبؤ الطلب باستخدام البيانات التاريخية، مقارنة الطرق التقليدية بنهج التعلم الآلي (tsooBGX)، وتحسين التوجيه من خلال نمذجة رياضية محلولة ببرنامج XELPC GOLI MBI. حقق tsooBGX تحسينات معتبرة، بينما وفر نموذج التوجيه تخفيضات في التكاليف وتحسيناً لاستخدام الأسطول. يدمج تطبيق ويب المكونين، مما يوفر أدوات للتخطيط الاستراتيجي وقدرات للتحسين العملي.

الكلمات المفتاحية: المنتجات الغذائية الطازجة، التنبؤ بالطلب، مسارات المركبات، tsooBGX، XELPC.

Abstract

This final year project develops a dual-component logistics system for fresh food distribution, addressing demand forecasting and vehicle routing as separate problems. Conducted with Celia, it tackles operational challenges through data-driven solutions. The methodology includes demand forecasting with historical data, comparing traditional methods to Machine Learning (XGBoost), and vehicle routing optimization via mathematical modeling solved with IBM ILOG CPLEX. XGBoost yielded substantial improvements, while the VRP model achieved cost reductions and fleet efficiency. A web application integrates both components, providing strategic planning tools and practical optimization capabilities.

Keywords: Demand Forecasting, Vehicle Routing Problem, XGBoost, CPLEX.

Résumé

Ce projet de fin d'études développe un système logistique à double composante pour la distribution de produits frais, abordant la prévision de la demande et le routage de véhicules comme problèmes distincts. Réalisé avec Celia, il répond aux défis opérationnels par des solutions fondées sur les données. La méthodologie inclut : prévision de la demande à partir de données historiques, comparaison des méthodes traditionnelles avec une approche d'apprentissage automatique (XGBoost), et optimisation du routage par modélisation mathématique résolue avec IBM ILOG CPLEX. XGBoost a montré des gains notables, tandis que le modèle VRP a réduit les coûts et a amélioré l'usage de la flotte. Une application web intègre les deux volets, offrant des outils de planification stratégique et d'optimisation opérationnelle.

Mots-clés : Prévision de la Demande, Routage de Véhicules, XGBoost, CPLEX.

Table of Contents

ملخص	i
Abstract	i
Résumé	i
General Introduction	1
Chapter 1: Presentation of the Company Celia	4
Introduction	4
1.1 Presentation of Lactalis Group	4
1.2 Presentation of Celia Algérie	5
1.2.1 Product Categories and Classification	5
Dry Products	5
Fresh Products	6
1.3 Organizational Architecture	7
1.4 Logistics Distinctions Between Fresh and Dry Product Lines	8
1.5 Fresh Products Distribution System	8
1.5.1 Depot Operations – Mars Logistique	8
1.5.2 Client Network and Fleet Configuration	9
1.5.3 Daily Transfers and Cold Chain Integrity	9
1.5.4 Order Consolidation and Palletization	9
1.5.5 Order Validation and Distribution Scheduling	9
1.5.6 Delivery Waves and Execution	10
1.5.7 Monitoring and Compliance	10
1.6 The Company’s Logistics Challenges	10
1.7 Problem Statement	11
Conclusion	11
Chapter 2: Overview of Time Series and Forecasting Methods	14
Introduction	14
2.1 Generalities About Forecasting	14
2.1.1 Methodological Classification of Forecasting Approaches	14
2.2 Time Series Theory	15
2.2.1 Components of a Time Series	15

2.2.2	Time Series Models: Additive vs. Multiplicative	16
2.2.3	Deterministic vs Stochastic Time Series	17
2.2.4	Stationarity in Time Series	18
2.2.5	Classification of Demand Patterns	18
2.3	Forecasting Approaches	20
2.3.1	Classical Forecasting Models	20
2.3.2	Limitations of Classical Methods	24
2.3.3	Artificial Intelligence-Based Approaches	24
2.3.4	Advantages of Gradient Boosting in Time Series Forecasting	30
2.3.5	Overfitting Prevention and Hyperparameter Optimization	31
2.3.6	Evaluation Metrics	32
	Conclusion	33

Chapter 3: The Vehicle Routing Problem: Foundations and Solution

	Approaches	35
	Introduction	35
3.1	Foundations of Operations Research and Combinatorial Optimization	35
3.1.1	Operations Research (OR)	35
3.1.2	Combinatorial Optimization (CO)	36
3.1.3	Computational Complexity	37
3.1.4	Complexity of Problems	37
3.2	Vehicle Routing Problem (VRP)	38
3.2.1	Mathematical Modeling of the Vehicle Routing Problem	39
3.2.2	Taxonomy of VRP Variants	41
3.2.3	Vehicle Routing Problem Variants	44
3.2.4	Multi-Trip Vehicle Routing Problem (MTVRP)	46
3.3	Solution Approaches for VRP	46
3.3.1	Theoretical Complexity of VRP Variants	46
3.3.2	Exact vs. Approximate Methods for Solving VRP	46
	Conclusion	50

Chapter 4: Exploratory Data Analysis and Forecasting Model Implementation

	Introduction	52
	Introduction	52
4.1	Dataset Description	52
4.2	Exploratory Data Analysis (EDA)	53
4.2.1	Data Quality Assessment	53
4.2.2	Descriptive Statistics of Order Quantities	54
4.2.3	Distribution of Order Quantities	54
4.2.4	Outliers Detection	55
4.2.5	Order Volume by City	57
4.2.6	Temporal Patterns and Demand Trends	58
4.2.7	Preprocessing for Time Series Consistency	60
4.2.8	Sparsity of Demand and Zero-Dominated Behavior	60

4.2.9	Analysis of Zero-Order Patterns by City	61
4.2.10	Customer Segmentation Based on Ordering Patterns	62
4.2.11	Segmentation Results	63
4.2.12	Visualizing Customer Ordering Behavior	63
4.2.13	Zero-Order Day Analysis by Segment	63
4.2.14	Stationarity Analysis	65
4.3	Demand Pattern Classification Analysis	66
4.3.1	Results	66
4.3.2	Key Findings	66
4.3.3	Implications	67
4.4	Forecasting Models and Results	67
4.5	Benchmark Model: Croston’s Method	67
4.5.1	Implementation Steps	68
4.5.2	Croston Method Results and Analysis	68
4.6	TSB Method (Teunter–Syntetos–Babai)	70
4.6.1	Implementaion steps:	70
4.6.2	TSB Method Results and Analysis	71
4.7	Towards Advanced Forecasting: Ensemble Learning models	73
4.7.1	Statistical Predictability Assessment of Demand Components	73
4.7.2	Feature Engineering for Positive Demand Prediction	74
4.7.3	XGBoost Implementation:	74
	Conclusion	78

Chapter 5:Implementation of the Mathematical VRP Model and Solution Approaches **80**

	Introduction	80
5.1	Problem Statement	80
5.2	Mathematical Formulation	81
5.2.1	Sets and Parameters	81
5.2.2	Decision Variables	82
5.2.3	Objective Function and Cost Modeling	83
5.2.4	Objective Function: Min–Max and Linearization	83
5.2.5	Constraints	83
5.3	Solution Methodology Overview	88
5.3.1	CPLEX Implementation Framework	88
5.3.2	Experimental Design and Scenario Testing	90
5.3.3	Results Analysis and Discussion	91
5.3.4	Data Preparation Methodology	92
5.3.5	Computational Complexity Analysis	95
5.3.6	Comparative Analysis Framework	96
5.4	Model Limitations	97
5.4.1	Unmet Demand and Postponement	97
5.4.2	Second Route Utilization and Overnight Stays	98
5.5	App Web Architecture	98

5.6	Technology Stack Overview	98
5.6.1	Core Architectural Components	99
5.7	Implementation Details	100
5.7.1	Authentication System	100
5.7.2	Dashboard Implementation	100
5.7.3	VRP Optimization Implementation	102
5.7.4	Advanced Analytics Implementation	104
5.7.5	System Management and Monitoring	106
	Conclusion	107
	General Conclusion	109

List of Tables

1.1	Organizational Structure of Celia Algérie	7
1.2	Organizational Structure of Celia Algérie (continued)	8
1.3	Fleet Configuration by Vehicle Type	9
3.4	Classification of VRP Variants by Key Features	41
4.5	Missing Values per Column	53
4.6	Summary Statistics for Non-Zero Order Quantities	54
4.7	Comparison of Distribution Fits Based on Log-Likelihood, AIC, and BIC	55
4.8	Top 5 Order-Level Outliers Detected Using IQR	57
4.9	Top 10 Cities by Total Order Volume (Pallets)	58
4.10	Top 5 Cities with the Lowest Percentage of Zero Orders (Most Active)	61
4.11	Top 5 Cities with the Highest Percentage of Zero Orders (Least Active)	62
4.12	Customer Segmentation Summary	63
4.13	Summary of Stationarity Test Results (Before and After Differencing)	65
4.14	Demand Pattern Classification Results by City	66
4.15	Pattern Distribution Summary	67
4.16	Croston Method Performance Metrics by City (Ranked by RMSE)	68
4.17	TSB Method Performance Metrics by City ($\alpha=0.15$, $\beta=0.05$)	71
4.18	Performance Comparison: Croston vs. TSB Methods	71
4.19	Predictability Assessment Results by Component	73
4.20	Engineered Features for Demand Forecasting	74
4.21	XGBoost Hyperparameter Search Space	75
4.22	XGBoost Method Performance Metrics by City	77
4.23	Method Performance Comparison: XGBoost vs. Traditional Approaches	78
5.24	Core Performance Metrics for Test Scenarios	91
5.25	Operational Details and Special Cases	91
5.26	Complete OPL Data File Specification: Sets, Parameters, and Matrices	94
5.27	VRP Optimization Results: Cost and Operational Efficiency	97
5.28	Frontend Technology Stack	98
5.29	Backend, Optimization, and Data Technology Stack	99
30	CPLEX Solution Summary for Test Scenario 4	112
31	Preprocessing and Problem Reduction	112
32	Cut Generation Statistics	113
33	Vehicle Route Assignments and Performance	114
34	Optimization Performance Indicators	115

List of Figures

1.1	Logo of Group Lactalis [1]	5
1.2	Logo of Celia Algérie [2]	5
1.3	Example of Celia’s fresh products. Source: Internal company document	6
1.4	Fresh product distribution workflow from order intake to final delivery	10
2.5	Time series decomposition showing trend, seasonal, and random components in temporal data analysis [3]	16
2.6	Comparison of additive versus multiplicative time series models [4] . .	17
2.7	Classification of demand patterns based on variability in timing and quantity [5].	19
2.8	Schema of supervised learning [6]	26
2.9	Illustration of a decision tree for Regression [7]	26
2.10	Comparison of Bagging and Boosting methodologies [8]	27
2.11	Structure of the XGBoost model [9]	28
3.12	Concept of study by OR model [10].	35
3.13	Classical optimization methods [11].	36
3.14	Graph representation of the combinatorial optimization problem of finding a Minimum Spanning Tree (MST)[12].	37
3.15	Relationship between complexity classes P, NP, NP-Complete, and NP-Hard. Adapted from [13].	38
3.16	Graphical representation of a typical Vehicle Routing Problem (VRP) instance [14].	39
3.17	Decision framework for Vehicle Routing Problem variant selection based on problem characteristics and constraints [15]	43
3.18	Overall percentage of each VRP variant during the study period (2010-2020) [16]	44
3.19	Classification of algorithms for the VRP [17]	47
3.20	Branch and Price Approach [11]	48
3.21	Evolution of Branch and X algorithms. [11]	49
3.22	Example of a local and global optimum [18]	49
4.23	Initial overview of the dataset: structure, temporal coverage, and cites.	53
4.24	Distribution of Non-Zero Order Quantities	54
4.25	Boxplot of daily order volumes highlighting outliers.	56
4.26	Order volume distribution for top 10 customers using boxplot	56
4.27	Top 10 cities by total order volume (in pallets).	58
4.28	Total order volume per year.	59

4.29	Average monthly order volume (seasonal patterns).	59
4.30	Average daily order volume by weekday.	60
4.31	Distribution of Daily Order Quantities (Including Zeros)	61
4.32	Pairplot of ordering behavior variables by customer category.	63
4.33	Time series decomposition of daily orders from active customers (2020–2025).	65
4.34	Demand Pattern Classification Analysis	67
4.35	Croston method forecasts vs. actual demand for representative cities with varying demand patterns	69
4.36	TSB method forecasts vs. actual demand	72
4.37	XGBoost Demand Prediction Results for Three Representative Cities	77
5.38	Data extraction and transformation workflow	93
5.39	CPLEX Optimization Results and Route Analysis	95
5.40	Login Interface	100
5.41	Real-time dashboard navigation with connectivity status monitoring	100
5.42	Dashboard System Components - Analytics and Customer Analysis	101
5.43	Distribution Analysis and Temporal Patterns	102
5.44	Customer volume ranking and risk assessment matrix	102
5.45	Table of selected customers	102
5.46	Vehicle Selection and Automatic Dynamic Subsets Creation	103
5.47	VRP Optimization Process: Subset Generation, Execution, and Results	104
5.48	Objective value of the most recent optimization process	104
5.49	Customer Clustering Visualization and Analysis	105
5.50	Temporal Analysis and Predictive Analytics	105
5.51	Order Processing System with Analytics Integration	106
5.52	System Monitoring and Performance Analytics	107
53	Data file	111
54	CPLEX Output1	112
55	CPLEX Output2	113
56	CPLEX Solution Convergence and Objective Value Evolution	113
57	CPLEX Split Delivery and Demand Satisfaction Verification	114
58	Overview of custome’s dataset using in VRP	115
59	Overview of cost matrix	115
60	Historical data overview	116

List of Algorithms

1	Croston's Method for Intermittent Demand Forecasting	22
2	TSB (Teunter–Syntetos–Babai) Method	24
3	General XGBoost Algorithm	30

List of Abbreviations

ADI	Average Demand Interval
AI	Artificial Intelligence
API	Application Programming Interface
CO	Combinatorial Optimization
CV ²	Squared Coefficient of Variation
GBT	Gradient Boosted Trees
GPU	Graphics Processing Unit
HF-OVRP-SD-TW-MT	Heterogeneous Fleet Open Vehicle Routing Problem with Split Deliveries, Time Windows, and Multi-Trips
JSON	JavaScript Object Notation
JWT	JSON Web Token
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
MILP	Mixed Integer Linear Programming
ML	Machine Learning
MTZ	Miller-Tucker-Zemlin
OR	Operational Research
TSB	Teunter-Syntetos-Babai
XGBoost	eXtreme Gradient Boosting

General Introduction

In an economic environment marked by high volatility and intensified competition, organizations are increasingly required to make complex and rapid decisions to manage their supply chains and logistics operations effectively. In this context, forecasting and optimization have emerged as two essential pillars of data-driven decision-making. Forecasting enables anticipation of future demand, facilitating proactive resource allocation and strategic planning, while optimization ensures the efficient execution of operations through well-structured delivery plans that minimize costs and respect operational constraints.

Over recent decades, forecasting techniques have evolved from classical statistical models such as ARIMA and exponential smoothing to more advanced data-driven approaches rooted in machine learning. These modern methods enable the modeling of *stochastic demand* by capturing complex temporal patterns and incorporating contextual factors, thereby enhancing predictive accuracy and supporting informed planning decisions.

In parallel, optimization techniques ranging from linear and mixed-integer programming to metaheuristics have proven effective in addressing large-scale logistical problems, notably the Vehicle Routing Problem (VRP). These models are crucial for minimizing transportation costs, improving delivery efficiency, and maintaining high service levels, all while considering vehicle capacities, route constraints, and operational feasibility.

Despite their complementary roles, forecasting and optimization are often treated separately in practice. By using demand forecasts to support *strategic planning* such as stock management, fleet sizing or preliminary scheduling, companies can improve responsiveness and resource utilization. The final execution of deliveries, however, rely on *deterministic optimization* based on real, confirmed orders, ensuring that operational decisions remain grounded in actual data.

This thesis adopts an integrated perspective. Developed in collaboration with **Celia**, a key player in Algeria's fresh food distribution sector, this study addresses recurring logistical challenges faced by the company. These include demand variability, limited visibility and coordination, and transport planning difficulties, all of which increase operational costs, reduce overall performance, and impact service quality.

To address these challenges, this research focuses on two complementary components:

- To conduct data analysis on historical records in order to identify customer demand patterns and gain insights into the dataset, followed by the development

of a forecasting model capable of anticipating daily demand. This enables the company to simulate delivery plans, anticipate fleet needs, and improve coordination with clients;

- To implement an operational optimization module that minimizes transportation costs and improves service reliability, using *real customer order data* to assign trucks and design delivery routes in accordance with logistical constraints.

The central research question addressed in this thesis is as follows:

How can demand forecasting, under uncertainty, be combined with deterministic delivery optimization to improve transport planning and operational performance?

This thesis is structured into five chapters:

- **Chapter 1** introduces the company Celia and outlines the logistical challenges it faces;
- **Chapter 2** reviews time series and forecasting methods, including both classical statistical approaches and machine learning-based techniques for demand prediction;
- **Chapter 3** examines the Vehicle Routing Problem (VRP), presenting its mathematical formulation, main variants, and common solution approaches;
- **Chapter 4** presents the case study, including exploratory data analysis and the development of forecasting models based on the company's historical data;
- **Chapter 5** describes the implementation of the optimization model for delivery planning, together with the development of a web application that integrates demand analysis, forecasting, and VRP solutions.

Chapter 1

Presentation of the Company Celia

Introduction

Fresh dairy products such as milk, yogurt, and soft cheeses, occupy a uniquely sensitive position within the agri-food supply chain. Their inherently short shelf life, stringent cold chain requirements, and susceptibility to microbial degradation necessitate highly precise and responsive logistics operations. Any deviation in storage temperature or delivery timing can result in rapid spoilage, with serious implications for consumer safety, regulatory compliance, and brand integrity.

In this context, the efficiency of logistics planning becomes not only a matter of cost optimization but also a critical factor in preserving product quality and sustaining consumer trust. Companies operating in the fresh dairy sector must therefore develop robust, adaptable distribution systems capable of responding to daily demand fluctuations while maintaining strict cold chain integrity across all stages of transport.

This chapter presents the case of **Celia**, a prominent subsidiary of the multinational Lactalis Group and a key player in Algeria's fresh dairy market. It begins by outlining the company's mission, strategic positioning, and portfolio of perishable products. It then provides a detailed overview of Celia's end-to-end logistics operations from receiving client orders to planning delivery routes, emphasizing the operational challenges posed by uncertain demand patterns, product perishability, and the difficulty of consolidating orders across diverse delivery zones.

1.1 Presentation of Lactalis Group

Founded in 1933 in Laval, France, by André Besnier, the Lactalis Group has grown to become a global leader in the dairy industry, operating across all major segments including liquid milk, butter, cream, fresh and aged cheeses, dairy ingredients, and both clinical and infant nutrition. Now managed by the third generation of the Besnier family, the group has continuously pursued product quality and operational excellence.

Today, Lactalis is the world's largest cheese manufacturer and the European market leader in drinking milk, butter, and cream. With an annual turnover of €18.4 billion, 77% of which is generated outside France, the group ranks among the top 15 agri-food companies globally. Each year, it collects approximately 19.2 billion liters of milk. In addition, Lactalis produces 25 cheeses protected by the Appellation d'Origine Protégée (AOP) label and holds leading positions across all major dairy categories.



Figure 1.1: Logo of Group Lactalis [1]

1.2 Presentation of Celia Algérie

Founded in 1980 with the initial importation of “Celia Famille” milk powder, Celia Algérie has progressively reinforced its presence in the national dairy sector. In 2007, as part of a strategic partnership with the Soummam company, the subsidiary acquired the Beni Tamou dairy facility. This collaboration culminated in 2013 with the full acquisition of Soummam’s shares, thereby consolidating Celia’s operations across both “Fresh” and “Dry” dairy segments.

Today, Celia Algérie operates with approximately 465 employees and produces 33,237 tonnes annually from its production site in Beni Tamou, Blida.

This organizational structure enables Celia Algérie to leverage the global expertise of the Lactalis Group while ensuring that its operations are closely tailored to the specific characteristics and consumption patterns of the Algerian market.



Figure 1.2: Logo of Celia Algérie [2]

1.2.1 Product Categories and Classification

Celia’s product offering spans both dry and fresh dairy segments, each with its own characteristics and market dynamics. This section presents an overview of the company’s key product categories and their classification.

Dry Products

The *dry product category* includes long-shelf-life items, primarily milk powder products, which do not require refrigeration.

Celia operates two main product lines within the dry category:

- **Celia Famille:** Locally packaged milk powder for family consumption, adapted to the Algerian market.

- **Celia Infantile:** Imported infant milk formula designed to meet the nutritional needs of newborns and infants.

These products are handled under a model known as “*revente en état*” (resale in the same condition), whereby the goods are imported or packaged locally without undergoing transformation.

Fresh Products

The *fresh product category* includes perishable dairy items that require strict cold chain management. These products are produced locally at the Celia Algeria factory and are distributed on a daily basis due to their short shelf life and high sensitivity to temperature variations.

The fresh product range at Celia includes:

- **Subsidized Milk:** Offered at a controlled price in coordination with national policy.
- **Processed Cheese:** Spreadable and melting cheeses for general consumption.
- **Soft Cheese:** Such as Brie-style or creamy cheeses.
- **Fresh Cheese:** Includes white cheese and other unripened varieties.

The organization of these fresh dairy products into functional categories is summarized in Figure 1.3.



Figure 1.3: Example of Celia’s fresh products. Source: Internal company document

1.3 Organizational Architecture

Celia Algérie operates under a functional organizational structure designed to ensure efficient coordination across specialized departments. The company is led by a General Director who oversees strategic decision-making and operational coordination. The organization is structured around three main pillars: **Executive Leadership**, **Core Operations**, and **Support Functions**, each playing a critical role in the fresh dairy supply chain.

The following table presents the complete organizational architecture, highlighting each department's primary responsibilities:

Table 1.1: Organizational Structure of Celia Algérie

Level	Department/Unit	Key Responsibilities
Executive Leadership	General Director	Strategic oversight and interdepartmental coordination
	Factory Director	Production site management and operational alignment
Core Operations	<i>Production Units</i>	
	Milk Collection	Raw milk sourcing, quality standards, supplier coordination
	Quality Control	Product conformity, safety standards, process audits
	Planning	Production scheduling, demand-supply coordination
	Warehousing	Storage management, cold chain, FIFO(First In, First Out) dispatch
	<i>Distribution Units</i>	
	Supply Chain	End-to-end flow coordination, supplier relations
	Logistics	Route planning, carrier coordination, delivery scheduling
	Transport	Vehicle allocation, fleet management, delivery execution
	<i>Commercial Units</i>	
	Commercial Service	Customer relationships, order collection and validation
	Customer Service	Order tracking, complaint resolution, client communication

Table 1.2: Organizational Structure of Celia Algérie (continued)

Level	Department/Unit	Key Responsibilities
Support Functions	Finance & Accounting	Budget control, payment verification, financial risk management
	Purchasing	Raw materials and services procurement
	Import Service	International procurement, customs, freight logistics
	Human Resources	Staff management, training, performance evaluation

1.4 Logistics Distinctions Between Fresh and Dry Product Lines

Celia Algérie operates two distinct logistics models tailored to the characteristics of its **fresh** and **dry** product lines, reflecting differences in perishability, distribution frequency, volume, and storage requirements.

Fresh products are characterized by high-volume demand and daily distribution cycles, necessitating strict cold chain management and fast, reliable logistics. Daily transfers from the production site in Beni Tamou to the distribution depot are executed via refrigerated trucks operated by *Numilog* company, while last-mile deliveries to customers, mainly wholesalers and regional distribution centers, are handled by subcontracted carriers with refrigerated vehicles.

Dry products, in contrast, follow a more static distribution model with lower volumes and reduced frequency (typically monthly or semi-monthly deliveries). These items are stored under ambient conditions without refrigeration requirements. Both depot transfers and the transport of imported goods from maritime terminals are handled by *Mars Logistique* company, with final deliveries to clients executed by *La Flèche Bleue Algérienne* company.

1.5 Fresh Products Distribution System

Celia's fresh dairy products are managed through a dual-site logistics setup that separates manufacturing from distribution activities. While the Beni Tamou site focuses on production, storage and distribution operations are handled at the Boufarik depot through a partnership with Mars Logistique, which oversees both warehousing and daily order preparation.

1.5.1 Depot Operations – Mars Logistique

Mars Logistique is a third-party logistics provider offering comprehensive supply chain services since 2008. With over 300 employees, the company manages goods

reception, storage, order consolidation, and dispatching for Celia. This strategic partnership enables Celia to outsource its entire downstream logistics operations. [19]

1.5.2 Client Network and Fleet Configuration

Celia's fresh products are distributed to a wide network of wholesalers and regional distribution centers across Algeria. To accommodate varying regional accessibility and order volumes, the company utilizes a heterogeneous fleet of subcontracted refrigerated trucks with five distinct vehicle types, each with specific pallet capacities:

Table 1.3: Fleet Configuration by Vehicle Type

Truck Type	Pallet Capacity
2.5 Tons	4 pallets
4 Tons	7 pallets
7 Tons	11 pallets
10 Tons	14 pallets
20 Tons	33 pallets

1.5.3 Daily Transfers and Cold Chain Integrity

Finished products are transported daily from the factory to the depot, maintaining uninterrupted cold chain integrity. Upon arrival, goods are promptly stored in temperature-controlled zones until order processing. Products are then dispatched from the depot to distribution centers, with continuous temperature monitoring throughout the entire journey to ensure product quality and safety.

1.5.4 Order Consolidation and Palletization

Customer orders, initially submitted in crate units, are consolidated into standardized pallets during warehouse preparation. Pallet loading follows weight-based optimization to maximize truck space utilization while maintaining load stability. Each pallet can accommodate up to approximately 310 kilograms of product, enabling efficient vehicle capacity utilization across the heterogeneous fleet.

1.5.5 Order Validation and Distribution Scheduling

Upon receipt, customer orders undergo financial validation through the Finance & Accounting department. Orders exceeding a customer's authorized credit limit are automatically blocked from processing. Validated orders must be processed before the daily cut-off time of 4:00 PM to ensure next-day delivery scheduling.

After validation, orders are forwarded to the transport planning coordinator, who manually contacts third-party logistics partners to confirm vehicle availability and arrange next-day (D+1) deliveries.

1.5.6 Delivery Waves and Execution

Deliveries are organized into two distinct dispatch waves to optimize delivery timing across different geographic zones:

- **First Wave (11:00 PM, D-Day):** Trucks assigned to long-distance deliveries (southern, eastern, and western regions) are loaded during the night shift to ensure timely arrival at distant destinations.
- **Second Wave (6:00 AM, D+1):** Trucks serving central and nearby zones depart in the early morning, enabling same-day delivery within acceptable time windows for local customers.

1.5.7 Monitoring and Compliance

Throughout the distribution process, the logistics manager ensures operational compliance by issuing delivery notes, monitoring refrigerated transport conditions through digital temperature logs, and verifying adherence to food safety standards. This monitoring system maintains product quality assurance from depot departure to final delivery.

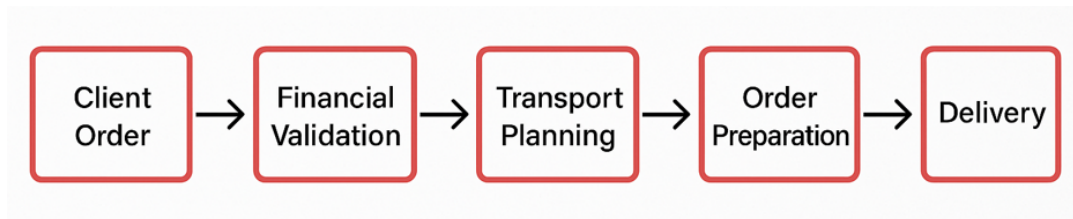


Figure 1.4: Fresh product distribution workflow from order intake to final delivery

This tightly coordinated yet manually intensive process is increasingly challenged by demand variability, rising costs, and the growing need for timely information to support decision-making. The next sections will explore how these challenges form the foundation of the approach proposed in this thesis.

1.6 The Company's Logistics Challenges

Despite a structured distribution framework, **Celia** faces a number of critical logistical challenges that compromise operational efficiency and service quality:

- **Demand Variability:** The absence of fixed ordering schedules for each customer and highly variable daily order volumes hinder accurate demand forecasting, making effective resource allocation difficult.

- **Lack of Real-Time Transport Visibility:** Without real-time truck tracking systems, the company is unable to monitor vehicle locations, arrival times or unloading durations, generating uncertainty in the delivery process and affecting coordination with distribution centers.
- **Cold Chain Vulnerability:** Fresh dairy products require strict temperature control. Excessive handling (more than three successive unloading operations) or prolonged unloading times can compromise product quality, leading to spoilage and increased waste.
- **Operational Time Constraints:** The compressed timeframe between order confirmation and dispatch creates pressure on warehouse operations, particularly when vehicles are unavailable or delayed.
- **Manual Route Planning:** Delivery routes and vehicle assignments are managed manually, without algorithmic optimization, resulting in inefficient capacity utilization and suboptimal route planning.

1.7 Problem Statement

Faced with growing logistical complexity, highly variable customer demand, and the absence of automated planning tools, Celia Algérie confronts two interconnected operational challenges that form the foundation of this research:

- (1) **How can daily customer demand be accurately forecasted to enable proactive resource planning and prevent operational bottlenecks?**
- (2) **Simultaneously, how can delivery routes be optimally designed to maximize fleet utilization while minimizing transportation costs and maintaining service quality?**

These questions address the core need for integrated demand forecasting and route optimization capabilities that can transform Celia's current manual, reactive approach into a data-driven, proactive logistics management system.

Conclusion

The logistics management of fresh products at Celia Algérie operates within a well-structured framework, yet faces several critical operational challenges. These include significant demand fluctuations, limited real-time visibility, product perishability constraints, and the absence of advanced optimization tools. Together, these limitations hinder effective planning and negatively impact both service quality and operational efficiency.

To address these constraints, this thesis will explore integrated solutions combining demand forecasting with delivery optimization. The following chapters will present time series forecasting methods, vehicle routing optimization techniques, and their practical implementation through a comprehensive case study at Celia, ultimately demonstrating how data-driven approaches can enhance planning accuracy, optimize distribution operations, and improve customer satisfaction.

Chapter 2

Overview of Time Series and Forecasting Methods

Introduction

Accurate forecasting from temporal data requires understanding both the underlying patterns in historical observations and the theoretical foundations of predictive methodologies. The challenge lies not only in selecting appropriate forecasting techniques but also in recognizing their inherent limitations and optimal application contexts.

This chapter provides the theoretical framework for temporal data analysis and forecasting in two complementary sections. The first establishes fundamental concepts in time series analysis, including pattern identification, statistical properties, and diagnostic procedures essential for characterizing temporal behavior. The second examines forecasting methodologies spanning classical statistical approaches and machine learning techniques, with particular emphasis on their theoretical assumptions, algorithmic foundations, and practical limitations.

2.1 Generalities About Forecasting

Forecasting involves predicting future events based on available data and is a fundamental tool used across a wide range of fields, including business, industry, government, economics, environmental sciences, medicine, social sciences, politics, and finance.

Forecasting problems are generally categorized according to their time horizon:

- **Short-term forecasts:** covering a few days, weeks, or months;
- **Medium-term forecasts:** spanning up to one or two years;
- **Long-term forecasts:** extending beyond several years. [20]

Each forecasting horizon serves different strategic and operational purposes. Short- and medium-term forecasts are particularly important for operational planning, budgeting, inventory control, supply chain coordination, and resource allocation. In contrast, long-term forecasts support broader decisions, such as infrastructure investments, strategic initiatives, and long-term policy development.

2.1.1 Methodological Classification of Forecasting Approaches

Forecasting techniques are generally divided into two broad categories:

- **Qualitative methods**, which rely on expert judgment, intuition, and subjective evaluations. These approaches are particularly useful when historical data is limited or unavailable. Common examples include the Delphi method, market research, and scenario planning.

- **Quantitative methods**, which utilize historical data and rely on mathematical or statistical models to predict future values. These include *causal models*, which assume a cause-effect relationship between variables, and *non-causal models*, which focus on identifying patterns and dependencies within the data itself [20].

2.2 Time Series Theory

Among the various quantitative forecasting approaches, time series analysis plays a central role because it captures temporal dependencies in data. Time series forecasting is especially effective when the order and timing of past observations carry critical information for future projections. A time series is defined as a sequence of observations collected at regular intervals (e.g., daily, weekly, monthly, or annually), where the chronological structure is essential for analysis [21].

