IEM Bachelor Thesis

2025

Developing an algorithm for optimizing decision-making in driver detours within transportation logistics.

Nijat Hasratli

S2211300

Colophon

INSTITUTION

University of Twente

FACULTY
Behavioral Management and Social Sciences

DOCUMENT

Developing an algorithm for optimizing decision-making in driver detours within transportation logistics

AUTHOR

Nijat Hasratli

TOOLS

Microsoft Word, Excel, Python

SUPERVISORS

Jean Paul Sebastian Piest (first supervisor)

Engin Topan (second supervisor)

Arjan Berndsen (company supervisor)

EMAIL

n.hasratli@student.utwente.nl

WEBSITE

www.utwente.nl

Management Summary

This project is about the development of an algorithm description, which will lead to decision-making optimization in driver detours and overnight stays in Bouwvervoer's logistics operations. The innovative family-owned logistics company Bouwvervoer is collaborating with Softec AI Solutions to use and improve their Advanced Planning System, Norma LIVE, which combines Artificial Intelligence and scientific algorithms to enhance operational efficiency. The project is purposed to switch from manual route planning to intelligent route planning to achieve cost efficiency, compliance of driver agreements and rest regulations. Bouwvervoer expects progress in terms of reduction in operational costs, and improvements in driver satisfaction if this algorithm is implemented. The company, Bouwvervoer is working closely with Softec AI for switching to more efficient system without human errors. This thesis project is focussing on only small part of this change related to driver decision-making. The problem is lack of algorithm or decision-making system for drivers to decide on what to do at the end of the day. In order to understand the problem deeply the key factors effecting decision making is analysed in this thesis. After outlining the key factors, formulas were created and implemented into the python code to compare the differences between options and picking the optimal one. The companies' requirements about costs and driver well-being is considered in the code scripts. The MPSM steps are followed in detail to achieve working solutions. According to the first steps of MPSM the algorithm design was constructed and was followed with python script with detailed explanation of each step. The actual result of the thesis is a defined algorithm logic with working code that compares the costs of staying in the truck and going to the base with using mainly fuel prices according to the change in the kilometres and additional sleepover costs. The developed algorithm provides a structured approach to driver detour decision-making, optimizing cost efficiency while maintaining compliance with driver agreements. It successfully evaluates fuel costs, sleepover expenses, and route constraints to determine the most cost-effective option. Testing showed that the algorithm performs well in minimizing unnecessary detours and applying constraints correctly. However, its outputs are significantly influenced by fixed sleepover costs and the assumption of a single base location, limiting its flexibility in real-world applications. The jupyter notebook is provided with detailed explanations of python codes. The full python codes can be found at the Appendix.

Table of Contents

List of tables

List of Abbreviations

Definition	Abbreviation
Managerial Problem-Solving Methods	MPSM
Constraint Programming	CP
Decision Tree	DT
Rule Based System	RBS
Linear Programming	LP
Genetic Algorithm	GA
Literature Review	LR
Prisma	Preferred Reporting Items for Systematic Reviews and Meta-Analyses

List of figures

Chapter 1: Introduction

The success of logistics operations depends highly on efficient planning and decision-making. Minor improvements in these processes can significantly impact costs and employee satisfaction. This chapter helps to understand the challenges faced by Bouwvervoer, a logistics company transitioning from manual to automated planning through an AI-powered system, Norma