2.2.1 Components of a Time Series

A typical time series is influenced by three primary components, which can be decomposed from the observed data:

- **Trend (T)**: The long-term progression of the series either upward, downward, or stable over an extended period. It reflects underlying growth or decline patterns.
- **Seasonality (S)**: Recurring patterns at fixed periods within a year (e.g., months, quarters), often influenced by climate, holidays, or calendar events.
- **Irregular (I)**: Unpredictable or residual fluctuations that cannot be attributed to the other two components. These are caused by unexpected events.

Understanding these components enables selecting appropriate modeling approaches, as discussed next. [21].

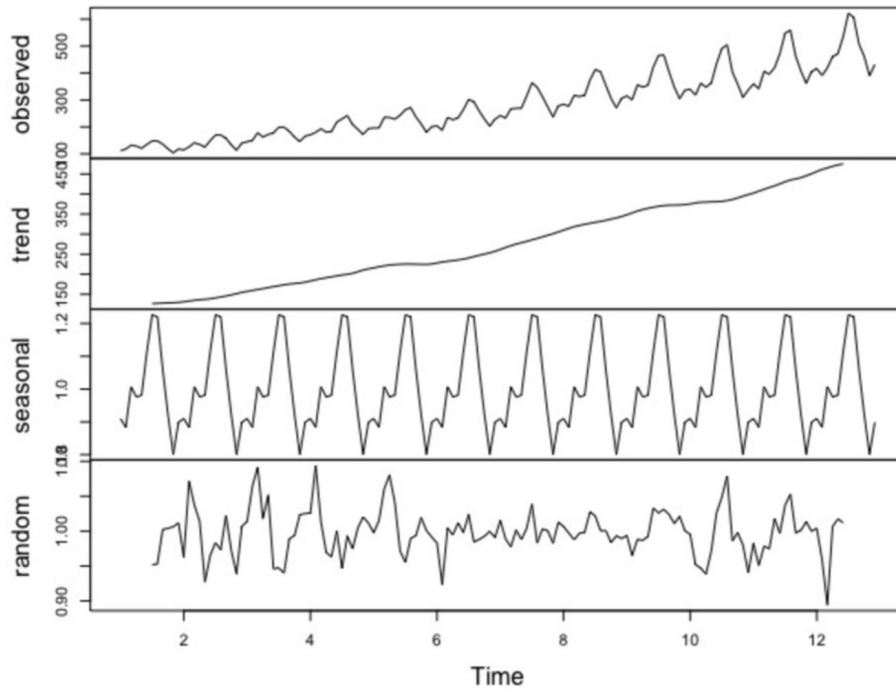


Figure 2.5: Time series decomposition showing trend, seasonal, and random components in temporal data analysis [3]

2.2.2 Time Series Models: Additive vs. Multiplicative

Two fundamental models are commonly used to represent the structure of time series data:

- **Additive model:**

$$Y(t) = T(t) + S(t) + I(t)$$

This model assumes that all components act independently and contribute additively to the observed value. It is typically used when seasonal variations remain constant over time.

- **Multiplicative model:**

$$Y(t) = T(t) \times S(t) \times I(t)$$

This model assumes that the components interact in a proportional manner. It is more appropriate when the magnitude of seasonal fluctuations increases with the trend.

The choice between models depends on whether seasonal variation remains constant in absolute terms (additive) or proportional to the trend level (multiplicative) [21].

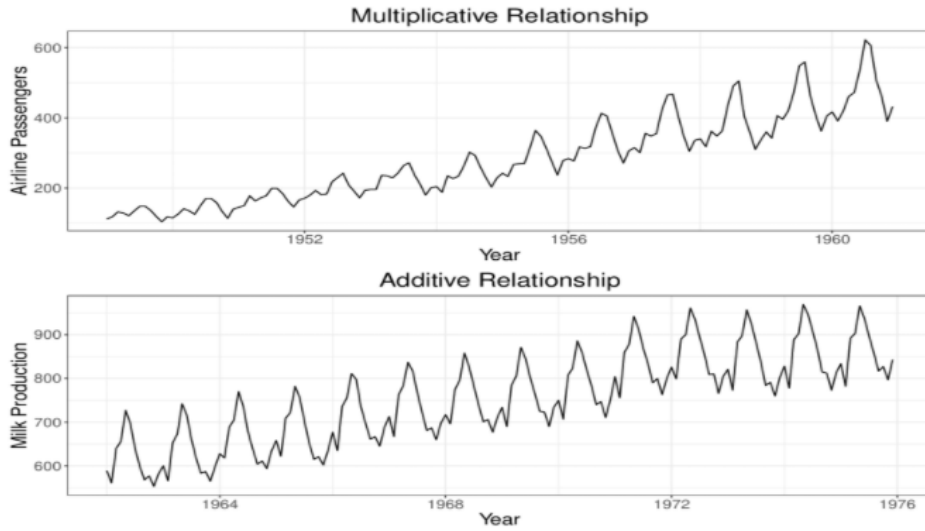


Figure 2.6: Comparison of additive versus multiplicative time series models [4]

2.2.3 Deterministic vs Stochastic Time Series

Building on the foundational components of time series, this subsection distinguishes between deterministic and stochastic series, two core classifications that influence model choice and forecasting accuracy.

Deterministic time series

A deterministic time series is a sequence where each value is completely determined by a mathematical function of time, without random variation:

$$x_t = f(t)$$

Examples include linear trends and periodic functions. These series are fully predictable once the generating function is known [22].

Stochastic time series

A stochastic time series contains random components that make future values unpredictable with certainty. Each observation is considered a realization of an underlying random process governed by specific **statistical properties**, termed a stochastic process [23, 24].

Fresh dairy demand exemplifies stochastic time series behavior, where despite identifiable patterns (lower demand on weekends, seasonal preferences), the exact order quantity from any specific customer on a given day remains unpredictable due to various random factors such as weather conditions, promotional activities, or competitor actions.

2.2.4 Stationarity in Time Series

Stationarity implies that the statistical properties of a series: mean, variance, and autocovariance, remain constant over time.

- **Strict Stationarity:** The joint distribution of any set of observations remains invariant under time shifts a strong condition rarely satisfied in practice.
- **Weak Stationarity (Covariance Stationarity):** Requires that:
 - $E[X_t] = \mu$ (constant mean)
 - $\text{Var}[X_t] = \sigma^2$ (constant variance)
 - $\text{Cov}(X_t, X_{t+h}) = \gamma(h)$ (covariance depends only on lag)

This more practical condition is widely used in time series analysis

Non-stationary series can often be transformed to achieve stationarity through differencing or detrending procedures [25].

- **Stationarity testing:** The Augmented Dickey–Fuller (ADF) test and the KPSS (Kwiatkowski–Phillips–Schmidt–Shin) test are commonly used to assess whether a series exhibits stationary behavior. These tests should be applied before model estimation. [26]

The theoretical foundations established above become particularly relevant when analyzing real-world demand data, which often exhibits complex patterns that challenge traditional forecasting assumptions.

2.2.5 Classification of Demand Patterns

In the context of demand forecasting, recognizing and understanding these patterns are essential for selecting models that align with the underlying data behavior.

The academic literature has classified time series patterns, particularly relevant for scenarios involving irregular observations. This classification system, established by Syntetos and Boylan [27], provides a quantitative framework for distinguishing between different types of temporal patterns based on two key statistical measures.

Classification Parameters

The classification employs two fundamental parameters to characterize time series behavior:

- **Average Demand Interval (ADI):** Measures the average time between successive non-zero observations:

$$ADI = \frac{\text{Total observation period}}{\text{Number of non-zero observations}}$$

- **Squared Coefficient of Variation (CV^2):** Quantifies the variability of non-zero values:

$$CV^2 = \left(\frac{\sigma}{\mu} \right)^2$$

where σ is the standard deviation and μ is the mean of non-zero observations.

Time Series Categories

Based on threshold values for ADI and CV^2 , time series are classified into four distinct categories [27]:

- **Smooth patterns:** $ADI \leq 1.32$ and $CV^2 \leq 0.49$
Characterized by regular occurrences with consistent magnitudes. These series exhibit predictable patterns suitable for traditional forecasting methods.
- **Intermittent patterns:** $ADI > 1.32$ and $CV^2 \leq 0.49$
Feature irregular timing of occurrences but relatively consistent magnitudes when events occur. The sporadic nature requires specialized forecasting approaches.
- **Erratic patterns:** $ADI \leq 1.32$ and $CV^2 > 0.49$
Display regular timing but highly variable magnitudes. The predictable occurrence combined with unpredictable sizes presents unique modeling challenges.
- **Lumpy patterns:** $ADI > 1.32$ and $CV^2 > 0.49$
Exhibit both irregular timing and highly variable magnitudes, representing the most challenging forecasting scenario. These patterns require sophisticated modeling approaches that can handle dual uncertainty.

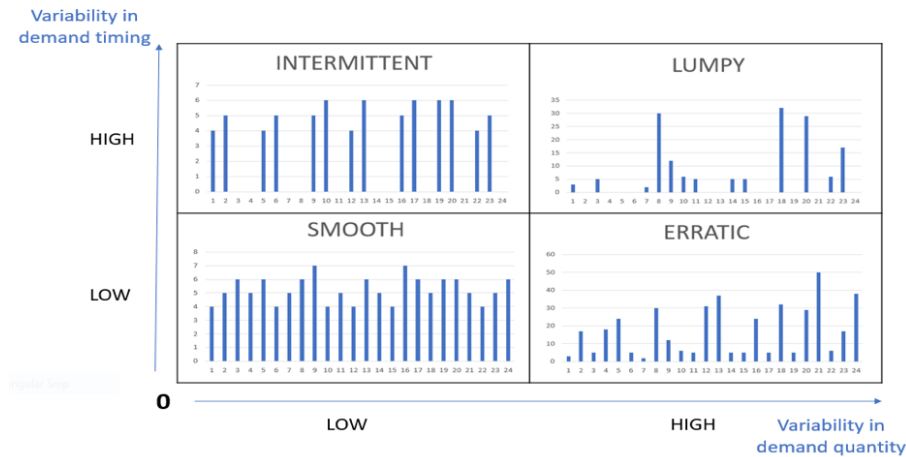


Figure 2.7: Classification of demand patterns based on variability in timing and quantity [5].

This framework provides essential guidance for forecasting method selection, as each category demands different modeling strategies and performance evaluation criteria. In the rest of the sections, we will focus on the intermittent time series.

2.3 Forecasting Approaches

Building on the foundational understanding of time series, this section delves into forecasting methods. In the case of intermittent demand, which is characterized by sporadic and irregular occurrences [28], classical models like Croston [29] and TSB [30] are particularly well-suited. These models specifically address the challenges posed by zero-inflated and irregular demand. For more advanced forecasting, machine learning models particularly gradient boosting methods [31] have also shown promising results for handling such demand patterns, as they can capture complex, non-linear relationships.

2.3.1 Classical Forecasting Models

Croston's Method: Definition and Foundation

Croston's method, introduced by J.D. Croston in 1972, is a statistical forecasting technique specifically developed for time series characterized by a large proportion of zero-valued observations interspersed with non-zero values. This class of time series, often referred to as intermittent series, poses challenges for conventional forecasting models due to the irregularity and sparsity of meaningful events.

The core idea of Croston's method is to decouple the forecasting process into two distinct components:

- The magnitude component: representing the value of non-zero observations.
- The interval component: representing the time elapsed between successive non-zero observations.

Each component is independently updated using simple exponential smoothing, with a smoothing parameter $\alpha \in (0, 1)$. The method assumes that the magnitude and interval components are independent and can be modeled separately without loss of information. The forecast is computed as the ratio between the smoothed magnitude and the smoothed interval, resulting in a constant estimate that reflects the average behavior of the non-zero occurrences over time.

Formally, if z_i denotes the i -th non-zero value and p_i the time since the previous one, the updates are:

$$\hat{z}_i = \alpha z_i + (1 - \alpha) \hat{z}_{i-1} \quad (2.1)$$

$$\hat{p}_i = \alpha p_i + (1 - \alpha) \hat{p}_{i-1} \quad (2.2)$$

and the forecast is given by:

$$\hat{y}_{t+h} = \frac{\hat{z}_i}{\hat{p}_i}, \quad \forall h > 0 \quad (2.3)$$

The method proceeds through the following key steps[32, 33]:

1. Initialization: At the first non-zero demand observed at time t_0 , the following initial estimates are defined:

- $\hat{z}_{t_0} = y_{t_0}$: the initial smoothed demand,
- $\hat{p}_{t_0} = 1$: assuming a one-period interval,
- $\hat{y}_{t_0} = \hat{z}_{t_0}/\hat{p}_{t_0}$: the initial forecast.

Additionally, a tracking variable t_{last} is set to t_0 to record the last time a non-zero demand occurred.

2. Iterative Updating: For each time step $t > t_0$, the update mechanism depends on whether the demand y_t is zero or non-zero:

- **If $y_t > 0$:**

- Update demand estimate:

$$\hat{z}_t = \alpha y_t + (1 - \alpha)\hat{z}_{t-1} \quad (2.4)$$

- Update interval estimate:

$$\hat{p}_t = \alpha(t - t_{\text{last}}) + (1 - \alpha)\hat{p}_{t-1} \quad (2.5)$$

- Set $t_{\text{last}} = t$.

- **If $y_t = 0$:**

- Retain previous estimates: $\hat{z}_t = \hat{z}_{t-1}$ and $\hat{p}_t = \hat{p}_{t-1}$.

In all cases, the forecast is recalculated as:

$$\hat{y}_t = \frac{\hat{z}_t}{\hat{p}_t} \quad (2.6)$$

3. Forecasting Future Demand: For any forecast horizon $h \geq 1$, Croston's method produces a constant forecast:

$$\hat{y}_{T+h} = \frac{\hat{z}_T}{\hat{p}_T} \quad (2.7)$$

Algorithm 1: Croston's Method for Intermittent Demand Forecasting

Input: Demand series y_t for $t = 1, \dots, T$
Input: Smoothing parameter $\alpha \in (0, 1)$
Output: Forecast $\hat{y}_{T+1}, \dots, \hat{y}_{T+h}$

- 1 **Initialization:**
- 2 Find first non-zero demand at time t_0
- 3 Set $\hat{z}_{t_0} = y_{t_0}$, $\hat{p}_{t_0} = 1$, $\hat{y}_{t_0} = \hat{z}_{t_0}/\hat{p}_{t_0}$, $t_{\text{last}} = t_0$
- 4 **for** $t = t_0 + 1$ **to** T **do**
- 5 **if** $y_t > 0$ **then**
- 6 $\hat{z}_t = \alpha y_t + (1 - \alpha)\hat{z}_{t-1}$
- 7 $\hat{p}_t = \alpha(t - t_{\text{last}}) + (1 - \alpha)\hat{p}_{t-1}$
- 8 $t_{\text{last}} = t$
- 9 **else**
- 10 $\hat{z}_t = \hat{z}_{t-1}$
- 11 $\hat{p}_t = \hat{p}_{t-1}$
- 12 **end**
- 13 $\hat{y}_t = \hat{z}_t/\hat{p}_t$
- 14 **end**
- 15 **for** $h = 1$ **to** *forecast horizon* **do**
- 16 $\hat{y}_{T+h} = \hat{z}_T/\hat{p}_T$
- 17 **end**

Teunter–Syntetos–Babai (TSB) Method

The Teunter–Syntetos–Babai (TSB) method is a forecasting technique developed to address limitations in earlier approaches. While Croston's method assumes a constant probability of occurrence over time, the TSB method models this probability explicitly and updates it dynamically[32].

The TSB method separates the forecasting problem into two components:

- **Occurrence component:** estimates the probability that a non-zero observation occurs at a given time
- **Magnitude component:** estimates the average size when a non-zero observation occurs

Each component is updated using simple exponential smoothing.

Step-by-Step Procedure Let:

- z_t be the observed value at time t
- $p_t = \mathbb{I}(z_t > 0)$ be an indicator function that is 1 if $z_t > 0$ and 0 otherwise
- p'_t be the estimated probability of occurrence at time t
- z'_t be the estimated magnitude of a non-zero value

- Y'_t be the forecast at time t

Then the method proceeds as follows:

1. Initialization:

- Set initial values p'_0 and z'_0 (e.g., based on historical averages).

2. Smoothing Updates: At each time step t , the estimates are updated based on whether demand occurs:

If $p_t = 0$ (no demand occurs):

$$p'_t = p'_{t-1} + \beta(0 - p'_{t-1}) \quad (2.8)$$

$$z'_t = z'_{t-1} \quad (2.9)$$

$$Y'_t = p'_t \cdot z'_t \quad (2.10)$$

If $p_t = 1$ (demand occurs):

$$p'_t = p'_{t-1} + \beta(1 - p'_{t-1}) \quad (2.11)$$

$$z'_t = z'_{t-1} + \alpha(z_t - z'_{t-1}) \quad (2.12)$$

$$Y'_t = p'_t \cdot z'_t \quad (2.13)$$

where α and β are smoothing parameters ($0 < \alpha, \beta < 1$).

3. Forecast Computation: The forecast at time t is always given by:

$$Y'_t = p'_t \cdot z'_t \quad (2.14)$$

Key Characteristics

- The method accounts for changing probabilities over time, so it can respond to demand drops or sudden spikes.
- Unlike Croston's method, the TSB method updates estimates at every time step, regardless of whether the value is zero or not.
- The forecast remains valid and dynamically updated, even during extended sequences of zero observations. [30]

Algorithm 2: TSB (Teunter–Syntetos–Babai) Method

Input: Time series $\{z_t\}_{t=1}^T$, smoothing parameters α and β
Output: Forecast Y'_{T+1} for next period

```
1 Initialization:
2 Estimate initial probability  $p'_0$  and initial demand size  $z'_0$  from non-zero
  observations
3 for  $t = 1$  to  $T$  do
4   if  $z_t > 0$  then
5      $p_t \leftarrow 1$  ;                                // Demand occurs
6      $p'_t \leftarrow p'_{t-1} + \beta(1 - p'_{t-1})$  ;      // Update probability
7      $z'_t \leftarrow z'_{t-1} + \alpha(z_t - z'_{t-1})$  ;    // Update magnitude
8   else
9      $p_t \leftarrow 0$  ;                                // No demand
10     $p'_t \leftarrow p'_{t-1} + \beta(0 - p'_{t-1})$  ;      // Update probability
11     $z'_t \leftarrow z'_{t-1}$  ;                          // Keep previous magnitude
12  end
13   $Y'_t \leftarrow p'_t \cdot z'_t$  ;                      // Forecast for period  $t$ 
14 end
15 return  $Y'_{T+1} = p'_T \cdot z'_T$ 
```

2.3.2 Limitations of Classical Methods

Classical intermittent demand forecasting methods face several fundamental challenges:

- **Bias issues:** Croston’s method exhibits positive bias, while variants like TSB reduce but don’t eliminate this bias [34].
- **Parameter optimization:** Lacks consistent methodology for parameter selection, with conventional accuracy metrics being inappropriate for intermittent contexts [35].
- **Statistical foundation:** Methods don’t correspond to underlying stochastic models, preventing computation of prediction intervals [36].
- **TSB limitations:** TSB’s frequent probability updates can negatively affect MSE performance, particularly for non-stationary demands [37].

These limitations have motivated research into machine learning approaches that can capture more complex relationships and incorporate diverse information sources.

2.3.3 Artificial Intelligence-Based Approaches

Artificial Intelligence (AI): is commonly defined as the science and engineering of creating intelligent machines that exhibit behaviors typically associated with human cognition, such as reasoning, learning, and problem solving [38, 39].

Machine Learning (ML): a subfield of Artificial Intelligence (AI), enables computers to improve their performance on a task through experience derived from data. As noted by Jeong (2020), ML is defined as the ability of a computer to “learn without being explicitly programmed” [40]. In practice, Machine Learning (ML) systems ingest data and automatically adjust internal models or parameters to make predictions or decisions. Depending on the nature of the available data and the research objectives, a suitable learning paradigm is selected. According to Sarker (2021), ML approaches are generally classified into four categories: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

- **Supervised learning:** trained on labeled data, enabling prediction by comparing to known outputs.
- **Unsupervised learning:** works with unlabeled data to identify hidden structures.
- **Semi-supervised learning:** uses a combination of a small labeled dataset and a large unlabeled one.
- **Reinforcement learning:** learns by interacting with the environment and receiving feedback in the form of rewards or penalties. Popular reinforcement learning algorithms include Q-learning and Deep Q-Networks (DQN), often applied in robotics and games [41].

The following subsections examine how these machine learning paradigms, particularly supervised learning approaches, can be applied to time series forecasting challenges.

Supervised Learning

Supervised learning consists of input features and their corresponding output labels. The learning process includes:

- **Training data:** A dataset of input-output pairs is fed into the model.
- **Learning process:** The algorithm adjusts internal parameters to minimize prediction error.
- **Evaluation and optimization:** A test dataset is used to validate and fine-tune the model using several techniques like cross-validation and regularization to prevent overfitting.

Supervised learning is generally divided into two algorithm categories:

- **Regression:** Predicts continuous values such as weight or sales.
- **Classification:** Predicts categorical outcomes like yes/no decisions [42].

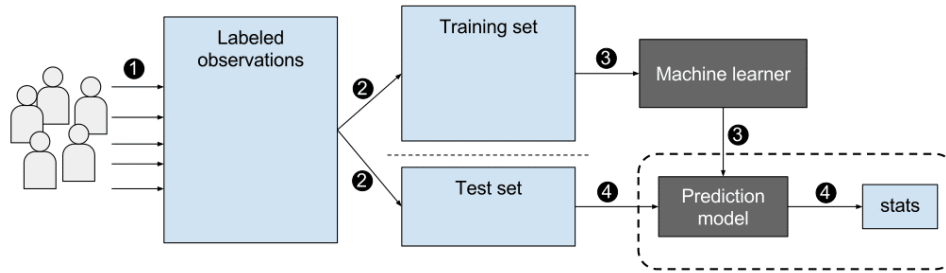


Figure 2.8: Schema of supervised learning [6]

Decision Trees

Decision trees are supervised learning models used for classification and regression. They work by splitting the data into smaller parts based on feature values. The tree starts at a root node and branches out through decision nodes, ending at leaf nodes that give the final output. Each decision checks a feature, and for numerical values, it compares them to a threshold. The model makes predictions by following a path from the root to a leaf based on the input features. [43]

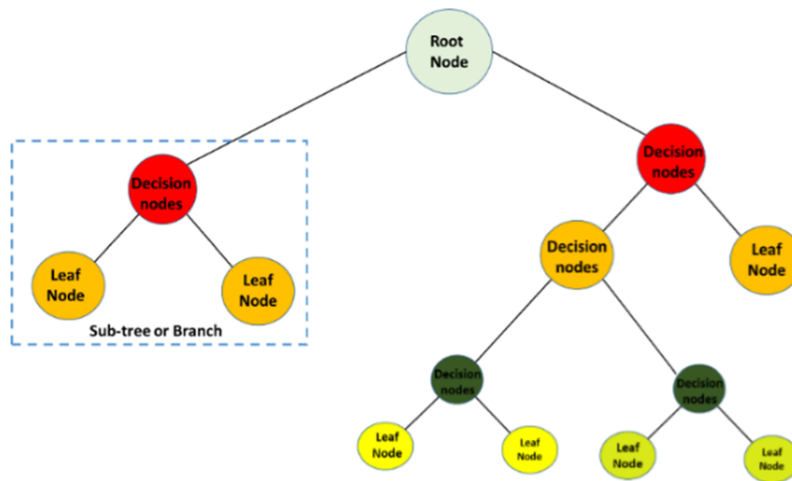


Figure 2.9: Illustration of a decision tree for Regression [7]

While decision trees are valued for their interpretability, non-linear modeling capability, and ability to handle both categorical and numerical data, they are also prone to overfitting, instability, and bias in the presence of imbalanced datasets. To address these limitations, ensemble methods such as Random Forests and XG-Boost combine multiple decision trees to improve performance, generalization, and robustness.

Ensemble Learning

Ensemble learning is a powerful approach in machine learning that combines the predictions of multiple base models, often referred to as weak learners, into a single,

more accurate and robust model. This technique is based on the idea that a group of diverse models can outperform any individual model, much like a panel of experts provides better decisions than a single voice [44].

Two of the most widely used ensemble strategies are bagging and boosting:

- **Bagging (Bootstrap Aggregating):** This method builds multiple models in parallel by training them on different random subsets of the training data, sampled with replacement. The final prediction is obtained by aggregating the outputs via majority vote for classification or averaging for regression. Bagging helps reduce variance and is particularly effective in stabilizing high-variance models like decision trees. A common example of bagging is Random Forest, which combines multiple decision trees trained on different bootstrap samples of the data [45, 46].
- **Boosting:** Unlike bagging, boosting trains models sequentially. Each new model attempts to correct the errors made by the previous one, assigning greater weight to misclassified instances. This process reduces bias and enhances accuracy by focusing learning on the most difficult cases [47].

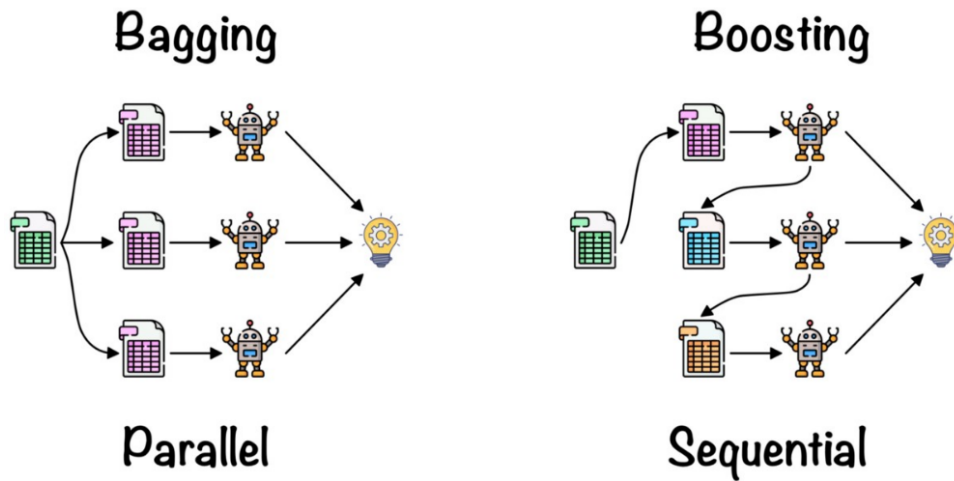


Figure 2.10: Comparison of Bagging and Boosting methodologies [8]

Among boosting algorithms, gradient boosting methods have proven particularly effective for time series forecasting tasks, as demonstrated by their strong performance in recent forecasting competitions [48, 49]. These methods, including XGBoost [50], can capture complex patterns in data through gradient optimization.

Gradient Boosting

Gradient Boosting is a sequential ensemble method that builds models incrementally to optimize a specified loss function. At each stage, a new decision tree is trained to fit the residuals (errors) of the current model. By iteratively minimizing the loss

function through gradient descent, the ensemble gradually improves its predictions. Gradient Boosting is widely used in both regression and classification tasks [31].

XGBoost

XGBoost (eXtreme Gradient Boosting) is an optimized implementation of gradient boosting that incorporates regularization, parallel computation, and pruning strategies to improve both speed and performance. It builds trees sequentially, with each tree correcting the errors of the previous ones based on gradient information. XGBoost is particularly known for its scalability and high predictive power [50].

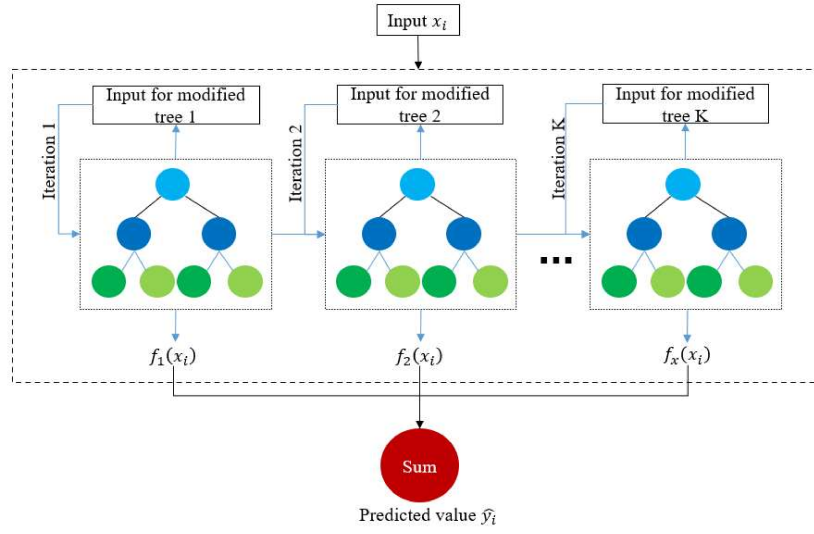


Figure 2.11: Structure of the XGBoost model [9]

XGBoost incorporates several key innovations that enhance its performance and flexibility. It introduces a regularization term in its objective function to penalize overly complex trees, which helps prevent overfitting and encourages better generalization to new data. Moreover, XGBoost is optimized for speed and scalability through features like parallel processing, efficient use of hardware cache, and support for out-of-core and GPU (Graphics Processing Unit), accelerated training on large datasets [50, 51].

Mathematically, the prediction for each data point is represented as the sum of the outputs from all individual trees:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i) \quad (2.15)$$

where:

- $\hat{y}_i$ is the final prediction for instance i ,
- K is the total number of trees,
- f_k represents the k^{th} decision tree.

The algorithm optimizes an *objective function* at each step, which is composed of two parts:

- A **loss function** $l(\hat{y}_i, y_i)$ that measures how well the model is performing (e.g., mean squared error for regression, or log-loss for classification),
- A **regularization term** $\Omega(f_k)$ that penalizes the complexity of the model.

The overall objective function is given by:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (2.16)$$

where the regularization term is defined as:

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (2.17)$$

Here, T is the number of leaves in the tree, w is the vector of leaf weights, γ is the complexity cost per leaf, and λ is the L2 regularization term on weights.

This combination enables XGBoost to maintain a balance between predictive accuracy and model simplicity.

Another critical aspect of XGBoost is its ability to evaluate all potential tree splits using both the first and second derivatives of the loss function, referred to as *gradients* and *Hessians*. This allows the algorithm to make more informed decisions when growing trees, selecting splits that improve performance while controlling complexity. Formally, for each data point i , the gradient g_i and Hessian h_i are computed as:

$$g_i = \frac{\partial l(\hat{y}_i, y_i)}{\partial \hat{y}_i}, \quad h_i = \frac{\partial^2 l(\hat{y}_i, y_i)}{\partial \hat{y}_i^2} \quad (2.18)$$

These values are then used in the approximate split finding process to optimize the gain at each node. This second-order approximation enables efficient and accurate optimization of the objective function. [50].

Finally, XGBoost incorporates several technical enhancements that make it “eXtreme”:

- **Level-wise tree growth:** Trees are grown depth-by-depth rather than leaf-wise.
- **Built-in handling of missing data:** The model can learn the optimal direction to follow when a feature is missing.
- **Shrinkage (learning rate):** Controls the impact of each new tree, helping improve robustness.
- **Scalability and speed:** Through parallelization, distributed computing, and cache optimization [50].

Algorithm 3: General XGBoost Algorithm

Data: A training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ **Result:** A predictive model $\hat{y}(x)$ as an ensemble of decision trees**1 Begin****2** Initialize predictions: $\hat{y}_i^{(0)} = 0$ for all $i \in \{1, \dots, n\}$;**3 for** $t = 1$ **to** T (*number of boosting rounds*) **do****4** **Step 1: Compute gradients and Hessians ;****5** **foreach** $i \in \{1, \dots, n\}$ **do****6** $g_i \leftarrow \frac{\partial \ell(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} ;$ **7** $h_i \leftarrow \frac{\partial^2 \ell(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2}$ **8** **Step 2: Learn a regression tree** f_t ;**9** Fit $f_t(x)$ by optimizing the regularized objective:

$$\mathcal{L}^{(t)} = \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t)$$

where $\Omega(f_t)$ is the regularization term controlling model complexity ;**10** **Step 3: Update predictions ;****11**

$$\hat{y}_i^{(t)} \leftarrow \hat{y}_i^{(t-1)} + \eta f_t(x_i)$$

where η is the learning rate**12 Return the final model:**

$$\hat{y}(x) = \sum_{t=1}^T \eta f_t(x)$$

13 End

2.3.4 Advantages of Gradient Boosting in Time Series Forecasting

Gradient boosting algorithms provide several distinct advantages over traditional forecasting methods when applied to demand forecasting task. The ensemble nature of gradient boosting excels at handling the inherent noise and volatility present in demand data by combining multiple weak learners to create robust predictions that are less susceptible to outliers and irregular demand patterns [50]. The algorithm's ability to automatically handle mixed data types allows for seamless integration of diverse forecasting features, including historical demand patterns, promotional calendars, pricing information, economic indicators, and external factors such as weather data, without requiring extensive preprocessing or transformation [52].

The sequential learning process of gradient boosting, where each successive

model corrects the errors of its predecessors, makes it particularly well-suited for the dynamic and evolving nature of consumer demand, allowing the model to adapt to changing market conditions, emerging trends, and seasonal shifts over time [53].

Despite their advantages in handling complex patterns, machine learning approaches introduce their own limitations that must be carefully considered in forecasting applications: Gradient boosted trees (GBT) have poor extrapolation ability and rely on piecewise constant approximations, which may cause them to struggle with capturing linear or complex temporal relationships without significant feature engineering. They also lack automatic feature engineering and often require domain knowledge for time series tasks [54].

2.3.5 Overfitting Prevention and Hyperparameter Optimization

Overfitting Prevention

While evaluating forecasting models, it is important to consider the risk of *overfitting*, where a model performs well on training data but poorly on unseen data. This issue is particularly common in complex models with many parameters, such as ensemble methods and deep learning approaches.

To reduce this risk and ensure robust model performance, several techniques can be applied:

Early Stopping Strategy Early stopping is a strategy that determines the optimal point at which training should be stopped. Stopping too early may lead to underfitting, while stopping too late can result in overfitting. The objective is to identify the moment when the model no longer improves. To achieve this, performance on the validation set is monitored at the end of each epoch, and training is stopped once no further improvement is observed [55]

Regularization Strategy Generally, model output can be affected by multiple features, and as the number of features increases, the model becomes more complicated. An overfitting model tends to consider all features, even those with very limited effect on the final output, or worse, those that represent meaningless noise. [55]

Regularization addresses this through two approaches:

1. Selecting only useful features and removing useless features from the model
2. Minimizing the weights of features that have little influence on the final classification

Since it is not always clear which features are useless, regularization attempts to limit the effect of all potentially problematic features by minimizing the cost function [55].

The general regularization methods are: (1) L1 regularization and (2) L2 regularization.

L1 Regularization (Lasso): This method minimizes the cost function by forcing some feature weights to zero, effectively removing less important variables. As a result, the model becomes simpler and easier to interpret, though it may also discard potentially useful features with lower influence.

L2 Regularization (Ridge): This method adds a penalty proportional to the squared magnitude of the weights (Euclidean norm). Instead of eliminating features, it reduces their influence by assigning smaller weights. Thus, the model retains more information, but only features with strong predictive power receive large weights.

Hyperparameter Optimization

Random Search: Following Bergstra and Bengio [56], it's a hyperparameter optimization strategy that samples parameter combinations randomly from specified distributions, rather than exhaustively testing all combinations like grid search. Random Search is more efficient because hyperparameter response functions typically have low effective dimensionality only a few parameters significantly impact performance.

2.3.6 Evaluation Metrics

Forecasting accuracy assessment requires multiple complementary metrics, each capturing different aspects of prediction quality:

- **Mean Absolute Error (MAE):** Measures average absolute deviations, treating all errors equally regardless of direction.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.19)$$

- **Root Mean Squared Error (RMSE):** Penalizes larger errors more heavily, making it sensitive to outliers.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.20)$$

- **Mean Absolute Percentage Error (MAPE):** Provides scale-independent comparison across different series.

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (2.21)$$

Lower values of RMSE, MAE, and MAPE indicate better predictive accuracy. [57, 58].

Conclusion

This chapter has established the theoretical foundations for temporal data analysis and forecasting, progressing from fundamental time series concepts to advanced machine learning approaches.

Building on these forecasting foundations, the following chapters will explore the practical implementation of forecasting models within the context of our case study. However, before delving into the implementation details, we will first present an overview of the Vehicle Routing Problem (VRP) and its solution approaches.

Chapter 3

The Vehicle Routing Problem: Foundations and Solution Approaches

Introduction

In real-world logistics, efficient delivery of goods using vehicle fleets represents a fundamental challenge in supply chain operations, directly impacting costs, service levels, and resource utilization. This challenge is formalized as the Vehicle Routing Problem (VRP), a cornerstone problem in combinatorial optimization that has received extensive attention due to its practical importance.

This chapter establishes the theoretical foundations for VRP analysis and solutions. We begin with the mathematical optimization framework, examine computational complexity theory that classifies problem difficulty, present the classical VRP formulation and its main variants, and conclude with an overview of solution methodologies.

3.1 Foundations of Operations Research and Combinatorial Optimization

“True optimization is the revolutionary contribution of modern research to decision processes.”

- George Dantzig [59]

3.1.1 Operations Research (OR)

Operations Research (OR) is a discipline that applies mathematical and statistical methods to analyze complex decision problems and optimize outcomes. It provides systematic approaches to resource allocation, scheduling, and strategic planning in business and organizational contexts [10].

Figure 3.12 presents the OR modeling process, transforming real-world inputs and constraints into mathematical models that generate optimal solutions for practical implementation.

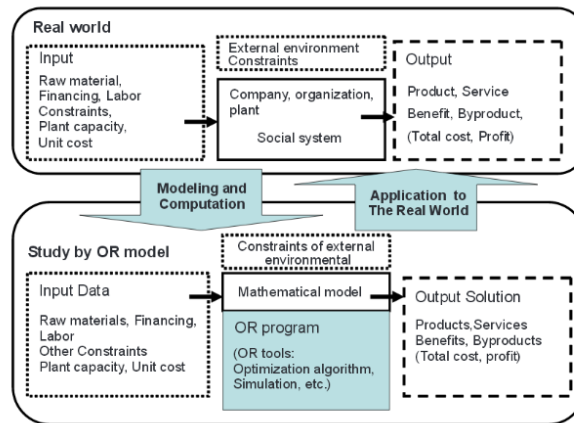


Figure 3.12: Concept of study by OR model [10].

The methods of operational analysis that are frequently used in logistics, production enterprises including the following methods: the graph and networks theory (scheduling tasks in production logistics), linear programming (optimization of transport distances, transport costs, etc.), methods of network analysis, inventory theory models[10].

3.1.2 Combinatorial Optimization (CO)

Optimization: refers to the process of determining the maximum or minimum of a given function, known as the objective function, over a defined domain. [60]

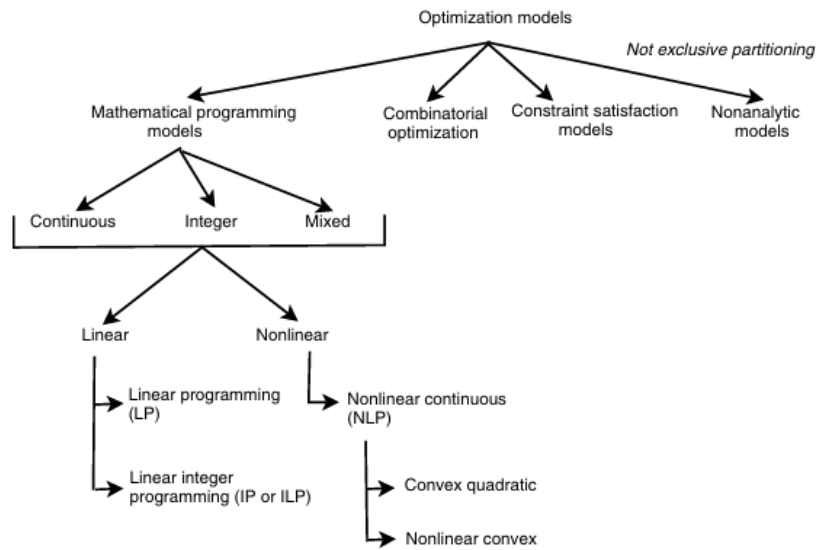


Figure 3.13: Classical optimization methods [11].

Combinatorial optimization: is a subfield of mathematical optimization, which seeks to find an optimal object from a finite set of feasible solutions, where the goal is to identify the solution that optimizes a given objective function among all possible candidate solutions.

Common problems include the Minimum Spanning Tree problem, which finds a spanning tree with minimum total weight in a connected graph; the Traveling Salesman Problem, which seeks the shortest route visiting all cities exactly once; the Knapsack Problem, which maximizes value while respecting weight constraints [59].

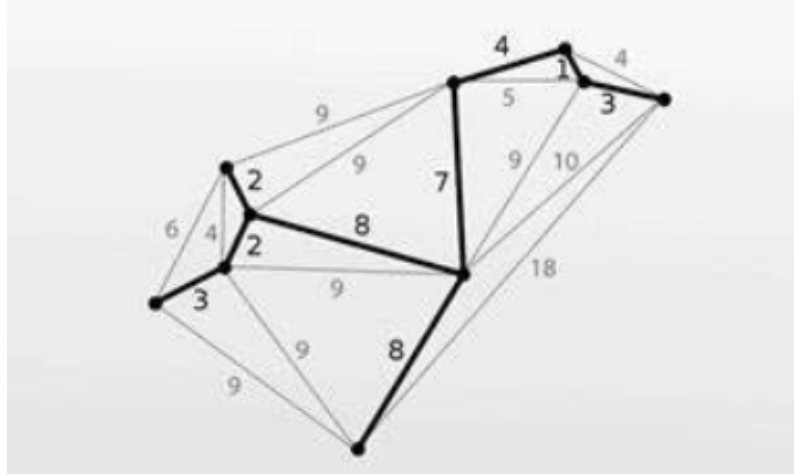


Figure 3.14: Graph representation of the combinatorial optimization problem of finding a Minimum Spanning Tree (MST)[12].

Mathematically, a combinatorial optimization problem can be formulated as:

$$\begin{array}{ll} \text{Minimize (or Maximize)} & f(x) \\ \text{subject to} & x \in F \end{array}$$

where x is a feasible solution from the finite set F of admissible configurations, and $f(x)$ is a cost or objective function to be optimized. The goal is to find a solution $x^* \in F$ such that $f(x^*) \leq f(x)$ (or $f(x^*) \geq f(x)$) for all $x \in F$ [61].

A fundamental challenge in CO is the rapid growth of the solution space, commonly referred to as the *combinatorial explosion*, which makes exhaustive enumeration computationally infeasible as problem size increases [60].

3.1.3 Computational Complexity

An algorithm requires two essential resources to solve a problem: time and space. The time complexity represents the number of steps required to solve a problem of size n , generally expressed in terms of worst-case analysis.

The aim of computational complexity analysis is to establish an asymptotic bound on the step count rather than determine the exact number of operations. In this regard, Big-O notation is a standard tool used to formalize asymptotic analysis [11].

3.1.4 Complexity of Problems

The complexity of a problem is defined as the complexity of the most efficient algorithm capable of solving it. A problem is considered tractable (or easy) if it can be solved by an algorithm operating in polynomial time. Conversely, a problem is deemed intractable (or difficult) when no polynomial-time algorithm exists to solve it.

The complexity theory of problems deals with decision problems. A decision problem always has a yes or no answer [11].

Computational complexity theory provides a framework for classifying problems according to their inherent difficulty. Problems are grouped into complexity classes, including:

- **P**: The class of decision problems that can be solved quickly, specifically, problems that have polynomial-time algorithms that can find solutions from scratch.
- **NP**: The class of decision problems for which a given solution can be verified in polynomial time.
- **NP-complete**: Decision problems that are both in NP and are as hard as any problem in NP. If any NP-complete problem can be solved in polynomial time, then $P = NP$ and all problems in NP can be solved in polynomial time.
- **NP-hard**: Problems that are at least as hard as the hardest problems in NP. NP-hard problems do not have to be in NP and may not even be decision problems.

The relationship among these complexity classes is central to understanding computational intractability, with the P versus NP question being one of the most important open problems in computer science [62].

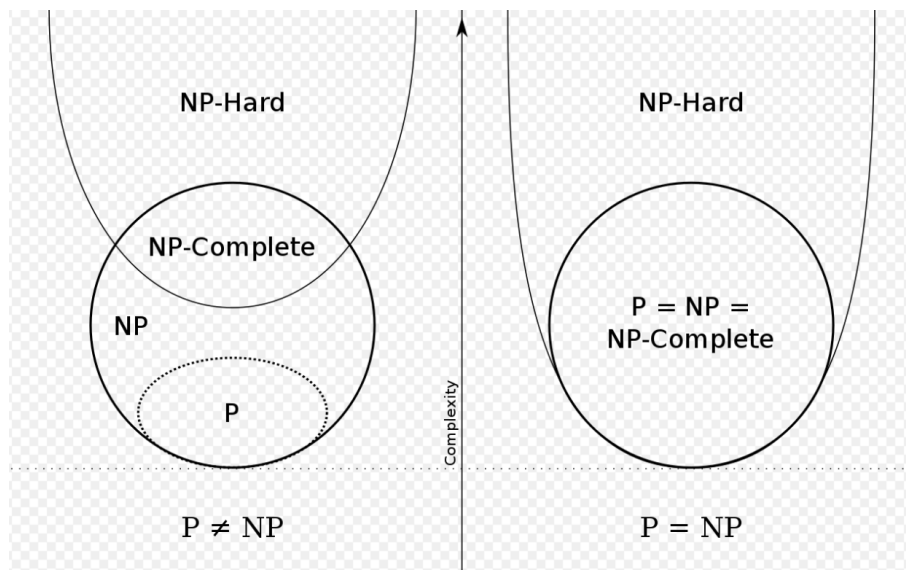


Figure 3.15: Relationship between complexity classes P, NP, NP-Complete, and NP-Hard. Adapted from [13].

3.2 Vehicle Routing Problem (VRP)

Among the various combinatorial optimization problems, the Vehicle Routing Problem (VRP) represents a particularly challenging domain. Toth and Vigo [63] emphasize

that this problem, defined more than 60 years ago, involves designing optimal route sets for vehicle fleets serving customer locations. The problem is one of routing the vehicles so that the total distance traveled (or time taken, cost incurred, etc.), is minimal [64].

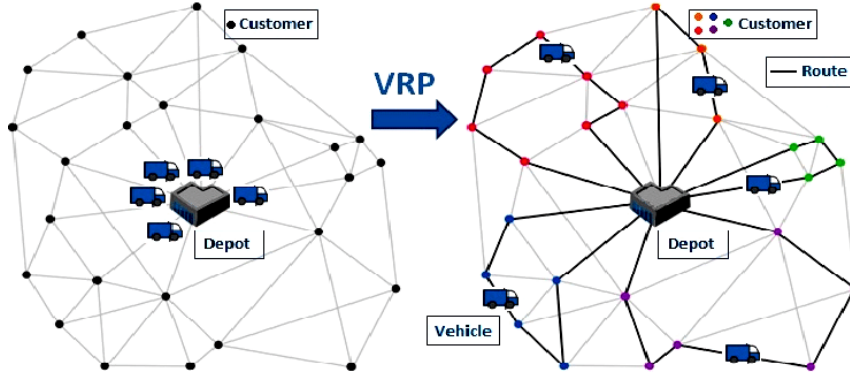


Figure 3.16: Graphical representation of a typical Vehicle Routing Problem (VRP) instance [14].

The VRP is used in supply chain management in the physical delivery of goods and services. There are several variants to the VRP. These are formulated based on the nature of the transported goods, the quality of service required, and the characteristics of the customers and the vehicles [65].

3.2.1 Mathematical Modeling of the Vehicle Routing Problem

The Capacitated Vehicle Routing Problem (CVRP), first formulated by Dantzig and Ramser [66], represents the foundational variant in vehicle routing optimization. The Capacitated Vehicle Routing Problem (CVRP) represents the most common variant, where capacity restrictions are imposed on the vehicles.

Following Laporte [67], the CVRP is defined on a graph $G = (V, A)$ where $V = \{1, \dots, n\}$ is a set of vertices representing customers with the depot located at vertex 1, and A is the set of arcs. The problem parameters are:

- $V = \{1, \dots, n\}$ is the set of vertices, with vertex 1 representing the depot and vertices $\{2, \dots, n\}$ denoting customer locations;
- A is the set of arcs with associated non-negative distance matrix $C = (c_{ij})$;
- $d_i \geq 0$ represents the demand of customer $i \in \{2, \dots, n\}$;
- D denotes the uniform capacity of each identical vehicle;
- m represents the number of available vehicles, where $m^L \leq m \leq m^U$;
- $x_{ij} \in \{0, 1\}$ is a binary decision variable equal to 1 if arc (i, j) is traversed in the optimal solution.

The CVRP seeks to design a set of least-cost vehicle routes such that [67]:

1. Each customer in $V \setminus \{1\}$ is visited exactly once by exactly one vehicle;
2. All vehicle routes start and end at the depot;
3. The total demand on any route does not exceed the vehicle capacity D .

Mathematical Formulation

Following the formulation presented by Laporte [67], the CVRP can be expressed as the following integer linear program:

Objective Function:

$$\min \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} c_{ij} x_{ij} \quad (3.22)$$

Subject to:

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ij} = 1 \quad \forall i \in \{2, \dots, n\} \quad (3.23)$$

$$\sum_{\substack{i \in V \\ i \neq j}} x_{ij} = 1 \quad \forall j \in \{2, \dots, n\} \quad (3.24)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - v(S) \quad \forall S \subseteq V \setminus \{1\}, |S| \geq 2 \quad (3.25)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V, i \neq j \quad (3.26)$$

where $v(S)$ represents an appropriate lower bound on the number of vehicles required to visit all vertices of S in the optimal solution. For the CVRP, it is valid to take:

$$v(S) = \left\lceil \frac{\sum_{i \in S} d_i}{D} \right\rceil$$

Constraint Analysis

Constraints (3.23) and (3.24) ensure that each customer is visited exactly once. The subtour elimination constraints (3.25) serve a dual purpose: they eliminate subtours that do not include the depot and enforce vehicle capacity restrictions by ensuring that any subset S of customers requires at least $v(S)$ vehicles to satisfy the total demand [67].

3.2.2 Taxonomy of VRP Variants

VRP research has expanded into diverse variants that reflect the varied constraints and objectives found in different distribution contexts.

Table 3.4 presents the most prominent VRP variants, organized by six key feature categories. Each category captures a unique aspect of logistical complexity and helps both practitioners and researchers identify the most appropriate VRP formulation for a given context.

Table 3.4: Classification of VRP Variants by Key Features

Feature Category	Main VRP Variants	Description
Fleet Configuration	• CVRP (Capacitated VRP) • HFVRP (Heterogeneous Fleet VRP) • FSVRP (Fleet Size VRP)	Describes the nature and structure of the vehicle fleet, including identical vs. heterogeneous vehicles, fleet size constraints [68, 15]
Depot Structure	• SDVRP (Single-Depot VRP) • MDVRP (Multi-Depot VRP) • Open VRP	Defines how goods are dispatched: from a single depot [69], multiple depots [70], or with vehicles not returning to the depot (open VRP) [71].
Customer Service Constraints	• SDVRP (Split Delivery VRP) • VRPPD (VRP with Pickups and Deliveries)	Covers scenarios when the single-visit assumption is relaxed and each customer may be served by more than one vehicle. [72] and the case when the vehicles deliver goods to customers and pick up items during the same visit [73].

Table 3.4 (continued)

Feature Category	Main VRP Variants	Description
Temporal Factors	• VRPTW (VRP with Time Windows) • TDVRP (Time-Dependent VRP) • DVRP (Dynamic VRP)	Accounts for delivery time windows [65], traffic-dependent travel times [74], and dynamic or real-time routing adjustments [75].
Uncertainty and Robustness	• SVRP (Stochastic VRP) • RVRP (Robust VRP)	Deals with uncertainty in demand or travel time using probabilistic or robust optimization approaches [15].
Integrated and Multi-Level Planning	• 2E-VRP (Two-Echelon VRP) • LRP (Location-Routing Problem) • IRP (Inventory-Routing Problem) • MPVRP (Multi-Period VRP) • MTVRP (Multi-Trip VRP)	Two-tier distribution with depot-to-satellite and satellite-to-customer routing [76], multi-level supply chain optimization integrating location, inventory, and green routing [77]. Vehicles performing multiple tours per day [78], or multi-period planning [79].

The classification presented in Table 3.4 establishes the theoretical foundation for understanding VRP variants. To complement this, Figure 3.17 provides a practical decision-making framework that guides the selection of appropriate VRP formulations based on problem-specific requirements and operational constraints.

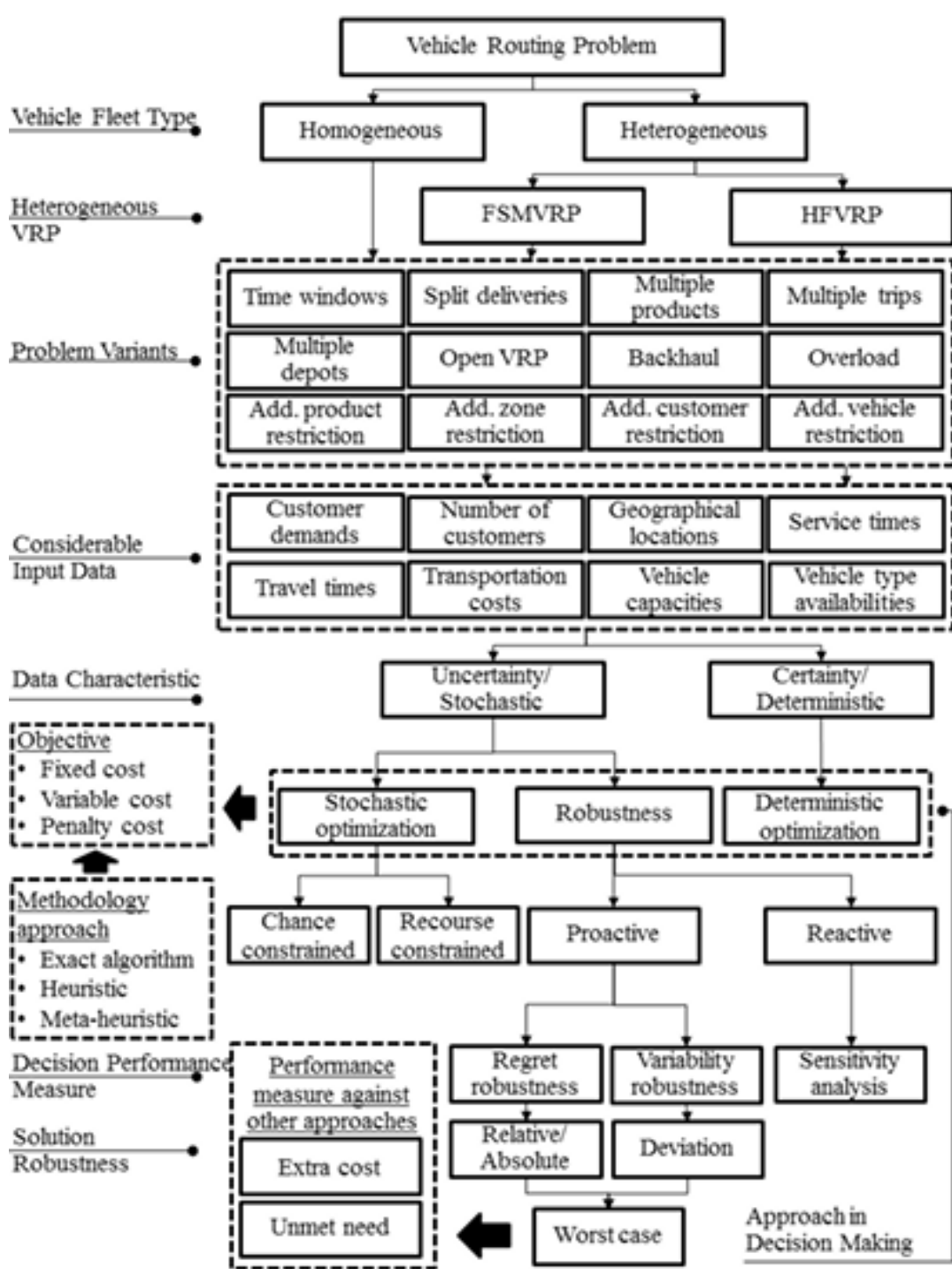


Figure 3.17: Decision framework for Vehicle Routing Problem variant selection based on problem characteristics and constraints [15]

Figure 3.18 presents the distribution of VRP variant research during the study period (2010-2020) based on Konstantakopoulos article [17], highlighting the relative research attention given to different problem formulations.

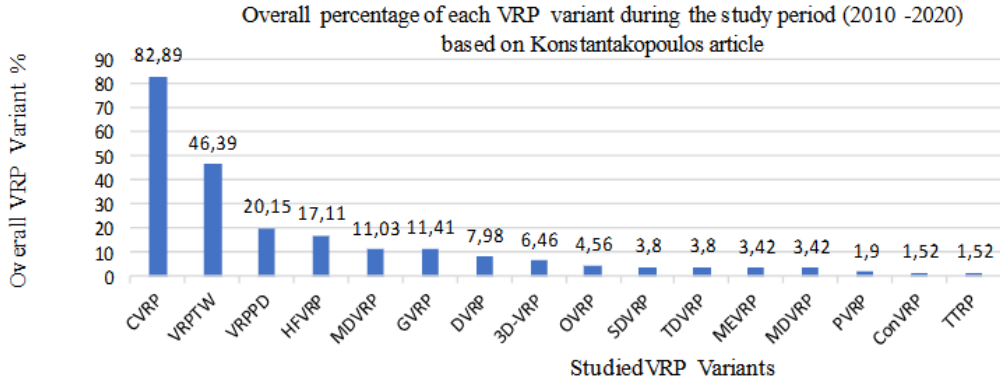


Figure 3.18: Overall percentage of each VRP variant during the study period (2010-2020) [16]

3.2.3 Vehicle Routing Problem Variants

Building upon the foundational CVRP model, numerous variants have been developed to address real-world complexities and operational requirements. In the following, we present the extensions most relevant to our study.

Heterogeneous Fleet Vehicle Routing Problem (HFVRP)

The HFVRP is an extension of the classical CVRP where the fleet is composed of vehicles with different capacities, fixed costs, and variable costs. The objective is to minimize the total distribution cost while determining both the assignment of customers to vehicles and the routing of those vehicles. Applications include freight transport, postal distribution, and supply chains where fleets are composed of trucks with heterogeneous features. [80]

Decision variable:

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

Main constraint (vehicle capacity constraint):

$$\sum_{v \in R_i} q_v \leq \bar{q}_t, \quad \forall \text{ routes } R_i \text{ assigned to vehicle type } t$$

where q_v is the demand of customer v and $\bar{q}_t$ is the maximum capacity of a vehicle of type t .

This ensures that each vehicle route does not exceed its specific vehicle capacity. [80]

Vehicle Routing Problem with Time Windows (VRPTW)

The Vehicle Routing Problem with Time Windows (VRPTW) extends the classical VRP by introducing service time constraints for each customer. Applications include

parcel delivery, home healthcare logistics, and transportation services under strict scheduling constraints. [81]

Decision variable:

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

Main constraint (service within time windows):

$$e_i \leq \text{start}_i \leq l_i, \quad \forall i \in \{1, 2, \dots, N\}$$

This constraint ensures that service at each customer i starts within the predefined time window $[e_i, l_i]$. [81]

Split Delivery Vehicle Routing Problem (SDVRP)

The goal of the SDVRP is to design a set of vehicle routes to serve all customer demands at minimum cost, allowing each customer to be visited more than once (i.e., demands can be split among vehicles). Applications include waste collection, newspaper distribution, and supply chain logistics. [82]

Decision variables:

$$x_{ij}^k = \begin{cases} 1 & \text{if vehicle } k \text{ travels directly from node } i \text{ to node } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$q_{ik} \geq 0 \quad \text{quantity delivered to customer } i \text{ by vehicle } k.$$

Main constraint (demand satisfaction with split deliveries):

$$\sum_{k=1}^M q_{ik} = d_i, \quad \forall i \in N \setminus \{0\}$$

This constraint ensures that the demand of each customer i is fully satisfied, but it may be split among multiple vehicles. [82]

Open Vehicle Routing Problem (OVRP)

The goal of the OVRP is to design a set of Hamiltonian paths (open routes) to serve customer demand. Applications: delivery of school meals, school bus routing, the home delivery of packages and newspapers trains, among others. [83]

Decision variable:

$$x_{ij} = \begin{cases} 1 & \text{if the arc from node } i \text{ to node } j \text{ is used by a vehicle,} \\ 0 & \text{otherwise.} \end{cases}$$

Main constraint (customer visit and route continuity):

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

This constraint ensures that each customer node is visited exactly once, while also maintaining route continuity: if a vehicle enters a node, it must leave it. [84]

3.2.4 Multi-Trip Vehicle Routing Problem (MTVRP)

The **Multi-Trip Vehicle Routing Problem (MTVRP)** is an extension of the classical Vehicle Routing Problem (VRP), in which a single vehicle can perform multiple successive tours within the same planning horizon. After completing a tour and returning to the depot, the vehicle can depart again to serve additional customers. This feature increases fleet utilization efficiency and helps reduce overall transportation costs [85].

Decision Variable

$$x_{ij}^{k,r} = \begin{cases} 1 & \text{if vehicle } k \in K \text{ directly travels from node } i \text{ to node } j \text{ during trip } r \in R, \\ 0 & \text{otherwise.} \end{cases}$$

where $i, j \in N$ denote the nodes (customers and depot).

Main Constraint (multi-trips)

Each customer $i \in N \setminus \{0\}$ must be served exactly once by a vehicle, regardless of the trip $r \in R$:

$$\sum_{k \in K} \sum_{r \in R} \sum_{j \in N} x_{ij}^{k,r} = 1 \quad \forall i \in N \setminus \{0\}$$

This constraint ensures that each customer is visited exactly once, even though the same vehicle is allowed to perform multiple distinct trips.

3.3 Solution Approaches for VRP

3.3.1 Theoretical Complexity of VRP Variants

The classical Vehicle Routing Problem (VRP) is proven to be NP-hard [86]. The complexity of the problem grows exponentially with the problem size, making exact solutions computationally intractable for larger instances.

While exact approaches can solve small to medium-sized instances with optimality guarantees, they suffer from exponential computational complexity as problem size increases. Given this limitation, heuristic and metaheuristic methods have become the predominant solution approach in practice.

These approximate methods sacrifice optimality guarantees in exchange for computational tractability, enabling practitioners to obtain high-quality solutions for large-scale problems within a reasonable time [11].

3.3.2 Exact vs. Approximate Methods for Solving VRP

Various solution strategies have been developed to cope with the complexity of the problem. These approaches span a wide methodological spectrum, as illustrated in

Figure 3.19, which summarizes the principal families of algorithms commonly used to tackle the VRP and its extensions, adapted from Labadie et al. (2016) and Lin et al. (2014).

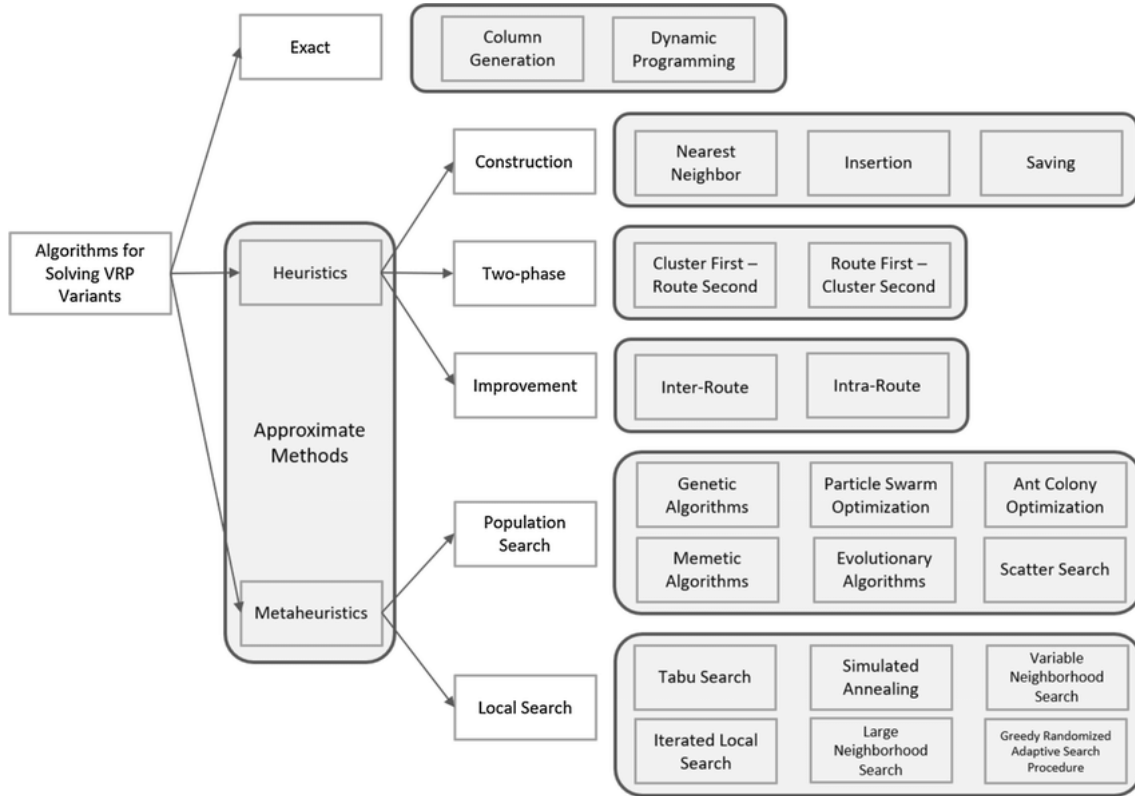


Figure 3.19: Classification of algorithms for the VRP [17]

Exact Methods

Within the category of exact methods, several classical algorithms can be identified. These include dynamic programming as well as the family of branch-and-X algorithms, notably branch and bound, branch and cut, and branch and price, which were developed within the field of operations research, and used for solving Integer Linear Programming (ILP) and Mixed Integer Linear Programming problems (MILP) where the problem is solved by subdividing it into simpler problems [11].

Dynamic programming: Dynamic programming is a method that relies on the recursive decomposition of a problem into simpler subproblems. It is grounded in Bellman’s principle of optimality, which states that “a subpolicy of an optimal policy is itself optimal”. As a stage-wise optimization approach, it builds the solution through a sequence of partial decisions. By systematically discarding partial decision paths that cannot yield the optimal result, dynamic programming significantly reduces the need for exhaustive enumeration of the entire search space. [11]

Branch and Bound (B&B): The branch and bound algorithm relies on the implicit enumeration of all feasible solutions to the given optimization problem. The exploration process is represented through a dynamically constructed search tree, where the root node corresponds to the original problem and its complete solution space. Internal nodes denote subproblems, while leaf nodes represent potential solutions. The efficiency of the method stems from the use of a bounding function, which eliminates subtrees that cannot contain an optimal solution, thereby reducing the search effort [11].

Branch and Cut (B&C): Branch and Cut is the extension of Branch and Bound with cutting planes. Cutting plane approaches have been proposed in 1958 by Gomory which consist in iteratively adding some specific constraints to the LP-relaxation of the problem, providing tighter bounds by making the linear programming polytope more closely approximate the convex hull of all feasible integer solutions [11].

Branch and Price (B&P): Branch and Price is the extension of Branch and Bound with column generation. Column generation has been first applied by Gilmore and Gomory which uses Dantzig-Wolfe decomposition to generate a decomposition of the problem into master and subproblems. handling problems with a huge number of variables [11].

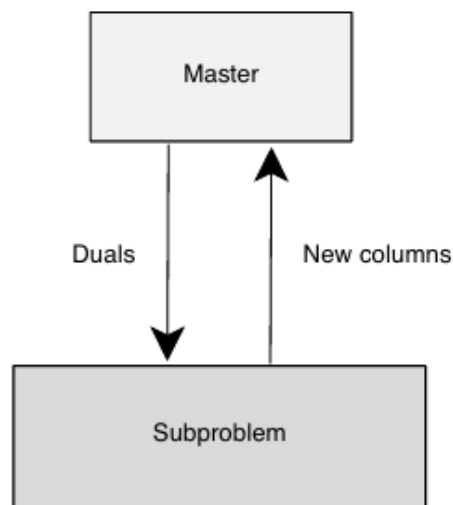


FIGURE 5.18 Branch and price approach.

Figure 3.20: Branch and Price Approach [11]

Figure 3.20 shows the two-way communication in column generation: dual information flows down to guide the search for new variables, while promising new columns flow up to improve the master solution. This iterative exchange continues until optimality is reached.

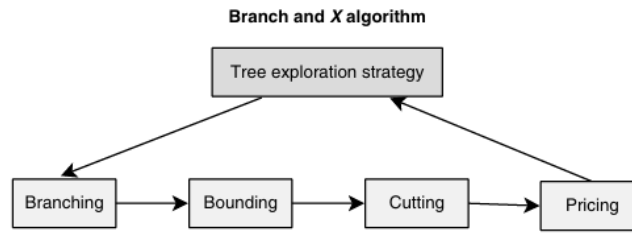


Figure 3.21: Evolution of Branch and X algorithms. [11]

These methods are powerful but often impractical for large-scale real-world instances, motivating the use of heuristics and metaheuristics.

Heuristic Methods

Heuristics provide high-quality solutions at reasonable computational cost with good generalization ability, making them popular among researchers and practitioners. They have dominated vehicle routing literature, comprising over 80% of publications between 2009 and 2013 [87]. Heuristics can be classified into three main categories:

Constructive heuristics: build routing solutions from scratch following fixed empirical procedures.

Improvement heuristics: iteratively enhance existing solutions through local search in neighborhoods. Though efficient at finding local optima, they can become trapped in local optimal solutions, with final solution quality dependent on the starting point [87].

Understanding Local Optima: A *local optimum* is a solution that is better than all of its immediate neighbors but is not necessarily the best overall solution (i.e., the global optimum). In routing problems, this means the algorithm may settle on a solution where no small change improves the result, even if a better solution exists elsewhere in the solution space. This limitation motivates the use of metaheuristics, which incorporate strategies to escape from local optima.

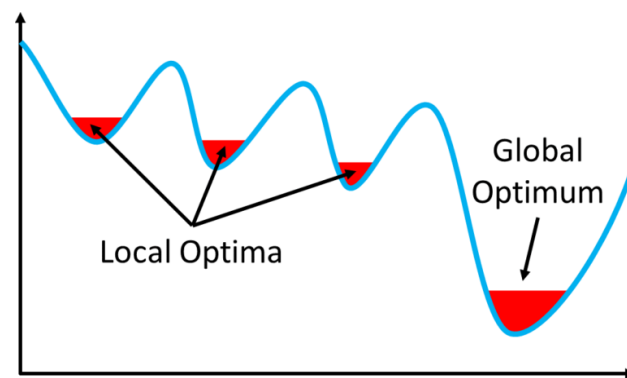


Figure 3.22: Example of a local and global optimum [18]

Metaheuristic Methods

Differ from problem-specific heuristics by providing high-level algorithmic principles that are less problem-dependent. Drawing inspiration from natural phenomena or physical processes, they typically offer efficient performance with global search capabilities, avoiding the local optima trap that limits improvement heuristics, but there is no guarantee to find global optimal solutions [87].

Conclusion

In this chapter, we have established the fundamental theoretical framework for the Vehicle Routing Problem and its variants, examining modeling structures, computational complexities, and solution methodologies. We explored both exact and approximate solution approaches, highlighting how exact methods like dynamic programming and branch-and-bound guarantee optimal solutions, while heuristics and metaheuristics offer computational efficiency at the cost of optimality.

These theoretical foundations will later support the development of an optimal routing solution for fresh dairy distribution at Celia.

Chapter 4

Exploratory Data Analysis and Forecasting Model Implementation

Introduction

Forecasting customer demand is a fundamental component of data-driven decision-making in logistics and supply chain management. Reliable forecasts support more efficient planning and enable better allocation of resources such as transportation capacity. To build effective predictive models, it is essential to first understand the structure and behavior of the underlying data, and to apply appropriate preprocessing techniques that enhance model performance and reliability.

This chapter presents a comprehensive workflow that begins with the analysis and preparation of the dataset, followed by the development of forecasting models tailored to the business objective. The first part focuses on data exploration, including the identification of key patterns and statistical properties. Preprocessing techniques are applied to clean, transform, and structure the data in a way that supports robust modeling. The second part introduces the predictive modeling approach, describing the methods used to generate forecasts and evaluate their accuracy.

4.1 Dataset Description

The dataset used in this study consists of customer order records collected between **January 2, 2020** and **March 19, 2025**, spanning a total of **7,833 daily non-zero entries**. Each record is defined by three fields: the *order date*, the *city (customer location)*, and the *order quantity* in pallets. The dataset includes a total of **37 distinct cities**, each representing a different delivery destination.

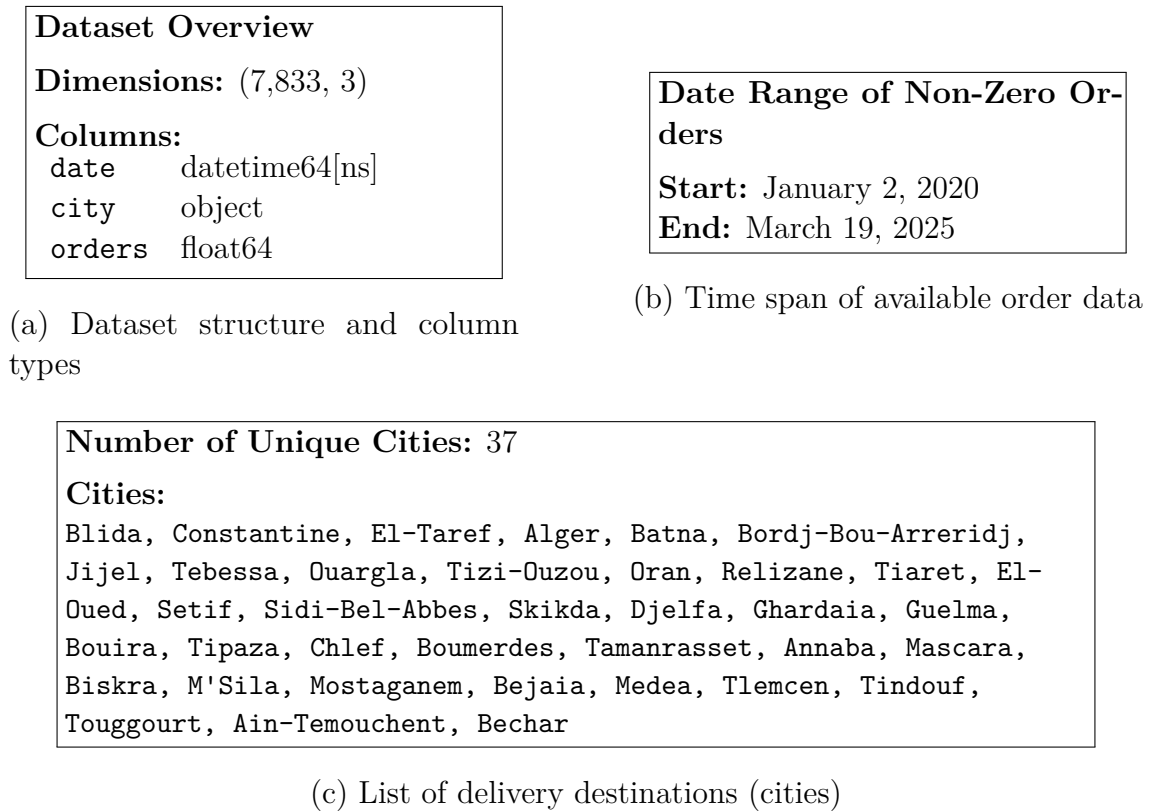


Figure 4.23: Initial overview of the dataset: structure, temporal coverage, and cites.

To ensure meaningful analysis, we first assess the quality and statistical characteristics of the data.

4.2 Exploratory Data Analysis (EDA)

This section aims to explore the key characteristics of the dataset in order to gain preliminary insights into the temporal and spatial patterns of customer orders.

4.2.1 Data Quality Assessment

We first examined the dataset for missing or invalid values. We found that it contains **no missing values** across its three key columns: *date*, *city*, and *order quantity*. This confirms the dataset is complete and suitable for further statistical and visual analysis.

Table 4.5: Missing Values per Column

Column	Missing Values	Data Type
date	0	datetime64[ns]
city	0	object
orders	0	float64

4.2.2 Descriptive Statistics of Order Quantities

The Table 4.6 presents the key descriptive statistics for the `orders` variable.

Table 4.6: Summary Statistics for Non-Zero Order Quantities

Count	Mean	Std	Min	25%	Median	75%	Max
7,833	11.25	12.05	0.004	5.00	8.37	13.83	98.390000

This wide range reflects a high variability in demand, from very small to exceptionally large orders, suggesting that both routine and peak logistics scenarios must be considered in the planning process.

4.2.3 Distribution of Order Quantities

To better understand the magnitude and frequency of order sizes, we analyzed their distribution. Figure 4.24 displays a histogram with a Kernel Density Estimation (KDE) overlay. The data reveals a distinctly **right-skewed** distribution, with the majority of orders concentrated between **1 and 15 pallets**. The highest frequency occurs around **6 to 8 pallets**, indicating that small to medium-sized orders are the most common. As the number of pallets in an order increases, such large orders become much less common. Orders with more than **40 pallets** are quite rare, and those exceeding **60 pallets** are exceptional.

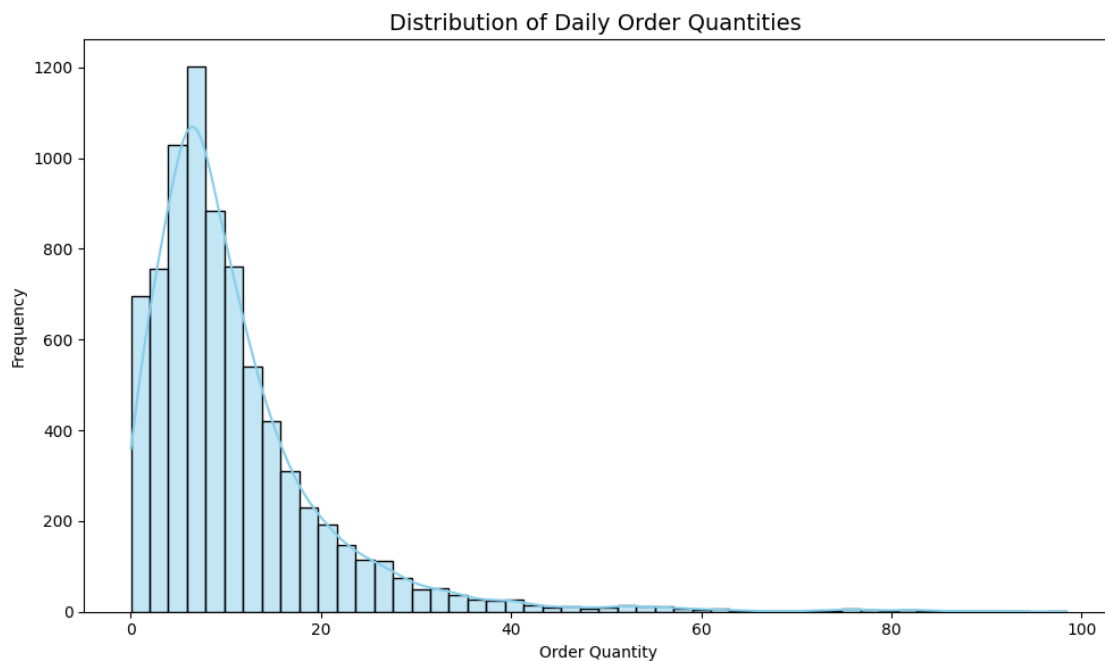


Figure 4.24: Distribution of Non-Zero Order Quantities

While visual inspection already suggests a skewed distribution, a formal goodness-of-fit analysis was performed to determine which theoretical probability distribution

best matches the observed order data. Several candidate distributions were fitted using **Maximum Likelihood Estimation (MLE)** (**R. A. Fisher, 1922**). to estimate their parameters such as mean, standard deviation, shape, and scale by maximizing the likelihood of observing the actual data under each distribution.

To evaluate the performance of each fitted model, we used two model selection criteria that balance fit quality and model complexity:

- **Akaike Information Criterion (AIC)**(**H. Akaike, 1974**):

$$\text{AIC} = 2k - 2\ln(L) \quad (4.27)$$

where k is the number of model parameters, and L is the maximized likelihood. A lower AIC indicates a better trade-off between model complexity and goodness-of-fit.

- **Bayesian Information Criterion (BIC)**(**G. Schwarz, 1978**):

$$\text{BIC} = k \ln(n) - 2\ln(L) \quad (4.28)$$

where n is the number of observations. BIC penalizes complexity more than AIC and also favors models with lower values.

The results, summarized in Table 4.7, show that the **log-normal distribution** provided the best fit, achieving the lowest AIC and BIC scores. This confirms the presence of a heavy-tailed, positively skewed distribution, consistent with the observed concentration of small orders and occasional large peaks.

Table 4.7: Comparison of Distribution Fits Based on Log-Likelihood, AIC, and BIC

Distribution	Log-Likelihood	AIC	BIC
Log-Normal	-26 319.23	52 644.46	52 665.36
Weibull	-26 421.20	52 848.39	52 869.29
Exponential	-26 607.20	53 218.41	53 232.34
Gumbel	-26 867.74	53 739.48	53 753.41
Normal	-29 265.52	58 535.04	58 548.97
Gamma	-59 115.47	118 236.94	118 257.84

4.2.4 Outliers Detection

Boxplot Analysis of Daily Order Volumes Figure 4.25 presents a boxplot of daily order volumes (in pallets). The median is around **42 pallets**, and 50% of the values fall between **35 and 50 pallets**, showing stable daily demand. Some outliers exceed **100 pallets**, indicating occasional peaks. Overall, daily volumes are more consistent than individual orders.

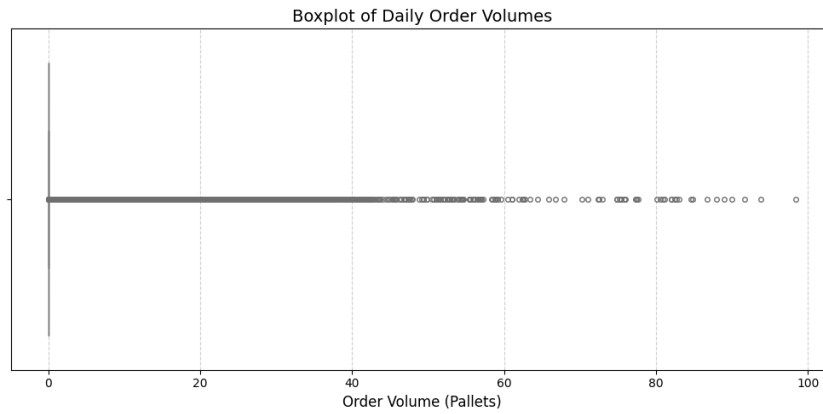


Figure 4.25: Boxplot of daily order volumes highlighting outliers.

While daily aggregates reveal operational patterns, analyzing individual order distributions provides deeper insights into customer behavior and logistics planning requirements. Figure 4.26 shows order volume distributions for the top 10 customers by total volume, representing the most significant contributors to logistics operations. The analysis reveals considerable variation in individual order patterns, with customers like Alger showing both high-volume orders and significant outliers that require special handling considerations.

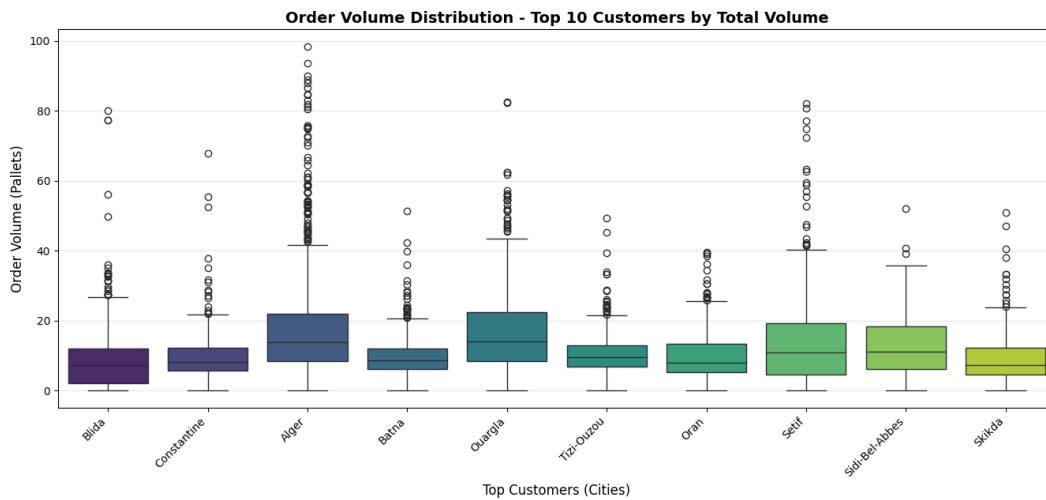


Figure 4.26: Order volume distribution for top 10 customers using boxplot

The IQR Method Analysis To validate the presence of outliers visually observed in the boxplot, we applied the **Interquartile Range (IQR)** method (John W. Tukey, 1977).

The **IQR** is a statistical measure that represents the range between the first quartile ($Q1$) and the third quartile ($Q3$) of a dataset:

$$IQR = Q3 - Q1$$

In our dataset:

- $Q1 = 5.0$ pallets
- $Q3 = 13.82$ pallets
- $IQR = 13.82 - 5.0 = 8.82$ pallets

According to the IQR method, any value above the upper bound:

$$\text{Upper Bound} = Q3 + 1.5 \times IQR = 13.82 + (1.5 \times 8.82) = 27.05$$

is considered an outlier. Therefore, any order exceeding **27.05 pallets** is identified as an outlier in this dataset.

A total of **469 order-level outliers** were identified representing approximately **6%** of all individual orders. While statistically extreme, these high-volume requests are **operationally significant**. Table 4.8 lists the five most extreme order volumes, all of which significantly exceed the IQR-based upper bound.

Table 4.8: Top 5 Order-Level Outliers Detected Using IQR

Date	City	Order Volume	× Threshold
2023-07-31	Alger	98.39	3.64×
2022-04-02	Alger	93.75	3.47×
2023-01-31	M'Sila	91.63	3.39×
2022-04-30	Alger	89.98	3.33×
2025-02-07	Alger	88.95	3.29×

The presence of a highly skewed distribution and extreme values highlights the need for deeper investigation before applying predictive models.

As a next step, we examine how demand varies across cities and over time to uncover spatial and temporal patterns that may inform both preprocessing and model design.

4.2.5 Order Volume by City

A detailed aggregation of the data reveals that certain cities contribute significantly more to overall demand. Alger leads with over 23,000 pallets across 1,259 active days, followed by Ouargla, Setif, and Sidi-Bel-Abbès. These cities not only show high total volumes but also consistent ordering activity. Table 4.9 summarizes the top 10 cities by total order volume, along with descriptive statistics such as mean, standard deviation, and range.

Table 4.9: Top 10 Cities by Total Order Volume (Pallets)

City	Count	Mean	Std	Min	Max	Sum
Alger	1259	18.51	19.83	0.06	98.39	23300.89
Ouargla	444	17.72	14.34	0.03	65.55	7868.64
Setif	439	15.01	16.56	0.03	58.90	6589.11
Sidi-Bel-Abbes	474	12.90	8.66	0.01	52.15	6115.61
Blida	580	9.08	11.10	0.01	62.29	5267.21
Tizi-Ouzou	472	10.46	6.15	0.05	49.39	4935.15
Oran	453	9.98	7.01	0.16	39.68	4522.89
Batna	454	9.90	6.08	0.10	51.41	4493.35
Constantine	382	9.88	7.21	0.01	67.86	3772.64
Skikda	359	9.33	7.19	0.01	51.05	3350.53

To visualize the distribution of demand, bar plots were generated showing the top cities by total volume and average order size.

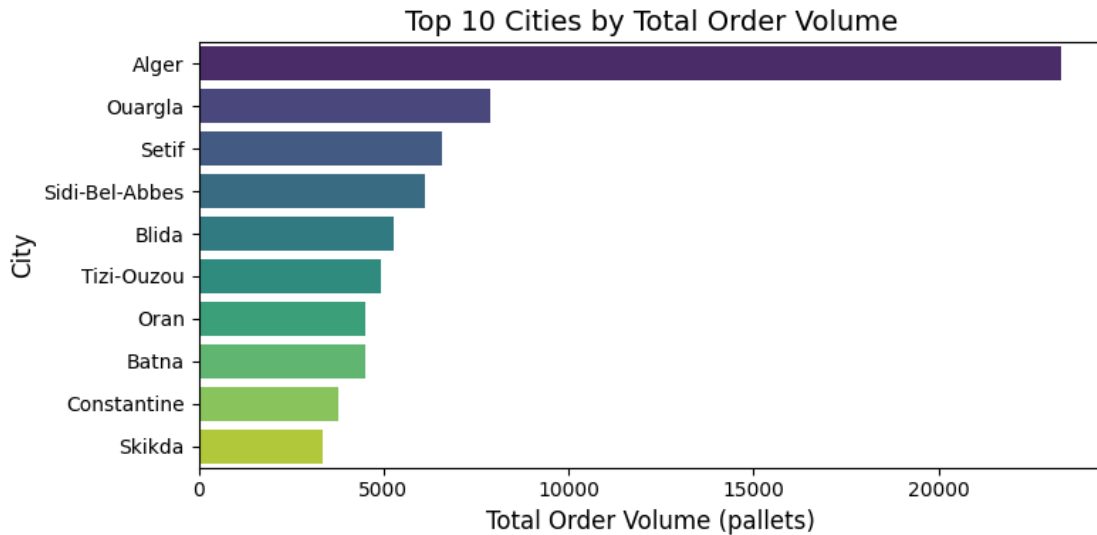


Figure 4.27: Top 10 cities by total order volume (in pallets).

4.2.6 Temporal Patterns and Demand Trends

Understanding how customer demand evolves over time is essential for effective forecasting and logistics planning. This section explores the temporal dynamics of order volumes by analyzing daily, monthly, and yearly trends.

Yearly Demand Evolution

Figure 4.28 presents the total order volume per year from 2020 to 2025. A clear upward trend is observed between 2020 and 2022, followed by sustained high volumes

in 2023 and 2024. The apparent drop in 2025 is due to data availability being limited to mid-March, not an actual decline in demand.

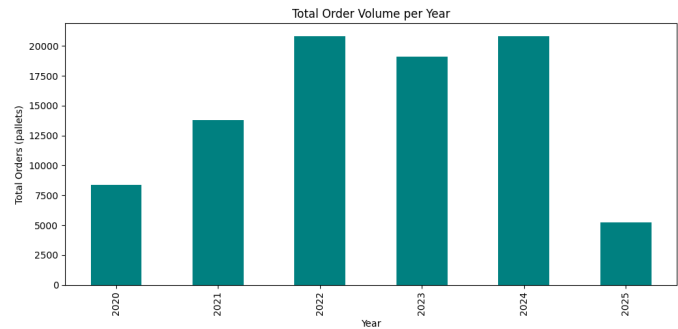


Figure 4.28: Total order volume per year.

Monthly Seasonality

Figure 4.29 displays the average monthly order volumes. While relatively stable across the year, slightly higher volumes are noted in January, November, and December. April records the lowest average, hinting at a potential seasonal slowdown. Overall, the demand profile remains consistent, with moderate seasonal effects.

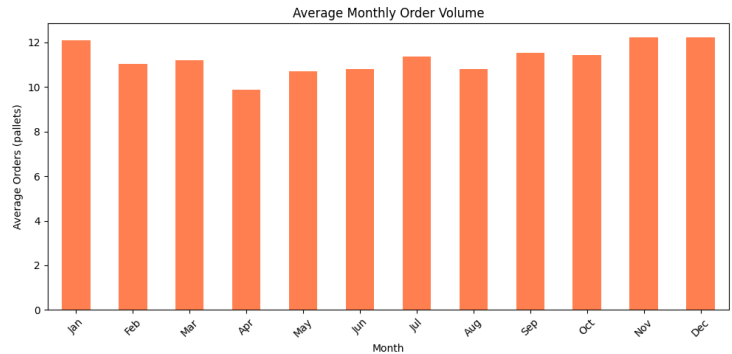


Figure 4.29: Average monthly order volume (seasonal patterns).

Weekly Demand Patterns

As shown in Figure 4.30, average daily orders remain relatively stable throughout the week. Thursday records the highest average volume, while Wednesday shows the lowest. These variations are minor, suggesting no strong weekday-specific demand trends.

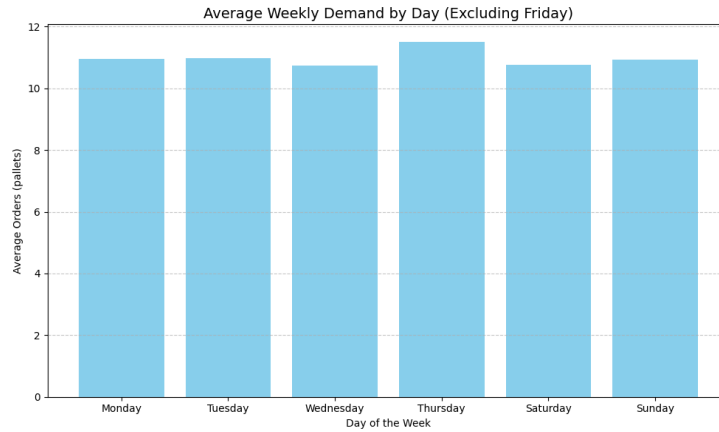


Figure 4.30: Average daily order volume by weekday.

Having established a general understanding of spatial and temporal demand patterns, we now focus on modeling the underlying time series behavior of daily order volumes. This analysis will assess the presence of trends, seasonality, and stationarity; key components that guide the selection of appropriate forecasting techniques.

4.2.7 Preprocessing for Time Series Consistency

The time series analysis requires a complete and continuous temporal index. However, in the original dataset, only days with non-zero orders were recorded, leading to implicit gaps that could distort trend and seasonal pattern detection. To address this, we reindexed the data to span the full date range from **January 2, 2020** to **March 19, 2025**, and explicitly assigned a value of zero to dates without recorded orders.

4.2.8 Sparsity of Demand and Zero-Dominated Behavior

Following the reindexing step to ensure a continuous daily timeline, we analyzed the frequency of days with zero orders. This analysis revealed a substantial sparsity in the dataset: **88.88%** of all days recorded no orders. Such a high proportion of zero-order days significantly impacts forecasting accuracy and requires specialized modeling approaches. This level of sparsity is typical in logistics where customers order irregularly rather than daily, which requires careful consideration in model design.

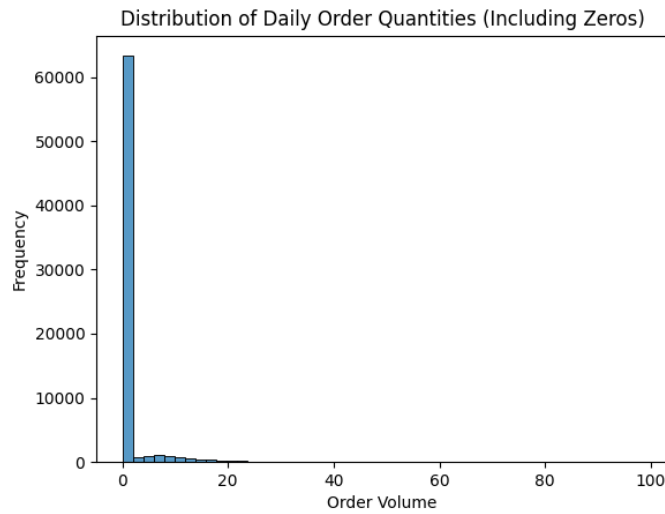


Figure 4.31: Distribution of Daily Order Quantities (Including Zeros)

As shown in Figure 4.31, these results provide strong evidence that the order behavior is highly irregular and dominated by inactivity periods.

4.2.9 Analysis of Zero-Order Patterns by City

After establishing that approximately **88%** of the observations in the dataset represent days with zero customer orders, it is crucial to investigate the origin of this sparsity. Specifically, we aim to detect potential heterogeneity in the activity levels of different cities. Identifying cities with consistently active demand patterns versus those with mostly inactive periods can inform decisions on customer segmentation and modeling strategies.

To this end, we computed the percentage of days with zero orders (%) for each city. Tables 4.10 and 4.11 present the results, based on the proportion of days with no recorded demand.

Table 4.10: Top 5 Cities with the Lowest Percentage of Zero Orders (Most Active)

City	Zero Orders (%)
Alger	33.88%
Blida	69.54%
Sidi-Bel-Abbes	75.11%
Tizi-Ouzou	75.21%
Batna	76.16%

Table 4.11: Top 5 Cities with the Highest Percentage of Zero Orders (Least Active)

City	Zero Orders (%)
Tindouf	99.95%
Relizane	99.89%
Touggourt	99.89%
Boumerdes	99.68%
Bechar	99.42%

The results highlight a significant disparity between cities: while a city like **Alger** shows relatively consistent activity with only ~34% zero-order days, others like **Tindouf** and **Relizane** show nearly ~100% inactivity. This analysis confirms the heterogeneity in the dataset and motivates a deeper understanding of their ordering behavior.

4.2.10 Customer Segmentation Based on Ordering Patterns

Given the high sparsity observed in the dataset, it is essential to distinguish between customers with regular ordering behavior and those with irregular or minimal activity. Some customers contribute substantially to demand, while others place orders infrequently or have long periods of inactivity.

To better characterize this heterogeneity and reduce the noise in modeling, a rule-based segmentation approach was adopted using key operational indicators such as:

- **Recency of Orders:** Days since the last recorded order.
- **Total Orders:** Cumulative number of orders over the observation period.
- **Order Frequency:** Average monthly and weekly ordering rates.
- **First and Last Order Dates:** Indicators of customer lifespan and engagement.

After analyzing the order behavior of each customer across several behavioral indicators, two principal categories were identified using this relative approach:

- **Active Customers:** Cities that show recent and relatively consistent ordering behavior over time, with a significant contribution to total demand when compared to others.
- **Rare Customers:** Cities with irregular, infrequent, or outdated ordering patterns, introducing volatility into the overall demand signal.

4.2.11 Segmentation Results

Out of the 37 cities (customers) analyzed, **24** were classified as *Active* and **13** as *Rare*, based on their ordering behavior. Table 4.12 summarizes the classification.

Table 4.12: Customer Segmentation Summary

Category	Number of Customers
Active	24
Rare	13

4.2.12 Visualizing Customer Ordering Behavior

To visually support the segmentation, we plotted key behavioral features: total orders, recency, average weekly frequency, and monthly frequency. The pairplot in Figure 4.32 clearly distinguishes the two segments.

Active customers (in red) tend to cluster around lower recency and higher frequency values, while Rare customers (in yellow) are more dispersed, often concentrated in areas representing lower activity.

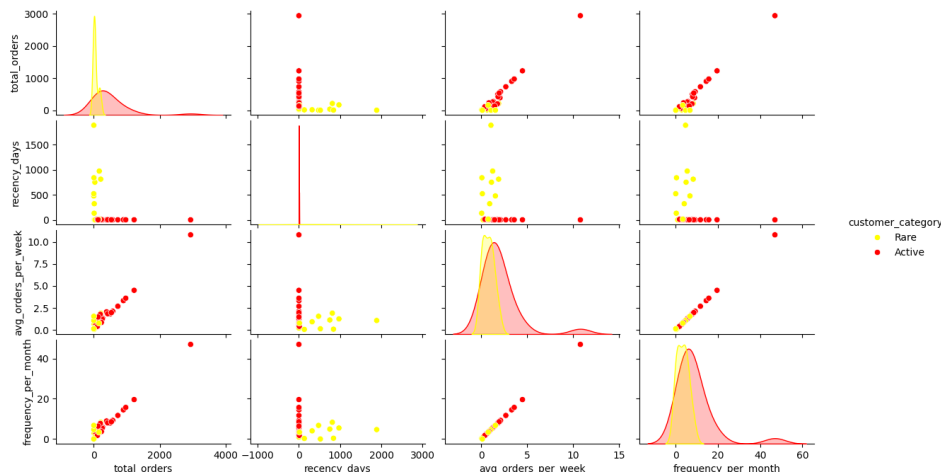


Figure 4.32: Pairplot of ordering behavior variables by customer category.

4.2.13 Zero-Order Day Analysis by Segment

A key differentiator between the two segments is the proportion of days with no orders:

- **Active Customers:** 83.96% of days are zero-order days.
- **Rare Customers:** 97.98% of days are zero-order days.

Despite both segments showing sparsity, Active customers exhibit significantly more consistent ordering behavior, contributing to the overall demand signal. In

contrast, Rare customers generate demand in an infrequent and volatile manner, which justifies their separation for modeling purposes.

Time Series Decomposition

To better understand the temporal structure of active customer demand, we performed time series decomposition analysis. Figure 4.33 presents the decomposition of the daily order time series for active customers from 2020 to 2025. The decomposition separates the original time series into three fundamental components: trend, seasonality, and residuals. This analysis helps to better understand the underlying structure of customer ordering behavior.

- **Original Data (Top Panel):** The raw daily order data shows a clear upward trajectory over time. Order volumes were relatively low during 2020–2021 but steadily increased, with frequent spikes reaching over 200 daily orders by 2024–2025. The pattern becomes increasingly volatile in recent years, indicating growing variability in customer activity.
- **Trend Component (Second Panel):** The trend component highlights a consistent long-term increase in order volume. The average daily orders rise from around 20–30 in early 2020 to over 80–100 by 2025. This reflects a strong growth trajectory, especially accelerating from 2022 onward.
- **Seasonal Component (Third Panel):** The seasonal component reveals regular, repeating fluctuations tied to weekly or monthly ordering cycles. The amplitude of seasonality remains relatively stable throughout most of the period (2020–2023), with some evidence of slightly increasing seasonal variation in recent years (2024–2025), indicating largely predictable customer ordering behavior with potentially strengthening seasonal effects.
- **Residual Component (Bottom Panel):** The residuals capture the random variation in the data after accounting for trend and seasonality. These become more volatile from 2023 onward, suggesting the influence of additional unpredictable factors or changes in customer behavior not captured by the model.

Overall, the decomposition indicates strong and sustained growth, along with stable seasonal patterns. However, the increasing variability in the residuals in recent periods may reflect growing uncertainty or external influences.

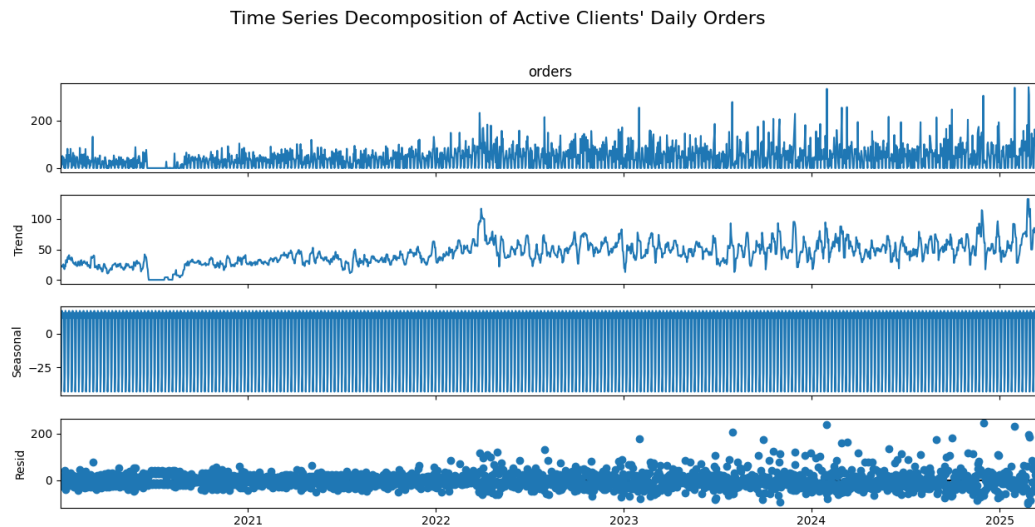


Figure 4.33: Time series decomposition of daily orders from active customers (2020–2025).

4.2.14 Stationarity Analysis

To assess the stationarity of the daily order time series for active customers, two complementary statistical tests were applied: the Augmented Dickey–Fuller (ADF) test and the KPSS (Kwiatkowski–Phillips–Schmidt–Shin) test. These tests were performed both on the original series and after applying a first-order differencing transformation.

Table 4.13: Summary of Stationarity Test Results (Before and After Differencing)

Test	Series	Test Statistic	p-value	Decision	Conclusion
ADF	Original	−4.0387	0.0012	Reject H_0	Stationary
KPSS	Original	8.3115	≤ 0.01	Reject H_0	Non-stationary
ADF	Differenced	−14.3236	0.0000	Reject H_0	Stationary
KPSS	Differenced	0.0163	≥ 0.10	Fail to reject H_0	Stationary

Interpretation: The results from the original series are conflicting: the ADF test suggests stationarity, while the KPSS test indicates non-stationarity. This discrepancy is often symptomatic of a **trend-stationary** process, where the time series becomes stationary after removing a deterministic trend. After applying first-order differencing, both tests converge in their assessment, confirming that the differenced series is stationary. Thus, differencing may lose important real-world information, so models that naturally handle non-stationary data are often preferred in practice.

4.3 Demand Pattern Classification Analysis

The demand pattern classification was conducted using the Average Demand Interval (ADI) and Squared Coefficient of Variation (CV^2) methodology across 24 selected customers over a 1,904-day observation period (2020-2025).

Cities were classified using established thresholds ($ADI = 1.32$, $CV^2 = 0.49$) into four categories: Smooth (frequent, low variability), Erratic (frequent, high variability), Intermittent (infrequent, low variability), and Lumpy (infrequent, high variability).

4.3.1 Results

Table 4.14: Demand Pattern Classification Results by City

City	ADI	CV^2	Pattern	Non-Zero Days
Alger	1.51	0.69	Lumpy	1,259
Batna	4.19	0.38	Intermittent	454
Biskra	23.22	0.41	Intermittent	82
Blida	3.28	1.06	Lumpy	580
Bordj-Bou-Arreridj	9.92	0.36	Intermittent	192
Bouira	7.77	0.28	Intermittent	245
Chlef	5.99	0.40	Intermittent	318
Constantine	4.98	0.53	Lumpy	382
El-Oued	20.92	0.60	Lumpy	91
Guelma	8.50	0.34	Intermittent	224
Jijel	18.67	0.11	Intermittent	102
M'Sila	20.04	0.81	Lumpy	95
Mascara	24.41	0.29	Intermittent	78
Medea	36.62	0.48	Intermittent	52
Mostaganem	17.15	0.48	Intermittent	111
Oran	4.20	0.49	Lumpy	453
Ouargla	4.29	0.61	Lumpy	444
Setif	4.34	0.87	Lumpy	439
Sidi-Bel-Abbes	4.02	0.45	Intermittent	474
Skikda	5.30	0.59	Lumpy	359
Tebessa	8.98	0.27	Intermittent	212
Tiaret	18.67	0.25	Intermittent	102
Tizi-Ouzou	4.03	0.35	Intermittent	472
Tlemcen	17.15	0.32	Intermittent	111

4.3.2 Key Findings

The analysis reveals a binary classification outcome with no cities exhibiting Smooth or Erratic patterns:

Intermittent Pattern Cities (n=15): Characterized by infrequent but relatively predictable demand (mean $ADI=13.95$, mean $CV^2=0.35$). These cities demonstrate sporadic ordering patterns with low variability when orders occur, suggesting planned procurement behaviors.

Lumpy Pattern Cities (n=9): Display infrequent and highly variable demand (mean $ADI=7.65$, mean $CV^2=0.70$). Despite lower average ADI values, these cities exhibit significant demand volatility, accounting for 66.1% of total order volume (54,669 orders).

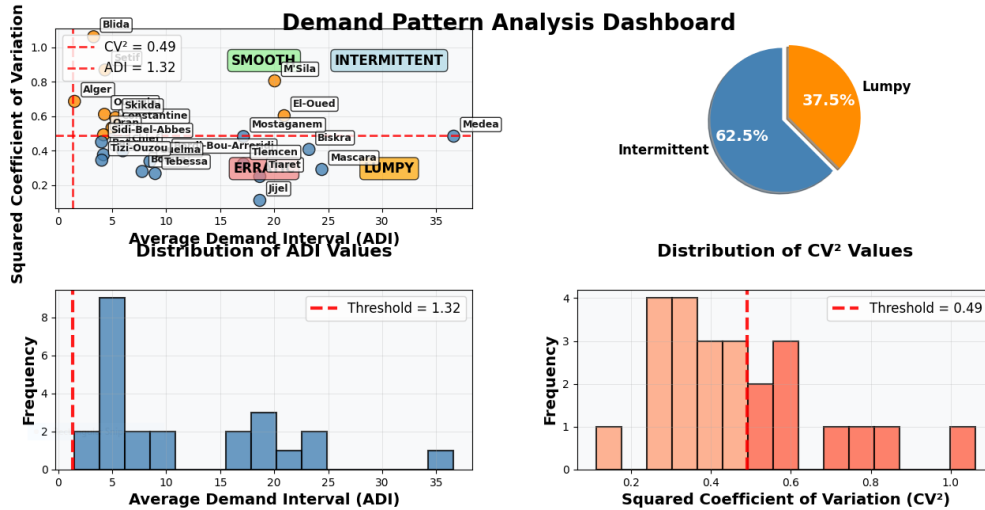


Figure 4.34: Demand Pattern Classification Analysis

Table 4.15: Pattern Distribution Summary

Pattern	Count	Percentage	Characteristics
Intermittent	15	62.5%	Infrequent, predictable
Lumpy	9	37.5%	Infrequent, highly variable
Smooth	0	0.0%	Frequent, predictable
Erratic	0	0.0%	Frequent, highly variable

4.3.3 Implications

The absence of frequent demand patterns (Smooth/Erratic) across all cities indicates predominantly intermittent ordering behaviors. Lumpy cities generate 66.1% of total orders while comprising only 37.5% of all cities.

4.4 Forecasting Models and Results

This section presents the implementation and evaluation of forecasting models used to predict daily order volumes at the city level. All models were implemented in **Python**, which provides a wide range of libraries for time series analysis and machine learning. To ensure a comprehensive and balanced benchmarking process, both classical time series models and modern machine learning techniques were considered.

4.5 Benchmark Model: Croston's Method

As a specialized approach for handling intermittent demand, Croston's method was adopted as a baseline model.

4.5.1 Implementation Steps

The implementation follows the Croston method detailed in Algorithm 1, adapted for our 24 customer dataset with 30-day forecast horizons.

The logic is encapsulated in a custom `CrostonMethod` class, which processes each city’s daily time series and returns a vector of predicted demand. For cities with no non-zero demand, the model outputs zero forecasts.

Forecast Evaluation Window: Forecasts are assessed on the last 30 days of the dataset to simulate operational usage and measure predictive accuracy. The cutoff point for the train/test split is fixed at **February 17, 2025**.

4.5.2 Croston Method Results and Analysis

Performance Summary

Table 4.16 presents the performance metrics for Croston’s method across all 24 cities, ranked by RMSE performance. The analysis reveals significant variation in forecasting accuracy across different customer segments.

Table 4.16: Croston Method Performance Metrics by City (Ranked by RMSE)

City	RMSE	MAE
<i>Best Performing Cities</i>		
Medea	1.22	0.33
Mascara	1.59	0.74
Tebessa	1.77	0.88
El-Oued	1.88	0.62
Chlef	1.99	1.39
Tiaret	2.45	1.12
Bordj-Bou-Arredj	2.78	1.45
Mostaganem	3.15	1.89
Jijel	3.42	2.01
Guelma	3.67	2.15
<i>Worst Performing Cities</i>		
Blida	11.09	6.98
Ouargla	13.38	7.35
Oran	16.70	12.04
Alger	17.31	14.77
Tizi-Ouzou	39.17	33.98

Key Performance Insights

The results reveal several critical patterns in Croston’s method performance:

Overall Performance The cities exhibit a clear three-tier performance structure:

- **High performers** (RMSE < 4.0): Primarily smaller cities with sparse, predictable demand patterns

- **Medium performers** (RMSE 4.0-7.0): Mid-sized cities with moderate demand variability
- **Poor performers** (RMSE > 10.0): Major cities with high-volume, volatile demand

Demand Volume vs. Accuracy The negative correlation between city size and forecasting accuracy is evident. High-volume cities (Alger, Oran, Ouargla) show the worst performance, with RMSE values exceeding 13.0, while sparse-demand cities (Medea, Mascara) achieve RMSE below 2.0.

Model Limitations $R^2 = -0.095$ reveals a fundamental limitation: The negative, Croston's method fails to explain any meaningful variation in the demand data. This occurs because the model cannot capture the patterns and demand volatility present in the dataset. The exponential smoothing of demand sizes and intervals produces overly flat forecasts that systematically underestimate the actual variance in customer ordering behavior.

Forecast Accuracy Analysis

Figure 4.35 illustrates Croston's performance across three representative cities with different demand characteristics.

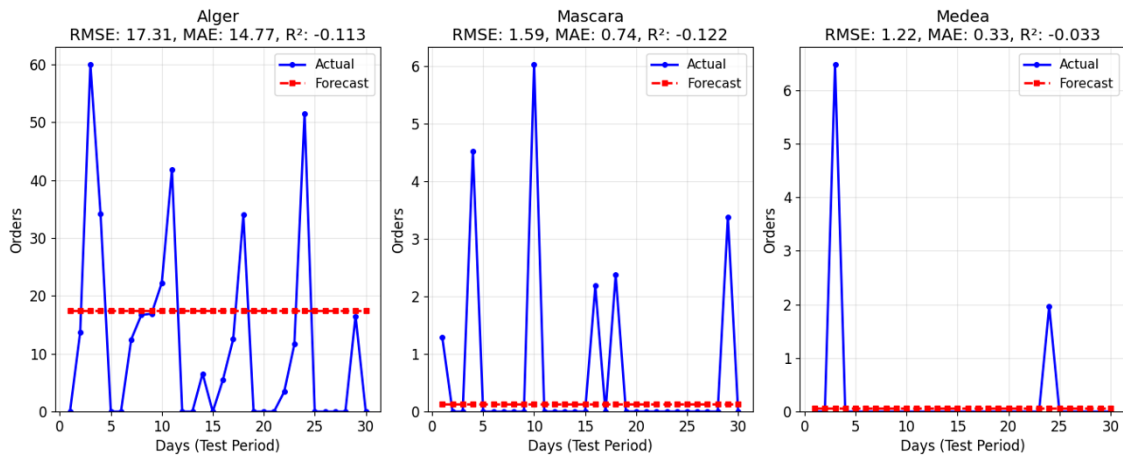


Figure 4.35: Croston method forecasts vs. actual demand for representative cities with varying demand patterns

The visual analysis confirms the quantitative findings:

- **Alger:** High forecast errors due to inability to capture demand spikes
- **Mascara:** Moderate performance with consistent under-forecasting
- **Mascara:** Best relative performance due to low demand occurrence

Method Assessment

Croston’s method demonstrates limited effectiveness for this dataset, with an overall MAE of 4.35 and mean RMSE of 6.40. The method’s core assumptions of stable demand sizes and intervals are violated by the highly irregular, spike-prone demand patterns observed across most cities. The conservative smoothing approach consistently under-estimates demand volatility, particularly problematic for the high-volume customers that drive business performance.

These limitations motivate the need for exploring other forecasting approaches capable of handling complex intermittent demand patterns.

4.6 TSB Method (Teunter–Syntetos–Babai)

The TSB method was implemented as a more refined solution for intermittent demand forecasting. Unlike Croston, the TSB method directly models the probability of demand occurrence alongside demand size.

4.6.1 Implementaion steps:

The implementation follows the TSB method detailed in Algorithm 15, adapted for our 24 customer dataset with 30-day forecast horizons.

Parameter Tuning: To enhance the TSB model’s responsiveness to varying demand behaviors, we conducted a grid search over a range of smoothing parameters α (for demand size) and β (for demand probability). Specifically, we tested the following combinations:

$$\alpha, \beta \in \{0.05, 0.10, 0.15, 0.20\}$$

For each (α, β) pair, the TSB model was re-run across all cities with a fixed forecast horizon of 30 days. Performance was evaluated using the Mean Absolute Scaled Error (MASE) and the absolute value of the forecast Bias. To select the best configuration, we defined a composite score:

$$\text{Combined Score} = \text{MASE} + \frac{|\text{Bias}|}{10}$$

This formulation prioritizes forecast accuracy while penalizing systematic under- or overestimation. After evaluating all parameter combinations, the optimal setting was identified as:

$$\boxed{\alpha = 0.15, \quad \beta = 0.05}$$

This configuration yielded the lowest combined error across all cities, striking a balance between stability and adaptability in demand modeling.

4.6.2 TSB Method Results and Analysis

Performance Summary

Table 4.17 presents the performance metrics for the TSB method across all 24 cities, using optimal parameters $\alpha = 0.15$ and $\beta = 0.05$. The TSB method shows marginal improvements over Croston's approach while maintaining similar limitations in handling intermittent demand patterns.

Table 4.17: TSB Method Performance Metrics by City ($\alpha=0.15$, $\beta=0.05$)

City	RMSE	MAE
<i>Best Performing Cities</i>		
Medea	1.20	0.48
Mascara	1.55	0.82
Tebessa	1.75	0.94
El-Oued	1.90	0.56
Chlef	2.01	1.38
Tiaret	2.15	1.02
Bordj-Bou-Arredj	2.34	1.25
Mostaganem	2.67	1.55
Jijel	2.89	1.78
Guelma	3.12	1.89
<i>Worst Performing Cities</i>		
Blida	11.10	6.98
Ouargla	13.34	7.38
Oran	16.29	11.85
Alger	18.01	15.92
Tizi-Ouzou	42.32	21.03

Method Comparison and Performance Analysis

Table 4.18 presents a direct comparison between Croston's method and the TSB approach, highlighting the marginal improvements achieved through separate modeling of demand probability.

Table 4.18: Performance Comparison: Croston vs. TSB Methods

Metric	Croston	TSB	Improvement (%)
RMSE	6.397	6.526	-2.0
MAE	4.350	3.879	+10.8
R ²	-0.095	-0.094	+1.4

Key Performance Insights

The TSB method demonstrates mixed results compared to Croston's approach:

Modest Improvements in Accuracy The TSB method achieves a notable 10.8% improvement in MAE, reducing the average absolute error from 4.35 to 3.88 pallets.

Persistent Fundamental Limitations Despite theoretical advantages, TSB exhibits similar structural problems as Croston’s method:

- **Negative R^2 values** (mean = -0.094) indicate continued inability to explain demand variation
- **RMSE deterioration** of 2.0% suggests increased sensitivity to extreme values
- **Overall Performance** remains unchanged, with the same cities dominating best and worst performer categories

Component Analysis Insights Figure 4.36 illustrates TSB’s forecast behavior across representative cities, revealing the method’s conservative forecasting tendency through flat, constant predictions that fail to capture demand volatility.

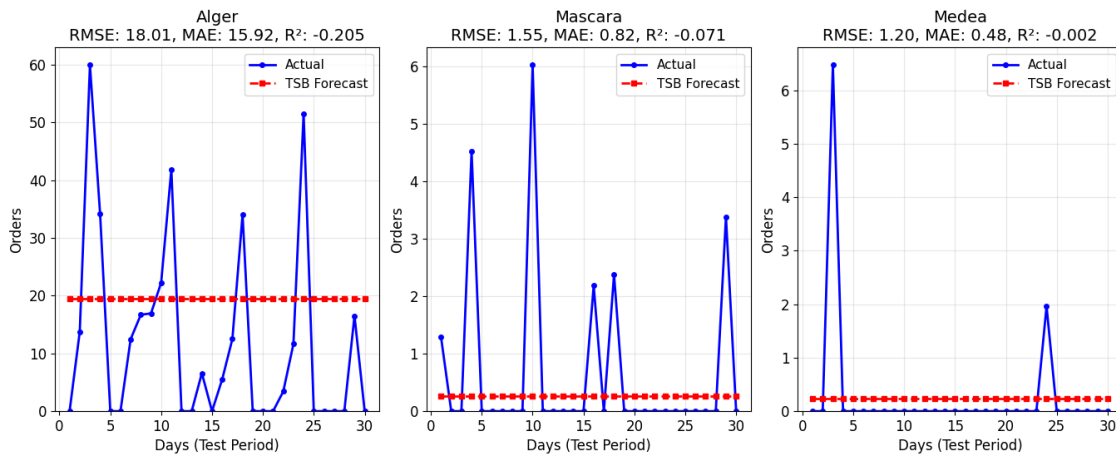


Figure 4.36: TSB method forecasts vs. actual demand

The visual analysis confirms quantitative findings:

- **Alger:** Severe under-forecasting with constant 18.0 pallet prediction failing to capture 60+ pallet spikes
- **Mascara:** Reasonable performance due to low demand variability, though still missing demand timing
- **Medea:** minimal forecast error due to consistently low demand levels but still lacks the ability to predict demand occurrence

The results shown highlight the limitations of traditional intermittent demand forecasting methods for this dataset. While the separate modeling of demand size and probability provides theoretical advantages, the fundamental assumption of stable parameters remains violated by the highly irregular demand patterns observed.

4.7 Towards Advanced Forecasting: Ensemble Learning models

While traditional intermittent demand forecasting methods like Croston and TSB provide interpretable baselines, they rely on simplified assumptions and often fall short in capturing complex demand dynamics across diverse customer segments. To address this, we transition to a more data-driven, adaptive framework that incorporates ensemble learning techniques.

4.7.1 Statistical Predictability Assessment of Demand Components

To determine the optimal forecasting strategy, we decomposed the intermittent demand problem into its constituent components and conducted formal statistical tests to assess their predictability characteristics.

For each city c , we extracted temporal gaps between consecutive positive demand events and analyzed positive demand values separately. We applied three key statistical measures: Coefficient of Variation (CV), lag-1 autocorrelation, and the Ljung-Box test for temporal dependence.

Statistical Results

The analysis encompassed 11,780 observations across 24 cities. Table 4.19 summarizes the comparative results.

Table 4.19: Predictability Assessment Results by Component

Statistical Measure	Gaps	Positive Demand
Mean Coefficient of Variation	1.36	0.65
Cities with favorable CV	62.5%	100.0%
Mean Lag-1 Autocorrelation	0.04	0.17
Cities with meaningful patterns	37.5%	70.8%
Ljung-Box test (temporal structure)	45.8%	66.7%
Mean Ljung-Box p-value	0.365	0.157

Methodological Decision

The statistical evidence reveals contrasting predictability characteristics between demand components. Gap sequences exhibit high variability (mean CV = 1.36) with negligible temporal structure (autocorrelation = 0.04) and no significant patterns (p -value = 0.365). In contrast, positive demand shows reasonable predictability with lower variability (CV = 0.65), meaningful temporal patterns in 70.8% of cities, and moderate autocorrelation (0.17).

These results indicate that gaps are largely unpredictable random events, while positive demand contains sufficient structure for forecasting. Consequently, we focus exclusively on positive demand forecasting, transforming the sparse intermittent demand problem into a dense regression task suitable for gradient boosting methods.

4.7.2 Feature Engineering for Positive Demand Prediction

We engineered features specifically for positive demand forecasting. All lag-based features were properly shifted to prevent data leakage.

Table 4.20: Engineered Features for Demand Forecasting

Category	Features	Purpose
Lag Features	demand_lag1, demand_lag2	Capture autoregressive patterns
Rolling Statistics	7-day mean, 14-day max	Smooth volatility and identify peaks
Temporal Features	Month, day_of_week, quarter	Capture seasonal patterns
Customer Features	City encoding	Account for customer heterogeneity

4.7.3 XGBoost Implementation:

We employed an XGBoost regression model 13 trained on positive demand observations from all 24 cities. The model predicts city-level demand $\hat{y}_{c,t}$ using lag features $\{y_{c,t-1}, y_{c,t-2}\}$, rolling statistics, and temporal variables. This single-model architecture learns cross-city patterns through shared parameters while maintaining city-specific predictions. The implementation followed an **object-oriented approach** with standard preprocessing including **feature engineering** for temporal patterns and **label encoding** of the categorical variable (**city**) to enable the algorithm to capture city-level demand variability.

Data Splitting Strategy

Conventional random splits were avoided to preserve temporal dependencies and prevent information leakage. A **chronological split** was applied, allocating observations from January 8, 2020 to May 16, 2024 for training (80% of data) and May 19, 2024 to March 19, 2025 for testing (20% of data). This split ensures that all training data precedes test data temporally.

Model Training and Hyperparameters

The XGBoost model was trained using comprehensive hyperparameter optimization to balance model complexity, accuracy, and overfitting control. Table 4.21 presents

the parameter search space, followed by detailed explanations of each parameter's role.

Table 4.21: XGBoost Hyperparameter Search Space

Category	Parameter	Search Values
Tree Structure	max_depth	{3, 5, 7, 9}
	min_child_weight	{1, 3, 5}
	gamma	{0, 0.1, 0.2}
Learning	learning_rate	{0.01, 0.05, 0.1, 0.2}
Sampling	subsample	{0.6, 0.8, 1.0}
	colsample_bytree	{0.6, 0.8, 1.0}
	colsample_bylevel	{0.6, 0.8, 1.0}
	colsample_bynode	{0.6, 0.8, 1.0}
Regularization	reg_alpha (L1)	{0, 0.01, 0.1}
	reg_lambda (L2)	{1, 2, 3}

Tree Structure Parameters:

- **Number of Trees (`n_estimators`):** Fixed to 1000. It controls the ensemble size. Higher values generally improve performance but risk overfitting. Early stopping was employed to terminate training when validation error stopped improving after 50 rounds.
- **Learning Rate:** Set between 0.01 and 0.2 to control the step size at each iteration. Smaller values make the model learn more carefully from each tree, helping avoid overfitting and improving generalization.
- **Maximum Depth (`max_depth`):** Controls the maximum depth of trees, restricted to values between 3 and 9 to prevent trees from becoming too deep, which would cause excessive partitioning of the data and high variance.
- **Minimum Child Weight (`min_child_weight`):** Defines the minimum sum of instance weight needed in a child. Values {1, 3, 5} were tested to control when tree splitting stops, preventing overfitting to noise.
- **Gamma:** Specifies the minimum loss reduction required for further partitioning. Tested values {0, 0.1, 0.2} provide additional complexity control.

Sampling Parameters: Multiple sampling parameters introduce stochasticity to reduce tree correlation and improve generalization:

- **Subsample:** Controls row sampling ratio, meaning each tree trains on 60% to 100% of available instances. This reduces overfitting by introducing randomness in training data selection.

- **Feature Sampling:** Three variants control feature selection at different stages:
 - * `colsample_bytree`: Fraction of features per tree (0.6–1.0)
 - * `colsample_bylevel`: Fraction of features per tree level (0.6–1.0)
 - * `colsample_bynode`: Fraction of features per node split (0.6–1.0)

Regularization Terms: L1 penalty (`reg_alpha`) tested in $\{0, 0.01, 0.1\}$ pushes small weights to zero for feature selection, while L2 penalty (`reg_lambda`) in $\{1, 2, 3\}$ smooths predictions by penalizing large weights.

Random Search: In our implementation, we conducted 25 random trials using Root Mean Square Error (RMSE) as the evaluation metric, allowing random search to efficiently explore the important parameter subspaces while avoiding wasted evaluations on irrelevant parameter combinations.

Optimal Configuration: The best hyperparameters found were:

- `subsample`: 1.0
- `reg_lambda`: 3
- `reg_alpha`: 0.01
- `min_child_weight`: 3
- `max_depth`: 5
- `learning_rate`: 0.01
- `gamma`: 0.2
- `colsample_bytree`: 0.8
- `colsample_bynode`: 1.0
- `colsample_bylevel`: 0.8
- Optimal `n_estimators` (via early stopping): 708

Model Evaluation and Performance Assessment

To assess the effectiveness of our demand forecasting model, we employed standard regression performance metrics. The model achieved the following results on the test dataset:

- **$R^2 = 0.8739$:** The model explains 87.39% of the variance in positive demand values, indicating strong predictive capability.
- **RMSE = 2.89:** Root Mean Square Error of 2.89 units demonstrates reasonable prediction accuracy relative to the demand range.
- **MAE = 1.66:** Mean Absolute Error of 1.66 units shows typical prediction errors are within acceptable bounds for business planning.

- **MAPE = 38%**: The Mean Absolute Percentage Error demonstrates that forecast deviations average 38% relative to observed demand values, indicating moderate prediction accuracy.

Visual validation through actual versus predicted plots for representative cities (Alger, Tlemcen, Mascara) confirms the model's ability to capture both trend directions and magnitude patterns, though with expected slight underestimation of extreme demand peaks.

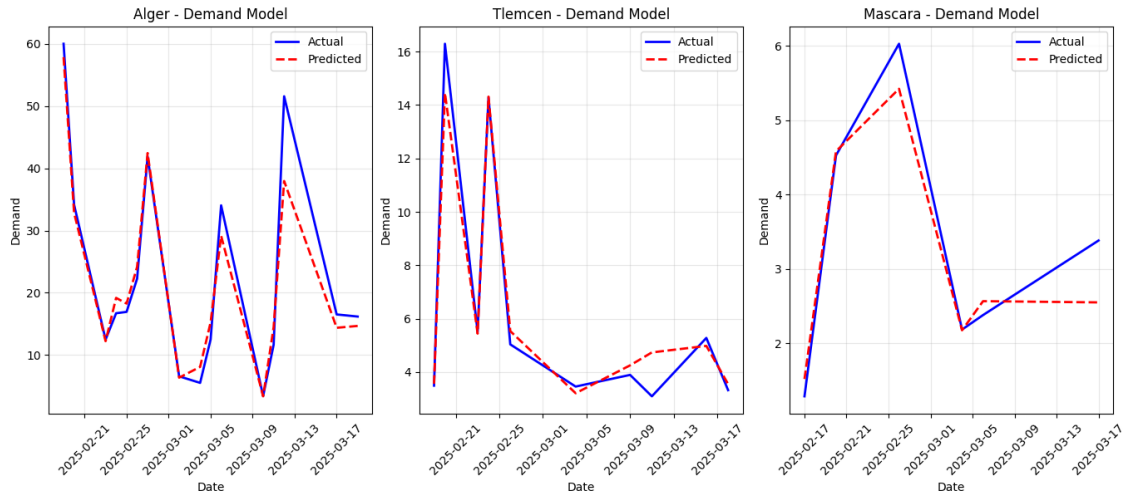


Figure 4.37: XGBoost Demand Prediction Results for Three Representative Cities

Table 4.22: XGBoost Method Performance Metrics by City

City	RMSE	MAE
<i>Best Performing Cities</i>		
Medea	0.408	0.339
Tebessa	0.436	0.325
Jijel	0.439	0.310
Guelma	0.454	0.327
Sidi-Bel-Abbes	0.455	0.335
Tiaret	0.455	0.369
Tlemcen	0.523	0.361
Mascara	0.525	0.357
Chlef	0.550	0.371
M'Sila	0.566	0.438
<i>Worst Performing Cities</i>		
Setif	1.651	0.992
Alger	2.609	1.852
Oran	2.814	1.670
Ouargla	2.944	1.498
Tizi-Ouzou	4.259	3.048

XGBoost Performance

The XGBoost model demonstrates **substantial improvements** over traditional intermittent demand forecasting methods, achieving good accuracy across all performance metrics and city categories. Table 4.23 compares the three forecasting approaches. Note that while Croston and TSB predict over the complete intermittent time series (including zero-demand periods), XGBoost focuses specifically on positive demand forecasting, representing a methodologically distinct but practically valuable problem formulation.

Table 4.23: Method Performance Comparison: XGBoost vs. Traditional Approaches

Method	Mean RMSE	Mean MAE	Best City RMSE
Croston	6.397	4.350	1.22
TSB	6.526	3.879	1.20
XGBoost	2.89	1.66	0.408
<i>Improvement vs. Best Traditional Method</i>			
RMSE Improvement	56% better than TSB		
MAE Improvement	57.2% better than TSB		

Conclusion

This chapter established a comprehensive analytical framework for understanding customer demand patterns in dairy distribution, identifying critical characteristics that inform both forecasting methodology and strategic logistics planning. Through systematic analysis of highly intermittent demand data, we developed robust forecasting capabilities using ensemble learning techniques that significantly outperformed traditional approaches.

The next Chapter will implement the VRP model to develop optimal routing solutions and evaluate their operational impact.

Chapter 5

Implementation of the Mathematical VRP Model and Solution Approaches

Introduction

This chapter presents the implementation of an optimization model designed to improve the distribution operations of **Celia**. Due to the perishable nature of the goods and the wide geographic distribution of clients, effective vehicle routing is essential to reduce transportation costs and ensure product freshness.

After analyzing Celia's distribution constraints, a mathematical model was developed to optimize delivery planning. This chapter explains how the model was built, how real-world data was prepared, and how the solution was applied to solve the problem.

The implementation includes both a formal mathematical formulation and practical resolution using optimization solvers. Finally, we compare the solutions in terms of cost, efficiency, and feasibility for operational use.

5.1 Problem Statement

The problem addressed in this work is a complex variant of the classical Vehicle Routing Problem (VRP), tailored to the real-world operations of **Celia**.

The specific characteristics of the problem include:

- **Open VRP:** Delivery routes end at customer locations; vehicles do not return to the depot.
- **Heterogeneous Fleet:** The available fleet consists of trucks with different capacities and costs.
- **Split Deliveries:** A single customer's demand may be served by multiple vehicles if necessary.
- **Multi-Trips:** Each truck may be assigned to more than one route within the same planning horizon, with a maximum of two routes as defined by company policy.
- **Time Windows:** Deliveries must occur within fixed operating hours.
- **Single Depot:** All deliveries originate from the central depot.
- **Company-Specific Constraints:**
 - Grouping of destinations to improve loading efficiency (No more than 3 customers).
 - Incompatibility of far destinations within the same truck (e.g., no mixing of east-west routes).
 - Assignment preferences between truck types and clients.
 - Overnight stays at customer locations when arrival times exceed allowable time windows.

The objective is to design optimal delivery routes that minimize total transportation costs while satisfying customer demands, vehicle and time constraints, and maintaining product quality through efficient routing and vehicle use.

5.2 Mathematical Formulation

The following section presents the mathematical formulation of the problem, modeled as a **Heterogeneous Fleet Open Vehicle Routing Problem with Split Deliveries, Time Windows, and Multi-Trips** (HF-OVRP-SD-TW-MT).

5.2.1 Sets and Parameters

Sets and subsets:

- N : Set of all nodes (customers and depot), where node 0 represents the depot.
- K : Set of available trucks.
- R : Set of possible trips (or rounds) a vehicle can perform.
- $N_{\text{south}} \subset N$: Subset of customers located in the southern region, subject to specific routing constraints.
- $N_{\text{no_heavy}} \subset N$: Subset of customers that cannot be served by heavy trucks.
- $K_{20t} \subset K$: Subset of trucks with 20-ton capacity.
- $K_{\text{heavy}} \subseteq K$: Subset of heavy trucks, including 20-ton and other large-capacity types.

Parameters:

- Q_k : Capacity of vehicle $k \in K$ (in number of pallets).
- d_i : Demand at node $i \in N$ (in number of pallets).
- c_{ik} : Fixed cost of serving node i with vehicle k .
- dist_{ij} : Distance between nodes i and j , for all $i, j \in N$.
- t_{ij} : Travel time between nodes i and j , for all $i, j \in N$.
- p_k : Maximum unloading time allowed for truck $k \in K$, to preserve product freshness.
- β_k : Unloading time coefficient for truck $k \in K$ (in time per pallet).
- e_i : Earliest service time at customer i (start of time window).
- l_i : Latest preferred service time at customer i (end of time window).

- $h_{\text{start},k}$: Earliest start time for truck k .
- R_{max} : Maximum allowed distance between customers.
- $\text{MAX_RETURN_TIME_FIRST_TOUR}$: Latest allowed return time to the depot for first tours (to enable a second trip).
- NEXT_DAY_START : Time at which deliveries may resume on the following day, used to model overnight stays and next-day departures.
- M : Big-M constant used in logical constraints to relax conditions.
- numNodes : Total number of nodes in the network, $|N|$.

5.2.2 Decision Variables

Binary Variables:

- x_{ijk_r} : 1 if vehicle k travels from node i to node j during trip r , 0 otherwise
- y_{ik_r} : 1 if node i is visited by vehicle k in trip r , 0 otherwise
- δ_{k_r} : 1 if vehicle k performs trip r , 0 otherwise
- last_{ik_r} : 1 if node i is the last node visited by vehicle k in trip r , 0 otherwise
- split_i : 1 if the demand of customer i is split across multiple deliveries, 0 otherwise.
- nightStay_{ik_r} : 1 if vehicle k stays overnight at node i in trip r , 0 otherwise.
- $\text{return_to_depot}_{k_r}$: 1 if vehicle k returns to depot after trip r , 0 otherwise

Continuous Variables:

- q_{ik_r} : Quantity delivered to customer i by vehicle k in trip r
- Z_{k_r} : Total route cost for vehicle k in trip r
- u_{ik_r} : Auxiliary variable for sub-tour elimination (node position)
- $t_{\text{arr}_{ik_r}}$: Arrival time at node i for vehicle k in trip r
- $t_{\text{dep}_{ik_r}}$: Departure time from node i for vehicle k in trip r
- s_{ik_r} : Service time at node i for vehicle k in trip r
- $t_{\text{return_depot}_{k_r}}$: Time vehicle k returns to depot in trip r

5.2.3 Objective Function and Cost Modeling

In this model, the objective is to minimize the total delivery cost across all trips and all vehicles, considering Celia's real cost logic per tour. The cost of each tour is composed of:

- A fixed cost of **3500 DA per customer** served by a vehicle on a trip, except for the last customer;
- A variable cost c_{ik} , which depends on the destination and vehicle type, is assigned to the **last customer** in the tour.

The cost for each vehicle-trip pair (k, r) is calculated as:

$$\text{cost}_{kr} = 3500 \times \left(\sum_{i \in N \setminus \{0\}} y_{ikr} - \delta_{kr} \right) + \sum_{i \in N \setminus \{0\}} \text{last}_{ikr} \cdot c_{ik} \quad (5.1)$$

5.2.4 Objective Function: Min–Max and Linearization

To ensure balanced routing and fair cost allocation per Celia's policy, we adopt a **min–max objective** that minimizes the sum of the maximum segment costs for each truck and each trip.

The cost logic is based on the principle of having the most distant customer in each tour bear the actual delivery cost. In contrast, earlier customers pay a fixed charge of 3,500 DA. This encourages routes that follow a logical geographical sequence (from nearest to farthest) and avoids illogical assignments.

Since the objective involves a min-max structure, we linearize the inner max operator using auxiliary continuous variables Z_{kr} such that:

$$\text{cost}_{kr} \leq Z_{kr} \quad \forall k \in K, \forall r \in R \quad (5.2)$$

The overall objective becomes:

$$\min \sum_{k \in K} \sum_{r \in R} Z_{kr} \quad (5.3)$$

5.2.5 Constraints

Capacity Constraints

$$\sum_{i \in N \setminus \{0\}} q_{ikr} \leq Q_k \quad \forall k \in K, r \in R \quad (5.4)$$

Vehicle capacity limits: the total loaded quantity cannot exceed the truck's capacity.

$$q_{ikr} \leq d_i \cdot y_{ikr} \quad \forall i \in N \setminus \{0\}, k \in K, r \in R \quad (5.5)$$

This constraint links delivery quantities to vehicle–route assignments, ensuring that a customer receives a positive delivery only if they are served, and that the delivered amount does not exceed their demand.

Demand Satisfaction Constraints

$$\sum_{k \in K} \sum_{r \in R} q_{ikr} = d_i \quad \forall i \in N \setminus \{0\} \quad (5.6)$$

This fundamental service constraint guaranties that each customer's demand d_i is exactly satisfied by the cumulative deliveries from all vehicles and routes, ensuring complete fulfillment.

$$\sum_{k \in K} \sum_{r \in R} y_{ikr} \leq 1 + split_i \quad \forall i \in N \setminus \{0\} \quad (5.7)$$

This constraint governs how deliveries can be divided among customers: when $split_i = 0$, customer i receives their entire order in a single visit; when $split_i = 1$, customer i 's order can be divided across up to two separate visits, allowing the system to handle large orders that exceed vehicle capacity.

Route Constraints

$$\sum_{i \in N \setminus \{0\}} y_{ikr} \leq 3 \quad \forall k \in K, r \in R \quad (5.8)$$

This constraint limits each route to serving at most 3 customers, balancing route duration and product freshness requirements.

$$\sum_{r \in R} \delta_{kr} \leq 2 \quad \forall k \in K \quad (5.9)$$

This constraint allows each vehicle to perform more than one route per day to satisfy more costumers.

$$\delta_{kr} \leq \sum_{i \in N \setminus \{0\}} y_{ikr} \quad \forall k \in K, r \in R \quad (5.10)$$

This logical constraint ensures that a route is only considered active if at least one customer is visited, excluding the depot, thus preventing empty or meaningless route activations.

$$\delta_{kr} \geq y_{ikr} \quad \forall i \in N, k \in K, r \in R \quad (5.11)$$

The constraint forces route activation whenever any customer is visited.

Last Customer Constraints

$$\sum_{i \in N \setminus \{0\}} last_{ikr} = \delta_{kr} \quad \forall k \in K, r \in R \quad (5.12)$$

This ensures that exactly one customer must be designated as the last stop for each active route.

$$last_{ikr} \leq y_{ikr} \quad \forall i \in N \setminus \{0\}, k \in K, r \in R \quad (5.13)$$

This logical constraint ensures that only visited customers can be designated as the last stop.

$$last_{ikr} \geq y_{ikr} - \sum_{j \in N \setminus \{i,0\}} x_{ijk} \quad \forall i \in N \setminus \{0\}, k \in K, r \in R \quad (5.14)$$

This constraint identifies the last customer: if customer i is visited but has no outgoing arc to another customer, it must be the last stop on that route.

Flow Conservation Constraints

1. Inflow Constraint

$$\sum_{j \in N: j \neq i} x_{jikr} = y_{ikr} \quad \forall k \in K, \forall r \in R, \forall i \in N : i \neq 0 \quad (5.15)$$

Each visited customer has exactly one incoming arc, ensuring proper vehicle arrival routing.

2. Outflow Constraint

$$\sum_{j \in N: j \neq i, j \neq 0} x_{ijk} = y_{ikr} - last_{ikr} \quad \forall k \in K, \forall r \in R, \forall i \in N : i \neq 0 \quad (5.16)$$

Each visited customer has exactly one outgoing arc, except for the last customer in the route.

$$\sum_{j \in N \setminus \{0\}} x_{0jkr} = \delta_{kr} \quad \forall k \in K, r \in R \quad (5.17)$$

This constraint ensures that each active route begins at the depot with exactly one outgoing arc, establishing the depot as the starting point for all routes.

Subtour Elimination (MTZ)

$$u_{0kr} = 0 \quad \forall k \in K, r \in R \quad (5.18)$$

This initialization sets the depot's position to zero in the MTZ sequencing, establishing it as the reference point for the subtour elimination mechanism.

$$u_{ikr} - u_{jkr} + (|N| + 1) \cdot x_{ijk} \leq |N| \quad \forall i, j \in N \setminus \{0\}, i \neq j, k \in K, r \in R \quad (5.19)$$

This Miller-Tucker-Zemlin constraint eliminates illegal subtours by imposing a sequencing order: if vehicle travels from i to j , then $u_j > u_i$, preventing cycles that exclude the depot.

Distance and Sequencing Constraints

$$y_{ikr} + y_{jkr} \leq 1 \quad \forall i, j \in N \setminus \{0\}, i < j, dist_{ij} \geq R_{max}, k \in K, r \in R \quad (5.20)$$

This constraint enforces a maximum service radius by preventing distant customer pairs (separated by more than R_{max}) from being served on the same route.

$$x_{ijk} = 0 \quad \forall k \in K, \forall r \in R, \forall i, j \in N : i \neq 0, j \neq 0, i \neq j, dist_{0i} > dist_{0j} \quad (5.21)$$

Vehicles cannot travel directly from a farther customer to a closer customer (relative to the depot).

Vehicle Type Restrictions

$$\sum_{k \in K \setminus K_{20t}} \sum_{r \in R} y_{ikr} = 0 \quad \forall i \in N_{south} \quad (5.22)$$

This strict constraint enforces that southern customers are exclusively served by 20-ton trucks.

$$y_{ikr} = 0 \quad \forall i \in N_{no_heavy}, k \in K_{heavy}, r \in R \quad (5.23)$$

This constraint prohibits heavy vehicles from serving specific customers with access restrictions.

Time Constraints

$$s_{ikr} = \beta_k \cdot q_{ikr} \quad \forall i \in N \setminus \{0\}, k \in K, r \in R \quad (5.24)$$

This equation calculates service time proportionally to the delivered quantity, where β_k represents the unloading time per pallet for vehicle type k .

$$s_{0kr} = 0 \quad \forall k \in K, r \in R \quad (5.25)$$

This constraint sets zero service time at the depot, as loading operations are typically handled outside the routing time horizon.

$$s_{ikr} \leq p_k \cdot y_{ikr} \quad \forall i \in N \setminus \{0\}, k \in K, r \in R \quad (5.26)$$

This constraint guarantees that the service time at a customer does not exceed the maximum allowable unloading time associated with the assigned vehicle type to maintain product quality.

$$t_{arr,ikr} \geq e_i \cdot y_{ikr} \quad \forall i \in N, k \in K, r \in R \quad (5.27)$$

This constraint enforces the earliest service time at each customer, but only activates when the customer is visited. The multiplication by y_{ikr} avoids imposing unnecessary time restrictions on non-visited nodes.

$$nightStay_{ikr} \geq \frac{t_{arr,ikr} - l_i}{M} \quad \forall i \in N : i \neq 0, \forall k \in K, \forall r \in R \quad (5.28)$$

Detects when a vehicle must stay overnight at customer i by activating the binary variable when arrival time exceeds the customer's latest service time.

$$nightStay_{ikr} \leq y_{ikr} \quad \forall i \in N \setminus \{0\}, k \in K, r \in R \quad (5.29)$$

This logical constraint enforces that overnight stays can only be assigned to customers who are actually visited by a vehicle. It prevents invalid night stay activations at non-visited locations.

$$t_{dep_{0k1}} \geq h_{start_k} \cdot \delta_{k1} \quad \forall k \in K \quad (5.30)$$

$$t_{dep_{0k2}} \geq t_{return_depot_{k1}} + 1.0 \quad \forall k \in K \quad (5.31)$$

The first constraint sets initial departure times from the depot; the second calculates the departure time for the second trip as the return time from the first trip plus the average loading time.

$$t_{dep_{ikr}} \geq t_{arr_{ikr}} + s_{ikr} \quad \forall i \in N : i \neq 0, \forall k \in K, \forall r \in R \quad (5.32)$$

Ensures departure time accounts for service duration, preventing vehicles from leaving before completing service.

$$t_{dep_{ikr}} \geq \text{NEXT_DAY_START} + s_{ikr} - M \cdot (1 - \text{nightStay}_{ikr}) \quad \forall i \in N \setminus \{0\}, k \in K, r \in R. \quad (5.33)$$

For overnight stays, this constraint resets departure to the next morning's opening time plus service duration.

$$t_{arr_{jkr}} \geq t_{dep_{ikr}} + t_{ij} - M \cdot (1 - x_{ijkr}) \quad \forall i, j \in N, i \neq j, k \in K, r \in R \quad (5.34)$$

This fundamental constraint propagates time along the route: if a vehicle travels from i to j , arrival at j must account for departure from i plus travel time, ensuring temporal feasibility.

Depot Return Constraints

$$t_{return_depot_{kr}} \geq t_{dep_{ikr}} + t_{i0} - M \cdot (1 - \text{last}_{ikr}) \quad \forall k \in K, \forall r \in R, \forall i \in N : i \neq 0 \quad (5.35)$$

Ensures the depot return time is at least the departure time from the last customer plus travel time to the depot.

$$t_{return_depot_{kr}} \leq t_{dep_{ikr}} + t_{i0} + M \cdot (1 - \text{last}_{ikr}) \quad \forall k \in K, \forall r \in R, \forall i \in N : i \neq 0 \quad (5.36)$$

Prevents unrealistic return times by setting an upper bound when customer i is the last in the route.

$$t_{return_depot_{k1}} \leq \text{MAX_RETURN_TIME} + M \cdot (1 - \delta_{k2}) \quad \forall k \in K \quad (5.37)$$

This constraint enforces a planning condition: if a second tour is scheduled for vehicle k (i.e., $\delta_{k2} = 1$), the first tour must return to the depot before a fixed deadline (`MAX_RETURN_TIME`). This ensures sufficient time for reloading, making multi-trip operations feasible within operational limits. The constraint is deactivated for single-tour vehicles.

$$\delta_{k2} \leq \delta_{k1} \quad \forall k \in K \quad (5.38)$$

This logical sequencing ensures that a second route can only be assigned if the first route exists.

$$return_to_depot_{kr} = \sum_{i \in N \setminus \{0\}} x_{i0kr} \quad \forall k \in K, r \in R \quad (5.39)$$

This constraint defines the variable $return_to_depot_{kr}$ by counting whether a truck returns to the depot at the end of route r .

$$return_to_depot_{k1} \geq \delta_{k2} \quad \forall k \in K \quad (5.40)$$

This operational constraint ensures that if a vehicle performs a second tour, it must return to the depot after the first.

$$return_to_depot_{k2} = 0 \quad \forall k \in K \quad (5.41)$$

This end-of-day constraint implements the open VRP structure for final routes, where vehicles end at the last customer location rather than returning to the depot.

Having established the complete mathematical framework, we now turn to the practical implementation methodology that translates this theoretical model into an operational solution.

5.3 Solution Methodology Overview

This section presents the implementation of the Vehicle Routing Problem (VRP), detailing the solution approach using IBM CPLEX Optimization Studio and the OPL (Optimization Programming Language) modeling framework. The implementation follows a structured methodology encompassing model translation, solver configuration, scenario testing, and validation through comparative analysis.

5.3.1 CPLEX Implementation Framework

Model Translation to OPL

The mathematical formulation was translated into OPL syntax to leverage CPLEX's optimization capabilities.

The implementation consists of three primary components **Model File (model.mod)**: Contains the complete mathematical model specification including:

- Declaration of decision variables with appropriate data types and bounds.
- Formulation of the objective function.
- Implementation of all constraint sets.
- Set and parameter declarations with domain specifications.

Data File (data.dat): Encompasses all numerical input parameters and problem instances:

- Problem dimensions and set cardinalities
- Resource capacities and demand requirements
- Numerical matrices and time parameters

Execution Framework: The model execution is managed through IBM CPLEX Optimization Studio IDE, enabling:

- Model compilation and syntax checking
- Data file integration and parameter loading
- Solution process execution
- Solution output generation and objective value reporting
- Basic performance statistics (solve time, iterations, nodes explored)

Solver Configuration

IBM CPLEX Optimization Studio was configured with the following specifications:

- **Solver Engine:** IBM CPLEX 22.1.1.0 (Build ID: 202211281837)
- **Solution Method:** Mixed Integer Programming with Branch-and-Cut algorithm
- **Problem Type:** Mixed Integer Linear Programming (MILP)
- **Algorithm Selection:** Automatic (CPLEX default optimization)
- **Optimality Gap Tolerance:** 0.0001 (0.01% - CPLEX default)
- **Preprocessing:** Enabled with automatic intensity
- **Parallel Processing:** Multi-threaded execution using available system cores
- **Time Limit:** Unlimited (allowing complete optimization)
- **Memory Management:** Dynamic allocation based on problem size
- **Computational Environment:** Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz, 8.00 GB DDR3 RAM, Windows (64-bit operating system, x64-based processor)

5.3.2 Experimental Design and Scenario Testing

Test Scenario Framework

To validate the mathematical model's robustness and verify its ability to handle diverse operational constraints, four comprehensive test scenarios were systematically designed. Each scenario employs synthetic datasets and different parameters with progressively increasing complexity, targeting specific aspects.

Scenario Specifications

Scenario 1: Basic Validation

- **Objective:** Fundamental model verification with minimal complexity.
- **Configuration:** 5 customers, with low to moderate demands and 2 vehicles.
- **Validation Focus:** Core routing logic, constraint satisfaction, and solution feasibility

Scenario 2: Capacity Stress Testing

- **Objective:** Evaluate model performance under high demand-to-capacity ratios.
- **Configuration:** 10 customers with total demand exceeds the capacity of available vehicles(2 vehicles).
- **Validation Focus:** Multi-trip necessity detection and optimal trip planning.

Scenario 3: Geographic Constraints

- **Objective:** Validate regional restrictions and vehicle-specific accessibility limitations.
- **Configuration:** 8 customers with different demand values and 5 vehicles
- **Special Constraints:**
 - 3 Southern customers(distant from depot)
 - Heavy truck restrictions: 2 Customers prohibit heavy vehicles.
- **Validation Focus:** Geographic constraint enforcement, vehicle-customer compatibility, and spatial optimization.

Scenario 4: Real-World Data Validation

- **Objective:** Model validation using company-representative parameters.
- **Customer Set:** $\{0, 1, 3, 8, 10, 15, 16, 32, 39, 40\}$ from the company database.
- **Fleet Composition:** 6 vehicles with heterogeneous starting times.

- **Starting Times:** [30, 30, 30, 30, 23, 23] (operational hours).
- **Demand Profile:** Real customer demands [8, 2, 15, 3, 5, 10, 12, 10, 11] pallets.
- **Operational Constraints:**
 - Southern region: Customers {32, 39, 40}.
 - Access restrictions: Customers {3, 10} exclude heavy vehicles.
 - Fleet classification: Vehicle 1 (20t capacity), Vehicles {1, 2, 5} (heavy category).
- **Validation Focus:** Real-world applicability, operational constraint integration, and scalability assessment.

Table 5.24: Core Performance Metrics for Test Scenarios

Test	Cost	Vehicles	Demand	Load %	Time (s)
1	29,750	50% (1/2)	100%	53.12%	0.09
2	90,632.5	100% (2/2)	100%	86%	0.58
3	289,400	60% (3/5)	100%	100%	1.51
4	242,645.8	83.33% (5/6)	100%	84.22%	2.20

Table 5.25: Operational Details and Special Cases

Test	Split Deliveries	Overnight	2nd Routes	Features
1	None	None	(T1×2)	Simple routing
2	None	None	(T1×2, T2×2)	Full fleet utilization
3	(C2: 7+1)	(T1)	(T3×2, T4×2)	Complex constraints
4	(C1: 3+5)	(T1)	(T4×2)	Mixed operations

Note: T = Truck, C = Customer

5.3.3 Results Analysis and Discussion

The experimental results presented in Tables 5.24 and 5.25 demonstrate the model's effectiveness across diverse operational scenarios, with several key insights emerging from the systematic testing framework.

Solution Quality and Optimality: The model successfully achieved 100% demand satisfaction across all test scenarios.

Fleet Utilization Efficiency: Vehicle usage rates varied significantly across scenarios, indicating the model's adaptive resource allocation capabilities.

Multi-Trip Operations Analysis: The model effectively identified multi-trip requirements across different scenarios. Tests 1 and 4 each utilized one vehicle for multi-trip operations (T1 and T4 respectively), while Tests 2 and 3 required two vehicles each to perform two routes (T1 and T2 in Test 2; T3 and T4 in Test 3).

Split Delivery Implementation: Split delivery functionality was selectively activated in Scenarios 3 and 4 where individual customer demands exceeded single-vehicle capacity constraints. In Test 3, Customer 2's demand (8 pallets) was partitioned into deliveries of 7 and 1 pallets across two vehicles. Similarly, in Test 4, Customer 1's demand (8 pallets) was split into deliveries of 3 and 5 pallets, enabling efficient capacity utilization while ensuring complete demand satisfaction.

Overnight Stay Optimization: The model identified that Vehicle T1 in both scenarios required overnight positioning due to the presence of southern customers and the considerable distance between these customers and the depot.

Loading Efficiency Metrics: Average loading rates demonstrate the model's capacity utilization optimization, ranging from 53.12% in the basic scenario to 100% in the geographically constrained scenario.

Computational Performance: Solution times remained highly efficient across all scenarios, with the most complex case (Scenario 4) requiring only 2.20 seconds.

5.3.4 Data Preparation Methodology

The optimization model utilizes a structured Excel database (`Clients.xlsx`) containing four core components for Celia's 41-node distribution network (40 customers + depot):

- **Customer profiles:** ID customer, Geographic coordinates and regional assignments
- **Distance matrix:** Symmetric 41×41 inter-customer distances (km)
- **Time matrix:** Symmetric 41×41 travel times (minutes)
- **Cost matrix:** 41×5 vehicle-specific service costs (DA)

Data extraction employs pandas DataFrame subsetting to generate problem-specific matrices for selected customer subsets, automatically formatted for CPLEX OPL input.

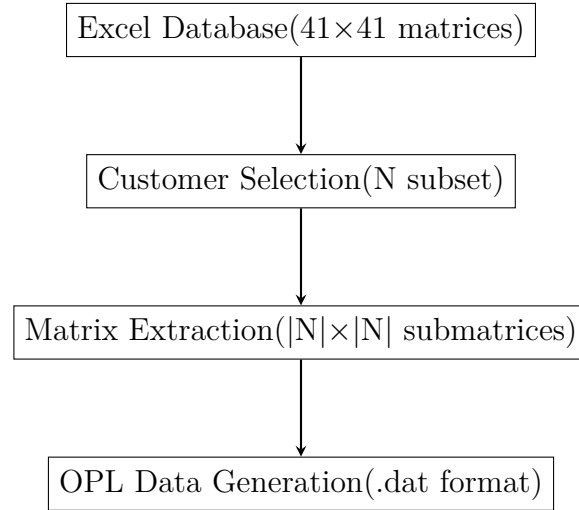


Figure 5.38: Data extraction and transformation workflow

The comprehensive OPL data file contains the following numerical components as detailed in Table 5.26.

Table 5.26: Complete OPL Data File Specification: Sets, Parameters, and Matrices

Component	Type	Definition
SETS DEFINITION		
Node Set	Set	$N = \{0, 1, 3, 8, 10, 15, 16, 32, 39, 40\}$
Vehicle Set	Set	$K = \{1, 2, 3, 4, 5, 6\}$
Route Set	Set	$R = \{1, 2\}$
SPECIALIZED SUBSETS		
Southern Customers	Subset	$N_south = \{32, 39, 40\}$
Heavy Vehicle Restrictions	Subset	$N_no_heavy = \{3, 10\}$
20-Ton Vehicles	Subset	$K_20t = \{1\}$
Heavy Vehicles	Subset	$K_heavy = \{1, 2, 5\}$
VEHICLE PARAMETERS		
Vehicle Capacities	Array[K]	$Q = [33, 16, 4, 7, 16, 4]$ (pallets)
Vehicle Start Times	Array[K]	$h_start = [30, 30, 30, 30, 23, 23]$ (hours)
Unloading Time Limits	Array[K]	$p = [2, 1.5, 0.5, 1, 1.5, 0.5]$ (hours)
Service Time Coefficients	Array[K]	$\beta = [0.05, 0.045, 0.0417, 0.0433, 0.045, 0.0417]$
CUSTOMER PARAMETERS		
Customer Demands	Array[N]	$d = [0, 8, 2, 15, 3, 5, 10, 12, 10, 11]$ (pallets)
Early Time Windows	Array[N]	$e = [0, 32, 32, 32, 32, 32, 32, 32, 32, 32]$ (hours)
Late Time Windows	Array[N]	$l = [0, 40, 40, 40, 40, 40, 40, 40, 40, 40]$ (hours)
NETWORK MATRICES		
Distance Matrix (First 2 rows)	Matrix[N][N]	$dist = [[0.0, 26.61, 39.04, 129.53, \dots],$ $[26.61, 0.0, 16.36, 108.90, \dots], \dots]$ (km)
Travel Time Matrix (First 2 rows)	Matrix[N][N]	$t = [[0.0, 0.444, 0.651, 2.159, \dots],$ $[0.444, 0.0, 0.273, 1.815, \dots], \dots]$ (hours)
Cost Matrix (First 2 rows)	Matrix[N][K]	$c = [[0.0, 0.0, 0.0, 0.0, 0.0, 0.0],$ $[16800, 10658, 3673, 3673, 10658, 3673], \dots]$ (DA)
OPERATIONAL CONSTRAINTS		
Maximum Radius	Parameter	$R_max = 500$ (km)
Number of Nodes	Parameter	$numNodes = 10$
Next Day Start Time	Parameter	$NEXT_DAY_START = 56$ (hours)
Big M Constant	Parameter	$M = 100$
Max Return Time (1st Tour)	Parameter	$MAX_RETURN_TIME_FIRST_TOUR = 36$ (hours)

Example of Output

```

â€œ 20-ton truck 1 used to serve customer 32 on route 1
â€œ 20-ton truck 1 used to serve customer 39 on route 1
â€œ 20-ton truck 1 used to serve customer 40 on route 1

ðŸ™• Total overnight stays: 2

=== SPLIT DELIVERIES CHECK ===
ðŸ™• Customer 1 has split delivery (2 parts)
    Truck 4 Route 1: 3
    Truck 4 Route 2: 5
ðŸ™• Total customers with split deliveries: 1

=== VEHICLE UTILIZATION ===
Vehicle 1: 1 route(s) assigned
Vehicle 2: 1 route(s) assigned
Vehicle 4: 2 route(s) assigned
Vehicle 5: 1 route(s) assigned
Vehicle 6: 1 route(s) assigned

=== DEMAND SATISFACTION CHECK ===
Customer 1: demand 8, delivered 8
Customer 3: demand 2, delivered 2
Customer 8: demand 15, delivered 15
Customer 10: demand 3, delivered 3
Customer 15: demand 5, delivered 5
Customer 16: demand 10, delivered 10
Customer 32: demand 12, delivered 12
Customer 39: demand 10, delivered 10
Customer 40: demand 11, delivered 11

=== OPTIMIZATION COMPLETED ===
Total objective value: 242645.8
    
```

(a) Solution Output Summary

```

ðŸ™• Truck 4, Route 1:
Clients: [1]
Quantities: [3]
Last customer: [1]
Total cost: 3672.9
=== SCHEDULE ===
Customer 1:
    Arrival: 32:00 (32h)
    Time window: [32:00 - 40:00]
    Service time: 8 min (max: 60 min)
    Departure: 32:07 (32.13h)
ðŸ™• Returned to depot: 32:34

ðŸ™• Truck 4, Route 2:
Clients: [1, 3]
Quantities: [5, 2]
Last customer: [3]
Total cost: 7172.9
=== SCHEDULE ===
Customer 1:
    Arrival: 34:01 (34.02h)
    Time window: [32:00 - 40:00]
    Service time: 13 min (max: 60 min)
    Departure: 34:14 (34.23h)
Customer 3:
    Arrival: 34:30 (34.51h)
    Time window: [32:00 - 40:00]
    Service time: 5 min (max: 60 min)
    Departure: 34:35 (34.59h)
ðŸ™• End of service (no return to depot)
    
```

(b) Multi-Route Configuration

Figure 5.39: CPLEX Optimization Results and Route Analysis

5.3.5 Computational Complexity Analysis

The implemented HF-OVRP-SD-TW-MT model belongs to the class of NP-hard combinatorial optimization problems. The complexity analysis reveals:

Decision Variables:

- Binary variables: $O(|N|^2 \cdot |K| \cdot |R|)$ for routing decisions (dominated by x variables)
- Continuous variables: $O(|N| \cdot |K| \cdot |R|)$ for time, quantity, and cost variables
- Total variables: $O(|N|^2 \cdot |K| \cdot |R|)$

Constraint Complexity:

- Flow conservation: $O(|N| \cdot |K| \cdot |R|)$ constraints
- Subtour elimination (MTZ): $O(|N|^2 \cdot |K| \cdot |R|)$ constraints
- Time propagation: $O(|N|^2 \cdot |K| \cdot |R|)$ constraints
- Capacity and routing: $O(|N| \cdot |K| \cdot |R|)$ constraints
- Total constraints: $O(|N|^2 \cdot |K| \cdot |R|)$

Empirical Verification: For our test scenario with $|N| = 10$, $|K| = 6$, $|R| = 2$:

- Variables: 2,061 total (1,449 binary + 612 continuous)
- Constraints: 4,743 (as confirmed by CPLEX solver statistics)
- Solution time: 2.20 seconds optimal
- Computational efficiency: 25,541 iterations, 983 branch-and-bound nodes

Operational Scalability Assessment: For Celia’s operational range (maximum 9-10 customers daily):

- Complete problem coverage with guaranteed optimal solutions
- Sub-second to 3 second solve times enable real-time re-optimization

5.3.6 Comparative Analysis Framework

To validate the effectiveness of our optimization model, we conduct a comprehensive comparison between the proposed solution and the existing company assignment strategy. This analysis provides empirical evidence of potential improvements in operational efficiency and cost reduction.

Company Assignment Data Preparation

Current Assignment Strategy Documentation The company’s existing assignment strategy operates through manual planning processes based on operational experience.

Data Collection and Standardization To ensure a fair comparison, we obtained the company’s assignment data. The data standardization process involves:

- **Route documentation:** Recording actual vehicle-customer assignments for the selected customer subset
- **Cost calculation:** Computing actual costs using the same cost matrix structure employed in our optimization model
- **Constraint validation:** Ensuring company assignments respect the same capacity and operational constraints as our model

Comparative Performance Analysis

Statistical Analysis We employ the percentage improvement calculation technique: $\text{Improvement} = \frac{\text{Company Cost} - \text{Optimized Cost}}{\text{Company Cost}} \times 100\%$

Results and Discussion

Quantitative Comparison Results Table 5.27 presents the comprehensive comparison between the company's current strategy and our optimization model:

Table 5.27: VRP Optimization Results: Cost and Operational Efficiency

Test	Manual Cost	Optimized Cost	Savings	Vehicles	Time
T1 (5 customers)	83,769.9	59,595.8	28%	4/4	0,31s
T2 (8 customers)	160,229.5	121,863.0	23%	4/5	0,67s
T3 (9 customers)	133,208.6	133,208.6	0%	7/8	1m 22s
Average	125,736	104,889	17%	-25%	27.7s

Operational Insights The comparative analysis reveals several key operational insights:

- **Cost optimization:** Achieved moderate cost reductions with 17% average improvement through optimal vehicle-customer matching
- **Resource utilization:** Reduced fleet requirements by 25% on average while improving vehicle capacity utilization
- **Computational performance:** Fast resolution times (0.31s-1m 22s) enable practical real-time implementation

The implemented VRP model demonstrates effectiveness in cost reduction and fleet utilization for Celia's daily routing operations. The CPLEX solver guarantees optimal solutions for the current small-scale instances typical of daily logistics operations.

Future scalability considerations include integrating heuristic and metaheuristic algorithms if problem instances increase beyond current parameters, ensuring maintained robustness and flexibility while preserving computational efficiency.

5.4 Model Limitations

Although the proposed VRP model is well-suited for real-world applications, certain operational aspects remain unaddressed in its current formulation. If not properly modeled, these aspects can lead to solutions that are theoretically optimal but impractical in day-to-day operations. The most notable limitations are as follows:

5.4.1 Unmet Demand and Postponement

In the current model, the constraint ensures that all customer demands are satisfied on the same day. In practice, however, unmet demand often occurs, with orders postponed to the following day ($D + 1$). We intentionally did not introduce an explicit

unmet-demand variable in the current formulation because, without a dedicated modeling mechanism, the solver may decide not to serve some demands even when sufficient capacity exists, postponing them instead in order to minimize the total costs. This is undesirable in operational contexts, as it affects service quality and customer satisfaction.

To address this issue, a **penalty term for unmet or postponed demand** should be integrated into the objective function. This penalty would encourage the model to prioritize fulfilling daily demand whenever possible, postponing orders only when absolutely necessary.

5.4.2 Second Route Utilization and Overnight Stays

The inclusion of a second delivery route in the model generally improves vehicle utilization and reduces the total number of trucks required. However, in certain cases, it results in a large number of overnight stays, particularly when vehicles are dispatched for a second route to minimize truck usage. While cost-effective from a purely routing perspective, this approach may not be operationally desirable.

To mitigate this, a **penalty for overnight stays** should be incorporated into the objective function. This would ensure that second routes are used judiciously and that overnight operations occur only when no other feasible option is available.

5.5 App Web Architecture

The app web represents a comprehensive logistics optimization platform that integrates multiple technologies to solve real-world vehicle routing problems while providing data analytics capabilities. The system architecture combines modern web technologies with specialized optimization tools to create an application that handles demand forecasting, customer analytics, and route optimization with real-time data synchronization.

5.6 Technology Stack Overview

Category	Technology	Purpose
Frontend	React.js + Hooks	Component-based UI with state management
	Tailwind CSS	Responsive design and styling
	Chart.js	Data visualization and interactive charts
	WebSocket	Real-time bidirectional communication

Table 5.28: Frontend Technology Stack

Category	Technology	Purpose
Backend	FastAPI	RESTful API development with async support
	Pandas/NumPy	Data processing and mathematical operations
	JWT + bcrypt	Authentication and password security
Optimization	IBM CPLEX	Vehicle routing problem solving
	Custom algorithms	Customer clustering and demand forecasting
Data	Excel integration (openpyxl)	Primary data source handling
	JSON files	Configuration and user data storage
	ReportLab	PDF report generation

Table 5.29: Backend, Optimization, and Data Technology Stack

This streamlined technology stack forms the foundation for implementing a comprehensive logistics optimization solution with real-time analytics capabilities.

5.6.1 Core Architectural Components

LogiOptim Pro follows an architecture that integrates all platform functionalities through a unified data flow and a real-time communication system. **Frontend Integration Hub:** The React-based interface provides six main functional modules:

- **Dashboard:** Real-time statistics, demand visualization, customer analytics
- **VRP Optimization:** Customer/truck selection, route optimization, results visualization
- **Advanced Analytics:** Customer clustering, seasonal patterns, forecasting
- **Order Management:** Order tracking, Excel synchronization, report generation
- **KPI Analytics:** Performance metrics, operational insights, and system efficiency monitoring
- **Alert System:** Intelligent monitoring and notification broadcasting

Data Flow and Integration

The platform implements a unified data pipeline where Excel serves as the single source of data, processed through standardized cleaning algorithms, distributed via WebSocket for real-time updates, and enhanced through specialized engines (analytics, optimization, reporting) to serve all functional modules seamlessly.

Data Reporting The system automatically generates weekly reports every Monday at 7:00 AM and sends them via email (when enabled) with key business

metrics, the top 5 customers, week-over-week performance changes, and dashboard access links. The email includes PDF and Excel attachments containing comprehensive analytics: executive summaries, customer rankings, regional performance breakdowns, daily statistics, and raw data across multiple worksheets for detailed business intelligence analysis.

5.7 Implementation Details

5.7.1 Authentication System

The authentication system implements JWT-based security with role-based access control (Admin, User, Viewer).

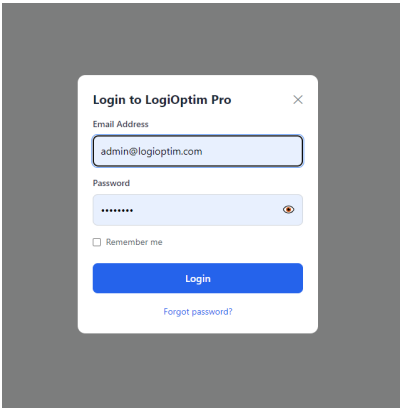


Figure 5.40: Login Interface

5.7.2 Dashboard Implementation

The dashboard module serves as the central hub for real-time demand analytics and business intelligence, implementing a comprehensive multi-component interface with interconnected analytical views.

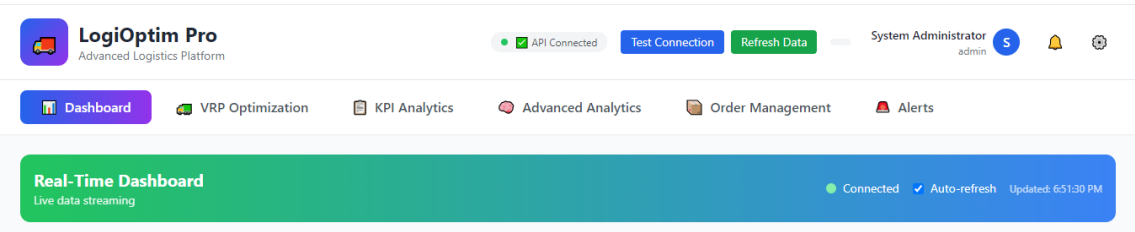
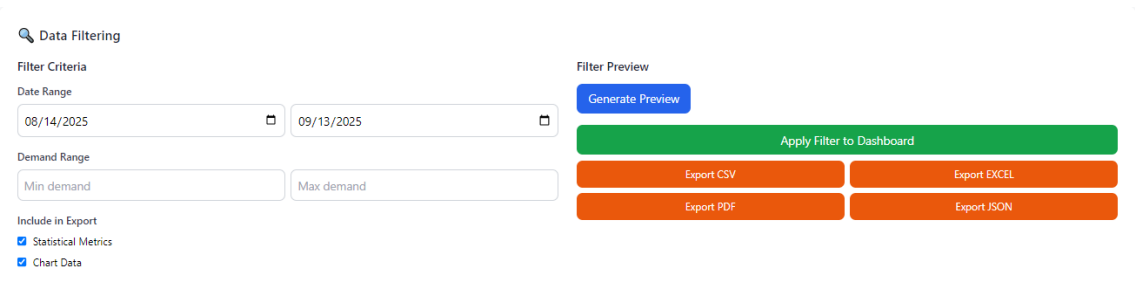
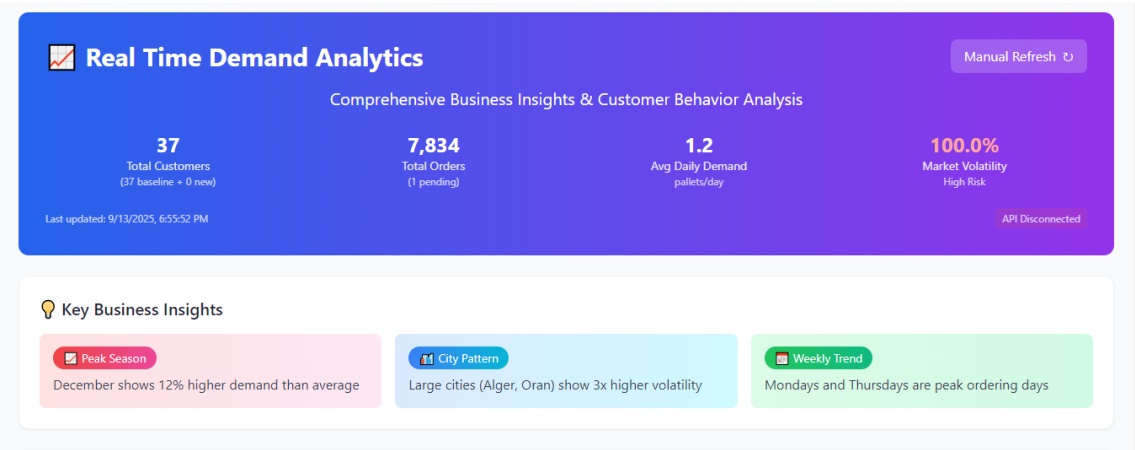


Figure 5.41: Real-time dashboard navigation with connectivity status monitoring

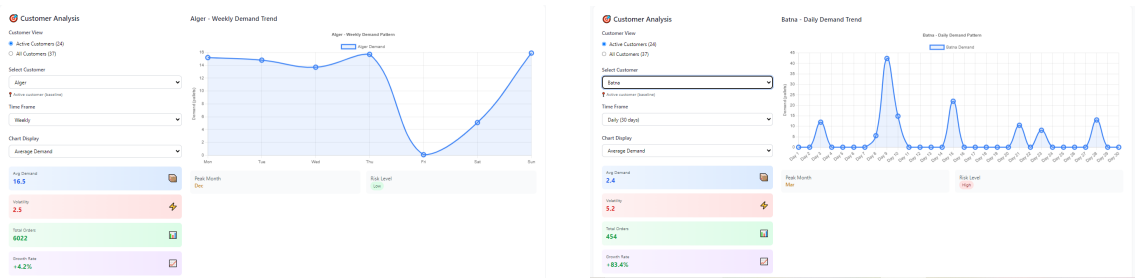
The system analyzes demand distribution patterns and gap frequencies to identify ordering behaviors, while radar charts visualize weekly demand patterns across different time periods as presented in figure 5.43. While the figure 5.44 presents



(a) Advanced filtering system with export capabilities



(b) Real-time analytics dashboard with business insights



(c) Alger - Weekly demand pattern analysis

(d) Batna - Average daily demand analysis

Figure 5.42: Dashboard System Components - Analytics and Customer Analysis

customer performance rankings and risk assessment matrix for strategic decision support.

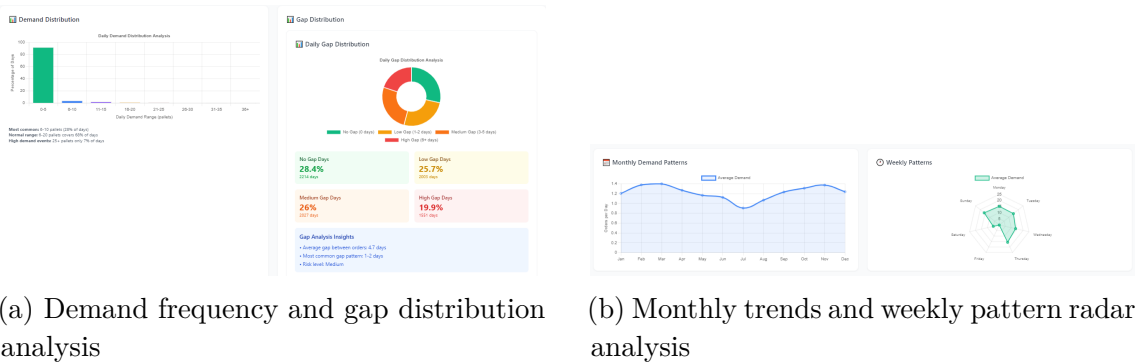


Figure 5.43: Distribution Analysis and Temporal Patterns

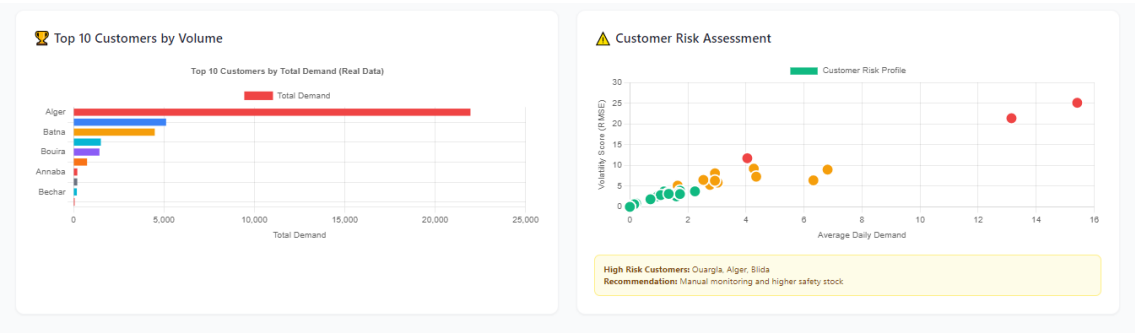


Figure 5.44: Customer volume ranking and risk assessment matrix

5.7.3 VRP Optimization Implementation

The Vehicle Routing Problem (VRP) module integrates IBM CPLEX solver with an interactive web interface to solve real-world logistics optimization challenges through a systematic workflow process.

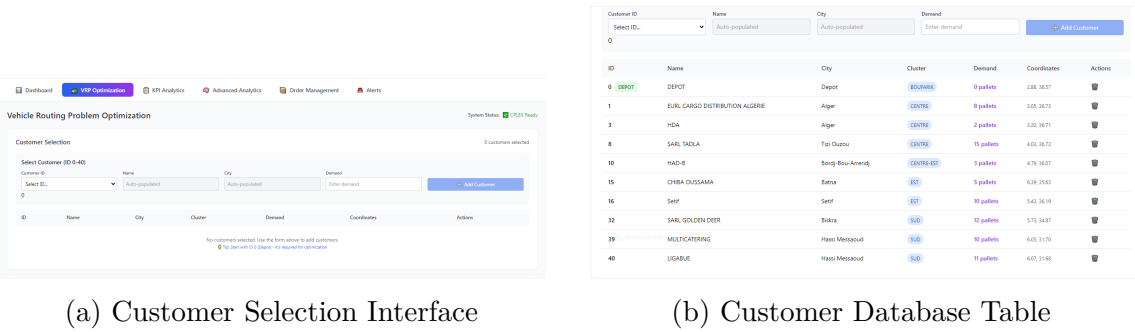
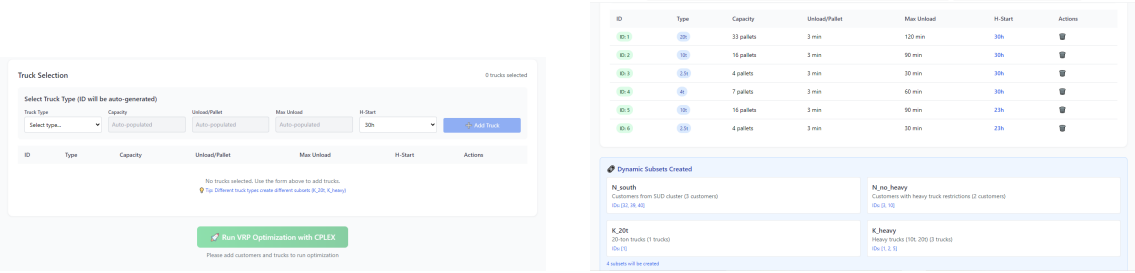


Figure 5.45: Table of selected customers



(a) Truck Selection Interface (b) Fleet Configuration Table

Figure 5.46: Vehicle Selection and Automatic Dynamic Subsets Creation

Optimization Execution and Results

Real-time Processing: When the user initiates the process by selecting Run VRP Optimization with CPLEX, the system executes the optimization and returns results in real time. The main outputs are as follows:

1. **Rapid computation:** The solution is computed within seconds, ensuring minimal processing delay.
2. **Performance metrics table:** A summary table is generated containing total cost per truck, return time, vehicle utilization, and loading rate.
3. **Interactive visualization:**
 - A Folium-based interactive map displaying optimized routes.
 - Color-coded paths for each vehicle with dynamic route tracing.
 - Customer location markers with detailed service information.

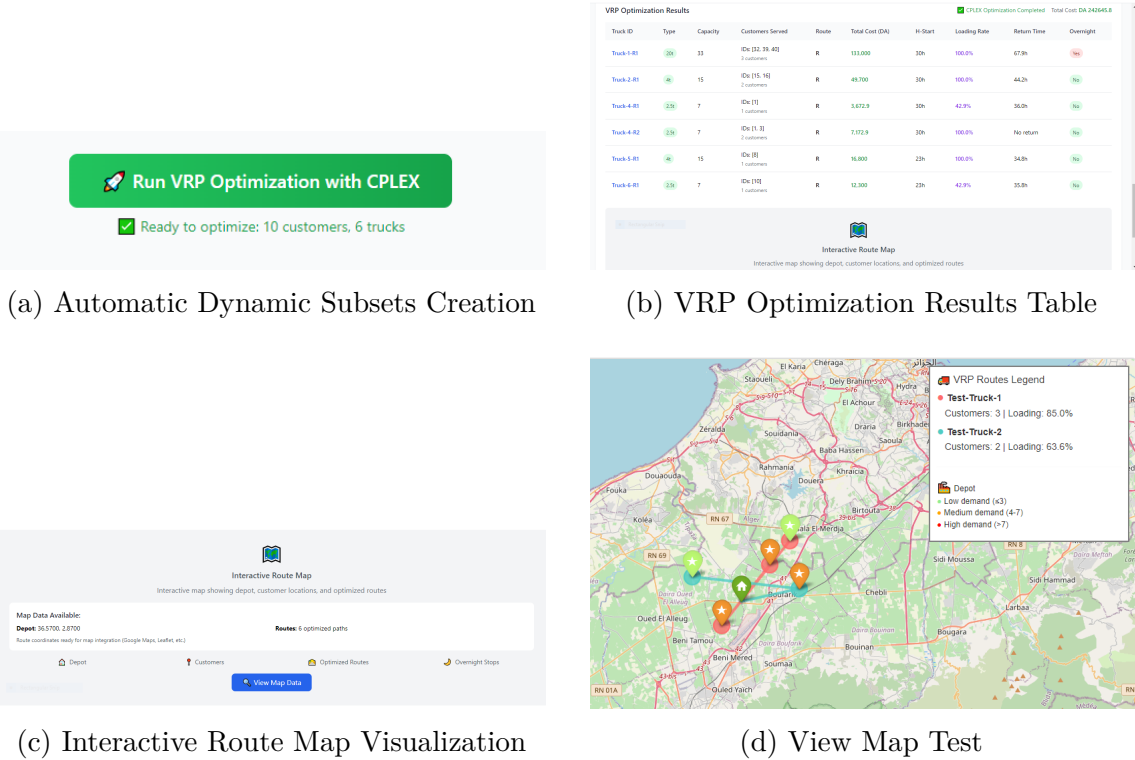


Figure 5.47: VRP Optimization Process: Subset Generation, Execution, and Results

Upon completion of the process, the objective value from the most recent optimization is displayed at the top of the page, as illustrated in Figure 5.48.

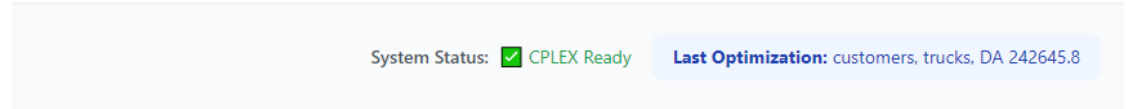


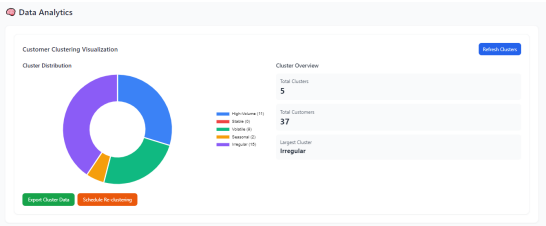
Figure 5.48: Objective value of the most recent optimization process

5.7.4 Advanced Analytics Implementation

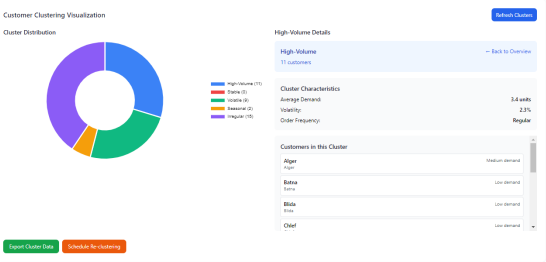
The Advanced Analytics module provides sophisticated data analysis capabilities through machine learning algorithms and statistical methods, enabling strategic business intelligence and predictive insights.

Customer Clustering and Behavioral Analysis

The clustering system implements rule-based classification algorithms to segment 37 customers into five distinct behavioral groups: High-Volume, Stable, Volatile, Seasonal, and Irregular. Interactive cluster exploration provides detailed characteristics including average demand, volatility metrics, and customer listings.

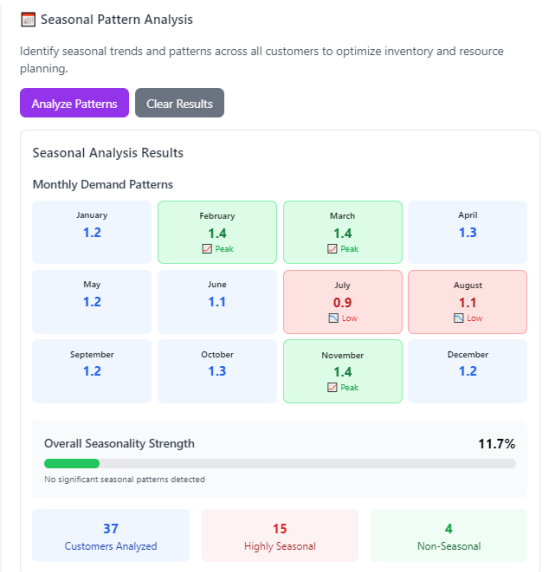


(a) Customer clustering distribution across 5 segments

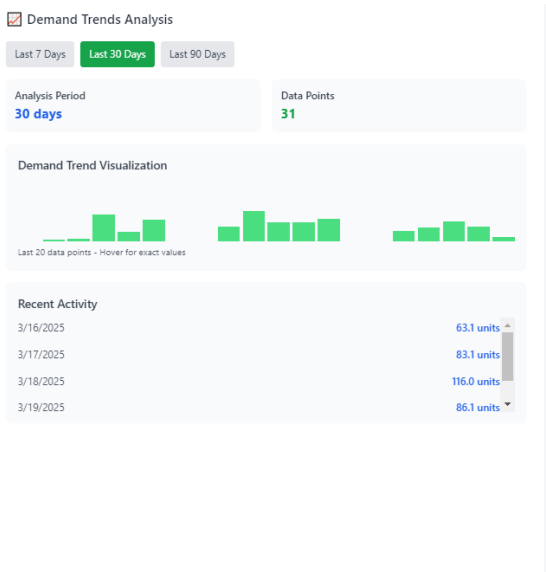


(b) High-Volume cluster detailed analysis

Figure 5.49: Customer Clustering Visualization and Analysis



(a) Seasonal pattern analysis with monthly demand visualization



(b) Geographic distribution and AI forecasting interface

Figure 5.50: Temporal Analysis and Predictive Analytics

Seasonal and Trend Analysis

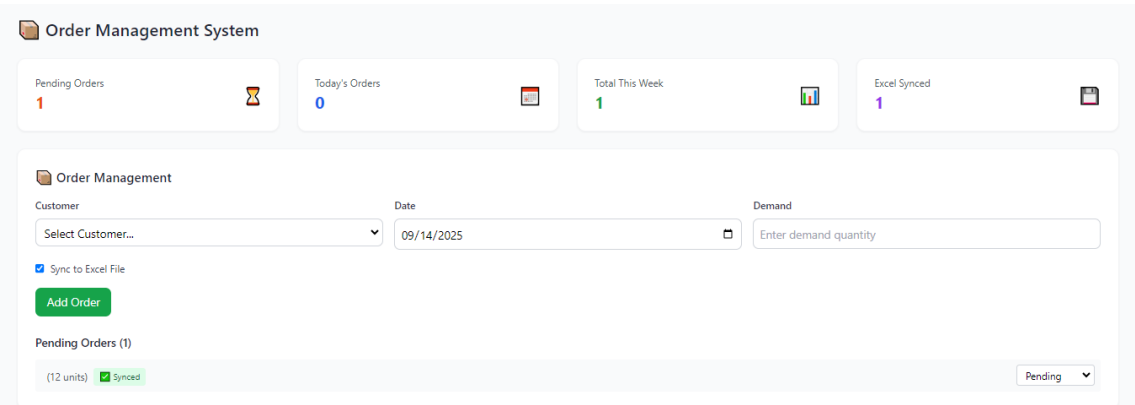
Seasonal Analysis: The system identifies peak months (February, March, November) and low-demand periods (July, August) with 11.7% overall seasonality strength across 37 customers.

Forecasting Capabilities: AI-powered demand forecasting provides customer-specific predictions with configurable timeframes (7, 30, 90 days).

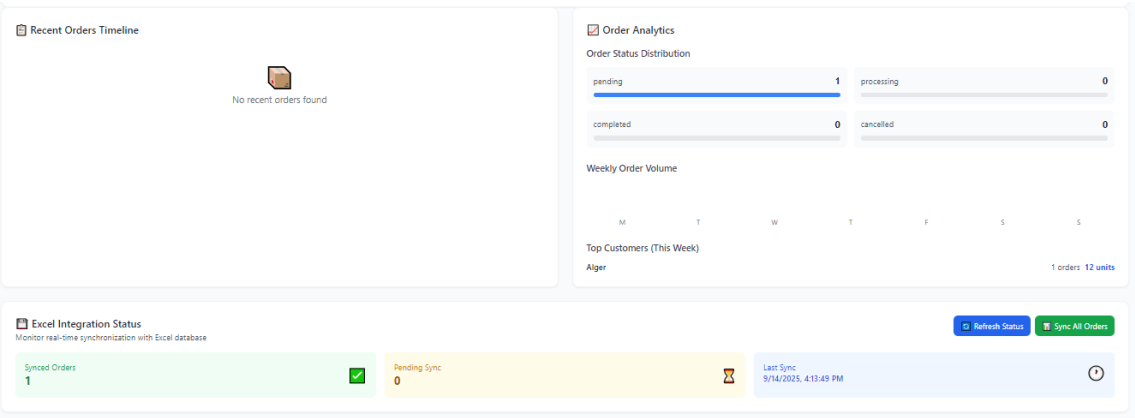
5.7.5 System Management and Monitoring

We implemented a comprehensive system management through integrated order processing, and performance analytics modules.

Order Management and Excel Integration



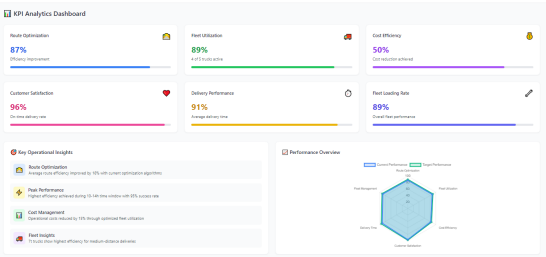
(a) Order management interface with real-time Excel synchronization



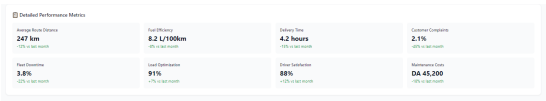
(b) Order analytics with timeline and Excel integration status

Figure 5.51: Order Processing System with Analytics Integration

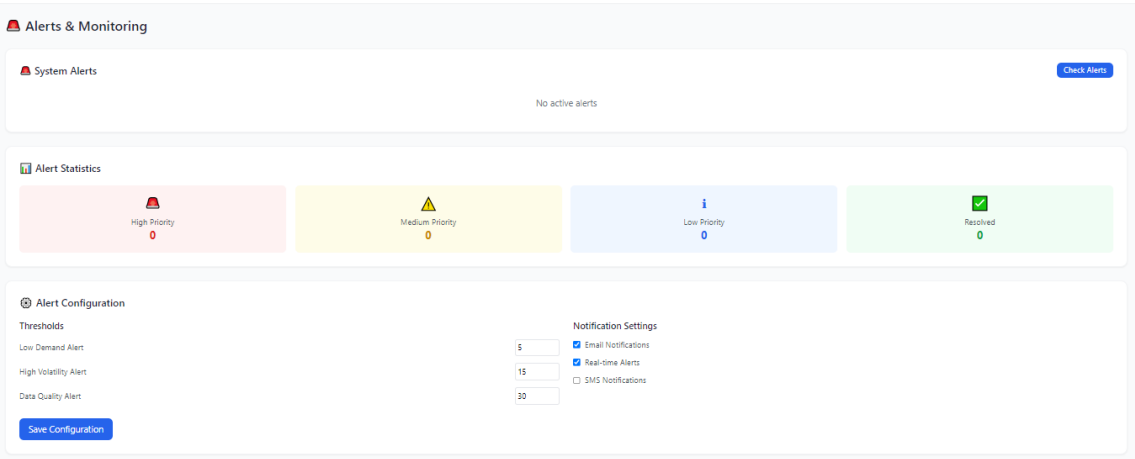
The order management system provides real-time order creation with automatic Excel synchronization, tracking 1 pending order with Excel sync status monitoring. The system displays comprehensive statistics including weekly volumes, status distribution, and top customer analysis.



(a) KPI analytics with performance metrics



(b) Detailed performance metrics with analysis



(c) Alert management with configurable thresholds

Figure 5.52: System Monitoring and Performance Analytics

Alerts Monitoring and KPI Analytics

KPI Analytics: Provides comprehensive performance tracking across six key metrics: Route Optimization, Fleet Utilization, Cost Efficiency, Customer Satisfaction , Delivery Performance, and Fleet Loading Rate.

Performance Monitoring: Detailed metrics tracking includes route distance optimization, fuel efficiency, delivery time reduction, and maintenance cost control.

Alert System: Implements threshold-based monitoring with configurable parameters (Low Demand, High Volatility, Data Quality) and multi-channel notifications (Email, Notification, SMS).

Conclusion

This chapter presented the implementation of a comprehensive VRP optimization model tailored to Celia’s distribution operations, encompassing mathematical formulation, CPLEX-based solution methodology, and systematic validation across operational scenarios. Comparative analysis reveals potential cost reductions of up to 28% over manual planning approaches. The integrated web-based platform demonstrates the practical deployment of the optimization framework, providing

real-time routing capabilities and comprehensive analytics for enhanced distribution management.

General Conclusion

This thesis addressed the optimization of distribution operations for Celia through a comprehensive approach integrating demand forecasting and vehicle routing optimization. The research encompassed demand prediction using both classical time series methods and advanced machine learning techniques, mathematical modeling of complex routing problems, and the development of an integrated web-based optimization platform.

The demand forecasting analysis revealed that XGBoost outperformed classical approaches when incorporating temporal features, though prediction accuracy remains constrained by the inherent randomness in customer ordering behavior. The heterogeneous ordering patterns, high demand volatility, and weak temporal structures present fundamental challenges that transcend methodological improvements. The Vehicle Routing Problem implementation successfully addresses Celia's complex operational constraints through a comprehensive HF-OVRP-SD-TW-MT formulation, achieving optimal solutions within seconds for daily operational scales. The integrated web application provides real-time optimization capabilities, comprehensive analytics, and user-friendly interfaces for practical deployment in daily operations.

The research identified critical organizational limitations that constrain both forecasting accuracy and operational efficiency. Customer ordering behavior exhibits excessive randomness and volatility that fundamentally limits predictive capabilities, regardless of algorithmic sophistication. The absence of structured communication protocols among stakeholders (customers, depot, 3PL providers, factory) creates operational inefficiencies and unpredictable resource demands. Current fleet management lacks real-time tracking and coordination systems, leading to vehicle availability uncertainties and suboptimal dispatching decisions.

To enhance forecasting accuracy and operational efficiency, Celia should implement structured ordering protocols with customers, negotiating fixed ordering schedules based on demand pattern analysis to facilitate predicting both timing and demand quantities. Improved customer stock management practices by encouraging customers to maintain better inventory control systems would avoid the randomness of order values that currently undermines forecasting accuracy.

The establishment of comprehensive communication frameworks among all stakeholders would facilitate better planning coordination. Implementation of real-time

fleet tracking, Transportation Management Systems (TMS) and formal agreements with 3PL(Third-Party Logistics) providers would improve vehicle dispatching and availability management, avoiding situations where insufficient trucks are available for daily operations. Additionally, customer satisfaction monitoring and fuel consumption tracking would provide comprehensive operational oversight.

These improvements would lead to better resource allocation, enhanced stock management at the depot to avoid congestion and poor organization, and reduced stress for depot supervisors.

Several technical improvements could enhance the system's capabilities and scalability. Integration of SQL databases would provide greater flexibility and efficiency compared to Excel-based data management. The VRP model could benefit from penalty mechanisms for unmet demand and overnight stays when they exist, to improve operational realism. As business scales increase, implementing approximate solution methods would maintain computational tractability for larger problem instances. The web platform requires database integration, cloud hosting implementation, and enhanced KPI monitoring capabilities with real-time operational data. Future research should also investigate multi-objective optimization frameworks that explicitly balance cost efficiency with service quality metrics.

This research demonstrates that while technological optimization tools provide significant operational improvements, fundamental organizational and communication challenges must be addressed to realize their full potential in real-world logistics operations.

Appendix

The following results correspond to the test scenario involving 10 customers and 6 truck types: The following figure shows the OPL data file (.dat):

```
1 N = {0, 1, 3, 8, 10, 15, 16, 32, 39, 40};
2 K = {1, 2, 3, 4, 5, 6};
3 R = {1, 2};
4 h_start = {30, 30, 30, 30, 23, 23}; // Starting hours (23=23:00 D, 30=06:00 D+1)
5 N_south = {32, 39, 40}; // Southern region nodes
6 N_no_heavy = {3, 10}; // Nodes that cannot handle heavy vehicles
7 K_20t = {1}; // 20-ton vehicles
8 K_heavy = {1, 2, 5}; // Heavy vehicles
9 d = {0, 8, 2, 15, 3, 5, 10, 12, 10, 11}; // Node demands (pallets)
10 Q = {33, 16, 4, 7, 16, 4}; // Vehicle capacities (in pallets)
11 c = [
12   [0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
13   [16800.0, 10657.5, 3672.9, 3672.9, 10657.5, 3672.9],
14   [16800.0, 10657.5, 3672.9, 3672.9, 10657.5, 3672.9],
15   [26250.0, 16800.0, 7345.8, 8570.1, 16800.0, 7345.8],
16   [31500.0, 22000.0, 12300.0, 13200.0, 22000.0, 12300.0],
17   [66000.0, 46200.0, 16324.0, 20521.0, 46200.0, 16324.0],
18   [47250.0, 29400.0, 12243.0, 15225.0, 29400.0, 12243.0],
19   [73500.0, 56700.0, 22281.0, 31500.0, 56700.0, 22281.0],
20   [126000.0, 99750.0, 41181.0, 61215.0, 99750.0, 41181.0],
21   [126000.0, 99750.0, 41181.0, 61215.0, 99750.0, 41181.0]
22 ]; // Cost matrix c[i][k]
23 dist = [
24   [0.0, 26.613922, 39.039014, 129.529469, 219.972253, 404.939943, 286.682646, 369.038654, 644.459027, 647.882746],
25   [26.613922, 0.0, 16.355806, 108.895567, 206.851076, 391.361668, 270.851689, 362.393945, 649.369151, 652.779971],
26   [39.039014, 16.355806, 0.0, 92.707305, 190.925799, 375.265733, 254.555881, 347.846127, 639.16021, 642.562669],
27   [129.529469, 108.895567, 92.707305, 0.0, 111.222947, 289.425844, 166.101339, 279.069904, 599.472704, 602.78262],
28   [219.972253, 206.851076, 190.925799, 111.222947, 0.0, 184.974, 71.807507, 169.15188, 503.183321, 506.384503],
29   [404.939943, 391.361668, 375.265733, 289.425844, 184.974, 0.0, 124.147203, 112.371518, 435.732767, 438.271671],
30   [286.682646, 270.851689, 254.555881, 166.101339, 71.807507, 124.147203, 0.0, 149.86062, 500.795457, 503.787642],
31   [369.038654, 362.393945, 347.846127, 279.069904, 169.15188, 112.371518, 149.86062, 0.0, 351.699211, 354.630344],
32   [644.459027, 649.369151, 639.16021, 599.472704, 503.183321, 435.732767, 500.795457, 351.699211, 0.0, 3.432849],
33   [647.882746, 652.779971, 642.562669, 602.78262, 506.384503, 438.271671, 503.787642, 354.630344, 3.432849, 0.0]
34 ]; // Distance matrix
35 p = [2, 1.5, 0.5, 1, 1.5, 0.5]; // Unloading time limits by vehicle type
36 e = [0, 32, 32, 32, 32, 32, 32, 32, 32, 32]; // Time windows start
37 l = [0, 40, 40, 40, 40, 40, 40, 40, 40, 40]; // Time windows end
38 beta = [0.05, 0.045, 0.0417, 0.0433, 0.045, 0.0417]; // Service time (unloading)
39 t = [
40   [0.0, 0.44356537, 0.65065023, 2.15882448, 3.66620422, 6.74899905, 4.7780441, 6.15064423, 10.74098378, 10.79804577],
41   [0.44356537, 0.0, 0.27259677, 1.81492612, 3.44751793, 6.52269447, 4.51419482, 6.03989908, 10.82281918, 10.87966618],
42   [0.65065023, 0.27259677, 0.0, 1.54512175, 3.18209665, 6.25442888, 4.24259802, 5.79743545, 10.65267017, 10.70937782],
43   [2.15882448, 1.81492612, 1.54512175, 0.0, 1.85371578, 4.82376407, 2.76835565, 4.65116507, 9.99121173, 10.046377],
44   [3.66620422, 3.44751793, 3.18209665, 1.85371578, 0.0, 3.0829, 1.19679178, 2.819198, 8.38638868, 8.43974172],
45   [6.74899905, 6.52269447, 6.25442888, 4.82376407, 3.0829, 0.0, 2.06912005, 1.87285863, 7.26221278, 7.30452785],
46   [4.7780441, 4.51419482, 4.24259802, 2.76835565, 1.19679178, 2.06912005, 0.0, 2.497677, 8.34659095, 8.3964607],
47   [6.15064423, 6.03989908, 5.79743545, 4.65116507, 2.819198, 1.87285863, 2.497677, 0.0, 5.86165352, 5.91050573],
48   [10.74098378, 10.82281918, 10.65267017, 9.99121173, 8.38638868, 7.26221278, 8.34659095, 5.86165352, 0.0, 0.05721415],
49   [10.79804577, 10.87966618, 10.70937782, 10.046377, 8.43974172, 7.30452785, 8.3964607, 5.91050573, 0.05721415, 0.0]
50 ]; // Travel time matrix
51 R_max = 500; // Maximum radius
52 numNodes = 10; // Number of nodes
53 NEXT_DAY_START = 56; // 08:00 next day (D+2)
54 M = 100; // Big M reduced to avoid aberrant values
55 MAX_RETURN_TIME_FIRST_TOUR = 36; // Return limit for 1st tour (12:00 D+1)
```

Figure 53: Data file

Table 30: CPLEX Solution Summary for Test Scenario 4

Parameter	Value
Optimal Objective Value	242,645.8
Solution Time	2.20 seconds
Total Iterations	25,641
Branch-and-Bound Nodes	983
Variables (Total/Binary/Other)	2,061 / 1,449 / 612
Constraints	4,743
Non-zero Coefficients	15,081

Table 31: Preprocessing and Problem Reduction

Stage	Rows	Columns	Nonzeros
Original Problem	2,188	1,140	–
After Preprocessing	2,432	798	6,313
Final Reduced Problem	1,843	631	4,807
Reduction Percentage	15.8%	44.6%	24.0%

	Nodes					Cuts/		
	Node	Left	Objective	IInf	Best Integer	Best Bound	ItCnt	Gap
*	0+	0			312513,8000	36800,0000		88,22%
	0	0	208572,7901	71	312513,8000	208572,7901	373	33,26%
*	0+	0			282730,4000	208572,7901		26,23%
	0	0	213999,0696	76	282730,4000	Cuts: 112	496	24,31%
*	0+	0			272316,8000	213999,0696		21,42%
	0	0	222010,2743	74	272316,8000	Cuts: 56	646	18,47%
	0	0	222868,0869	76	272316,8000	Cuts: 39	691	18,16%
	0	0	223574,5170	102	272316,8000	Cuts: 36	757	17,90%
	0	0	223870,2056	112	272316,8000	Cuts: 33	805	17,79%
	0	0	224191,8781	106	272316,8000	Cuts: 49	874	17,67%
	0	0	224381,7137	97	272316,8000	Cuts: 31	930	17,60%
	0	0	224586,8225	91	272316,8000	Cuts: 31	957	17,53%
	0	0	224586,8225	92	272316,8000	Cuts: 3	960	17,53%
*	0+	0			257375,4000	224586,8225		12,74%
Detecting symmetries...								
	0	2	224586,8225	77	257375,4000	227010,4967	960	11,80%
Elapsed time = 0,67 sec. (225,46 ticks, tree = 0,02 MB, solutions = 4)								
*	7+	2			249851,4000	227010,4967		9,14%
*	8+	1			242645,8000	227010,4967		6,44%
	567	158	240360,5667	13	242645,8000	234339,0818	16703	3,42%

Figure 54: CPLEX Output1

```

Clique cuts applied: 7
Cover cuts applied: 4
Implied bound cuts applied: 212
Flow cuts applied: 3
Mixed integer rounding cuts applied: 44
Zero-half cuts applied: 7
Lift and project cuts applied: 7
Gomory fractional cuts applied: 2

Root node processing (before b&c):
  Real time           =    0,64 sec. (224,90 ticks)
Parallel b&c, 4 threads:
  Real time           =    1,56 sec. (351,88 ticks)
  Sync time (average) =    0,41 sec.
  Wait time (average) =    0,00 sec.
-----
Total (root+branch&cut) =    2,20 sec. (576,78 ticks)

```

Figure 55: CPLEX Output2

Table 32: Cut Generation Statistics

Cut Type	Applied
Clique Cuts	7
Cover Cuts	4
Implied Bound Cuts	212
Flow Cuts	3
Mixed Integer Rounding	44
Zero-half Cuts	7
Gomory Fractional	2

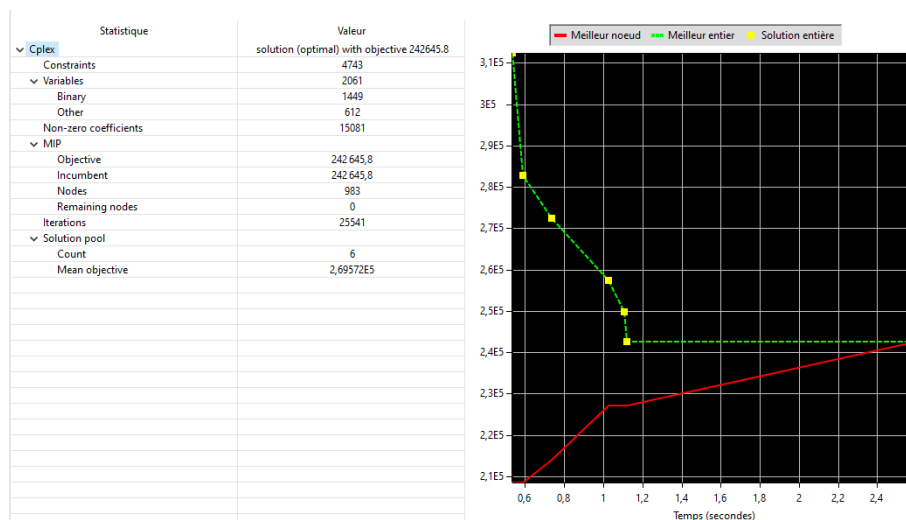


Figure 56: CPLEX Solution Convergence and Objective Value Evolution

Table 33: Vehicle Route Assignments and Performance

Vehicle	Customers Served	Routes	Total Cost	Utilization
Truck 1	[32, 39, 40]	1	133,000	100%
Truck 2	[15, 16]	1	49,700	31.25%
Truck 4	[1], [1, 3]	2	10,845.8	Multi-trip
Truck 5	[8]	1	16,800	93.75%
Truck 6	[10]	1	12,300	75%

```

=== SPLIT DELIVERIES CHECK ===
δY'' Customer 1 has split delivery (2 parts)
      Truck 4 Route 1: 3
      Truck 4 Route 2: 5
δY'' Total customers with split deliveries: 1

=== VEHICLE UTILIZATION ===
Vehicle 1: 1 route(s) assigned
Vehicle 2: 1 route(s) assigned
Vehicle 4: 2 route(s) assigned
Vehicle 5: 1 route(s) assigned
Vehicle 6: 1 route(s) assigned

=== DEMAND SATISFACTION CHECK ===
Customer 1: demand 8, delivered 8
Customer 3: demand 2, delivered 2
Customer 8: demand 15, delivered 15
Customer 10: demand 3, delivered 3
Customer 15: demand 5, delivered 5
Customer 16: demand 10, delivered 10
Customer 32: demand 12, delivered 12
Customer 39: demand 10, delivered 10
Customer 40: demand 11, delivered 11

=== OPTIMIZATION COMPLETED ===
Total objective value: 242645.8

```

Figure 57: CPLEX Split Delivery and Demand Satisfaction Verification

Table 34: Optimization Performance Indicators

Metric	Value	Status
Demand Satisfaction	100%	✓ Complete
Fleet Utilization	84.22%	✓ Efficient
Multi-trip Usage	1 vehicle	✓ Optimal
Geographic Constraints	All met	✓ Compliant
Time Window Violations	0	✓ Feasible
Solution Optimality Gap	0.00%	✓ Optimal

Client_id	customer_name	Ville	cluster	Long	Lat
0	DEPOT	Depot	BOUFARIK	36.57074958758743	2.8756046385743286
1	EURL CARGO DISTRIBUTION ALGERIE	Alger	CENTRE	36.73365764019669	3.05115471207546
2	SARL JALDI PLUS	Alger	CENTRE	36.64042352161425	3.0442585232349897
3	HDA	Alger	CENTRE	36.71279067839394	3.196762796255922
4	FOUR WEEKS	Alger	CENTRE	36.73965463976637	3.1427518790616746
5	BENDALI BRAHAM MOHAMED	Blida	CENTRE	36.485360211335994	2.8441016078805728
6	EURL BMN DISTRIBUTION	Medea	CENTRE	36.26970247524983	2.760682371826111
7	SARL MASYM DISTRIBUTION	M'Sila	CENTRE	35.720018967810944	4.523883529973087
8	SARL TADLA	Tizi Ouzou	CENTRE	36.7238244601289	4.03042689802958

Figure 58: Overview of custome's dataset using in VRP

Client_id	2,5T	4T	7T	10T	20T
0	-	-	-	-	-
1	3 672,90	3 672,90	6 930,00	10 657,50	16 800,00
2	3 672,90	3 672,90	6 930,00	10 657,50	16 800,00
3	3 672,90	3 672,90	6 930,00	10 657,50	16 800,00
4	3 672,90	3 672,90	6 930,00	10 657,50	16 800,00
5	2 625,00	2 625,00	5 250,00	8 400,00	13 125,00
6	4 893,00	6 993,00	8 925,00	12 600,00	18 900,00

Figure 59: Overview of cost matrix

date	city	orders
2020-01-02 00:00:00	Ain-Temouchent	0
2020-01-02 00:00:00	Alger	0
2020-01-02 00:00:00	Annaba	0
2020-01-02 00:00:00	Batna	0
2020-01-02 00:00:00	Bechar	0
2020-01-02 00:00:00	Bejaia	0
2020-01-02 00:00:00	Biskra	0
2020-01-02 00:00:00	Blida	3,75
2020-01-02 00:00:00	Bordj-Bou-Arreridj	0
2020-01-02 00:00:00	Bouira	0
2020-01-02 00:00:00	Boumerdes	0
2020-01-02 00:00:00	Chlef	0
2020-01-02 00:00:00	Constantine	12,93
2020-01-02 00:00:00	Djelfa	0
2020-01-02 00:00:00	El-Oued	0
2020-01-02 00:00:00	El-Taref	7,54
2020-01-02 00:00:00	Ghardaia	0
2020-01-02 00:00:00	Guelma	0
2020-01-02 00:00:00	Jijel	0
2020-01-02 00:00:00	M'Sila	0
2020-01-02 00:00:00	Mascara	0
2020-01-02 00:00:00	Medea	0
2020-01-02 00:00:00	Mostaganem	0

Figure 60: Historical data overview

Bibliography

- [1] Lactalis Group. Official website of lactalis group, 2025.
- [2] GoKid. Celia algérie, 2025.
- [3] User GitHub. Time series decomposition - visual explanation, 2018. Accessed: 2025-08-04.
- [4] Damaris K Whitney. Use example to describe multiplicative and additive time series relationships, 2022. Accessed: 2025-08-04.
- [5] FrePPLe. Classification of demand patterns based on variability in timing and quantity, n.d. Accessed: 2025-08-04.
- [6] Isha Salian. Supervize me: What’s the difference between supervised, unsupervised, semi-supervised and reinforcement learning. <https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/>, August 2018. NVIDIA Blog, accessed on 2025-08-06.
- [7] Unknown. The architecture of a typical decision tree [figure]. https://www.researchgate.net/figure/The-architecture-of-a-typical-decision-tree_fig7_370763536, n.d. Accessed via ResearchGate figure page.
- [8] Buggy Programmer. Difference between bagging and boosting | ensemble method. <https://buggyprogrammer.com/difference-between-bagging-and-boosting/>, n.d. Accessed: 2025-08-06.
- [9] Miao Zou, Wu-Gui Jiang, Qing-Hua Qin, Yu-Cheng Liu, and Mao-Lin Li. Optimized xgboost model with small dataset for predicting relative density of ti-6al-4v parts manufactured by selective laser melting. *Materials*, 15(15):5298, 2022. MDPI Open Access.
- [10] H. C. Henrieta, N. Hornakova, and D. Babcanova. Use of operational research methods in logistics. In *Proceedings of the Carpathian Logistics Congress*, 2015.
- [11] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. John Wiley & Sons, Hoboken, NJ, 2009.

- [12] YouTube. Combinatorial optimization. <https://www.youtube.com/watch?v=1wjhtNuDC6E>, 2015. [Video].
- [13] dizhuangrou5770. Complexity classes relationship. <https://blog.csdn.net/dizhuangrou5770/article/details/101191905>, 2019. Accessed: September 29, 2025.
- [14] A. Gupta and S. Saini. An enhanced ant colony optimization algorithm for vehicle routing problem with time windows. In *2017 Ninth International Conference on Advanced Computing (ICoAC)*, pages 267–274. IEEE, December 2017.
- [15] Kusuma Soonpracha, Anan Mungwattana, Gerrit K. Janssens, and Tharinee Manisri. Heterogeneous vrp review and conceptual framework. In *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2014)*, pages 1052–1059, Hong Kong, 2014. Newswood Limited.
- [16] S. Elatar, K. Abouelmehdi, and M. E. Riffi. The vehicle routing problem in the last decade: variants, taxonomy and metaheuristics. *Procedia Computer Science*, 220:398–404, 2023.
- [17] G. D. Konstantakopoulos, S. P. Gayialis, and E. P. Kechagias. Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Operational Research*, 22(3):2033–2062, 2022.
- [18] Christoph Roser. Illustration of local optima and global optimum. AllAbout-Lean.com, 2025. Licensed under Creative Commons Attribution-ShareAlike 4.0 International License.
- [19] Mars Logistique. Présentation de mars logistique. <http://www.marslogistique.com>, 2025. Accessed: June 25, 2025.
- [20] Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to Time Series Analysis and Forecasting*. John Wiley & Sons, 2015.
- [21] Ratnadip Adhikari and RK Agrawal. *An introductory study on time series modeling and forecasting*. LAP Lambert Academic Publishing, 2013.
- [22] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [23] John H Cochrane. *Time series for macroeconomics and finance*. University of Chicago, 1997.
- [24] Keith W. Hipel and A. Ian McLeod. *Time Series Modelling of Water Resources and Environmental Systems*. Elsevier, 1994.
- [25] Elena R. Rosca. Stationary and non-stationary time series. *The USV Annals of Economics and Public Administration*.

- [26] Linus Sjösten. A comparative study of the kpss and adf tests in terms of size and power, 2022.
- [27] Aris A. Syntetos, John E. Boylan, and John D. Croston. On the categorization of demand patterns. *Journal of the Operational Research Society*, 56(5):495–503, 2005.
- [28] Aris A Syntetos and John E Boylan. The accuracy of intermittent demand estimates. *International journal of forecasting*, 21(2):303–314, 2005.
- [29] John D Croston. Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, 23(3):289–303, 1972.
- [30] Ruud H. Teunter, Aris A. Syntetos, and Mohamed Zied Babai. Intermittent demand: Linking forecasting to inventory obsolescence. *European Journal of Operational Research*, 214(3):606–615, 2011.
- [31] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
- [32] Qinglong Xu, Ning Wang, and Hongyan Shi. Review of croston’s method for intermittent demand forecasting. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 1456–1460. IEEE, IEEE, 2012.
- [33] Ruud Teunter and Babak Sani. On the bias of croston’s forecasting method. *European Journal of Operational Research*, 194(1):177–183, 2009.
- [34] Aris A Syntetos and John E Boylan. On the bias of intermittent demand estimates. *International Journal of Production Economics*, 71(1-3):457–466, 2001.
- [35] Nikolaos Kourentzes. On intermittent demand model optimisation and selection. *International Journal of Production Economics*, 156:180–190, 2014.
- [36] Lynton Shenstone and Rob J Hyndman. Stochastic models underlying croston’s method for intermittent demand forecasting. *Journal of Forecasting*, 24(6):389–402, 2005.
- [37] M Zied Babai, Yves Dallery, Sofiene Boubaker, and Ridha Kalai. A new method to forecast intermittent demand in the presence of inventory obsolescence. *International Journal of Production Economics*, 209:30–41, 2019.
- [38] John McCarthy. *Artificial intelligence: a modern approach*. Prentice Hall, 2007.
- [39] Stuart Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, 4th edition, 2021.
- [40] Seungchul Jeong and Jongpil Kim. Machine learning applications in manufacturing: current technology and future trends. *International Journal of Precision Engineering and Manufacturing*, 21(6):1027–1054, 2020.

- [41] Iqbal H Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):1–21, 2021.
- [42] Vladimir Nasteski. An overview of the supervised machine learning methods. *Horizons. B*, 4:51–62, 2017.
- [43] K Srihith and A Kumar. Random forest algorithm: a comprehensive review. *International Journal of Computer Applications*, 182(52):8–15, 2023.
- [44] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.
- [45] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
- [46] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [47] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.
- [48] Casper Solheim Bojer and Jens Peder Meldgaard. Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, 37(2):587–603, 2021.
- [49] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364, 2022.
- [50] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
- [51] Rory Mitchell and Eibe Frank. Accelerating the xgboost algorithm using gpu computing. *PeerJ Computer Science*, 3:e127, 2017.
- [52] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, volume 30, 2017.
- [53] Tim Januschowski, Jan Gasthaus, Yuyang Wang, David Salinas, Valentin Flunkert, Michael Bohlke-Schneider, and Laurent Callot. Forecasting with gradient boosted trees: augmentation, tuning, and cross-validation strategies: Winning solution to the m5 uncertainty competition. *International Journal of Forecasting*, 38(4):1426–1433, 2022.
- [54] XGBlog. Xgboost is all you need part 3: Gradient boosting deep dive, 2024. Online blog post.

- [55] Xue Ying. An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168, page 022022. IOP Publishing, February 2019.
- [56] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(1):281–305, 2012.
- [57] B. S. Alanazi. A comparative study of traditional statistical methods and machine learning techniques for improved predictive models. *International Journal of Analysis and Applications*, 23:18–18, 2025.
- [58] Shanshan Wang and Wanpracha Art Chaovalitwongse. Evaluating and comparing forecasting models. In *Wiley Encyclopedia of Operations Research and Management Science*. Wiley, 2011.
- [59] Ding-Zhu Du, Panos M Pardalos, Xiaodong Hu, and Weili Wu. *Introduction to combinatorial optimization*, volume 196. Springer Nature, 2022.
- [60] Martin Grötschel and László Lovász. Combinatorial optimization. In *Handbook of Combinatorics*, volume 2, pages 1541–1597. Elsevier, 1995.
- [61] Guillaume Sagnol and Max Klimm. Introduction to linear and combinatorial optimization (adm i). Lecture notes, Zuse Institute Berlin (ZIB), 2021. Available online: https://www.zib.de/userpage/sagnol/uploads/adm/1_intro.pdf.
- [62] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, 4 edition, 2022.
- [63] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.
- [64] Nicos Christofides. The vehicle routing problem. *Revue française d’automatique, informatique, recherche opérationnelle. Recherche opérationnelle*, 10(V1):55–70, 1976.
- [65] S. N. Kumar and R. Panneerselvam. A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, 4(03):66–74, 2012.
- [66] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.
- [67] Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, 59(3):345–358, 1992.
- [68] Roberto Baldacci, Aristide Mingozzi, and Roberto Roberti. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6, 2012.
- [69] Barrie M Baker and Mohammed A Ayeche. A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5):787–800, 2003.

- [70] Andrew Lim and Fang Wang. Multi-depot vehicle routing problem: A one-stage approach. *IEEE Transactions on Automation Science and Engineering*, 2(4):397–402, 2005.
- [71] Feng Li, Bruce Golden, and Edward Wasil. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & Operations Research*, 34(10):2918–2930, 2007.
- [72] Claudia Archetti, M Grazia Speranza, and Alain Hertz. A branch-and-price algorithm for the split delivery vehicle routing problem. *Transportation Science*, 42(1):22–31, 2008.
- [73] The Jin Ai and Voratas Kachitvichyanukul. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 36(5):1693–1702, 2009.
- [74] Yiming Huang, Lu Zhao, Tom Van Woensel, and Jonathan P. Gross. Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, 95:169–195, 2017.
- [75] Harilaos N. Psaraftis. Dynamic vehicle routing: Status and prospects. *Annals of Operations Research*, 61(1):143–164, 1995.
- [76] Teodor Gabriel Crainic, Simona Mancini, Guido Perboli, and Roberto Tadei. Multi-start heuristics for the two-echelon vehicle routing problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 179–190, Berlin, Heidelberg, 2011. Springer.
- [77] Shahryar Gholipour, Ahmad Ashoftehfar, and Hassan Mina. Green supply chain network design considering inventory-location-routing problem: a fuzzy solution approach. *International Journal of Logistics Systems and Management*, 35(4):436–452, 2020.
- [78] José C. Brandão and Alan Mercer. The multi-trip vehicle routing problem. *Journal of the Operational Research Society*, 49(8):799–805, 1998.
- [79] Claudia Archetti, Ola Jabali, and M Grazia Speranza. Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 61:122–134, 2015.
- [80] Vinícius R. Máximo, Jean-François Cordeau, and Mariá C. V. Nascimento. An adaptive iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *arXiv preprint*, 2021.
- [81] G. Chen, J. Gao, and D. Chen. Research on vehicle routing problem with time windows based on improved genetic algorithm and ant colony algorithm. *Electronics*, 14(4):647, 2025.

- [82] Nariman Torkzaban, Anousheh Gholami, John S. Baras, and Bruce L. Golden. AASA: A Priori Adaptive Splitting Algorithm for the Split Delivery Vehicle Routing Problem. *Algorithms*, 17(9):396, 2024. Open access, published 6 September 2024.
- [83] Eliana Mirledy Toro-Ocampo, Antonio Hernando Escobar Zuluaga, and Mauricio Granada Echeverri. Literature review on the vehicle routing problem in the green transportation context. *Revista Luna Azul*, 42:362–387, 2016.
- [84] Diego Gasset, José Rodríguez, and Marta Sánchez. Route optimization for open vehicle routing problem (ovrp): A mathematical and solution approach. *Applied Sciences*, 14(16):6931, 2024.
- [85] Rui Xu, Shumin Li, and Jiayan Wu. Multi-trip vehicle routing problem with time windows and resource synchronization on heterogeneous facilities. *Systems*, 11(8):412, 2023.
- [86] Jan Karel Lenstra and Alexander H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.
- [87] F. Liu, C. Lu, L. Gui, Q. Zhang, X. Tong, and M. Yuan. Heuristics for vehicle routing problem: A survey and recent advances. *arXiv preprint arXiv:2303.04147*, 2023.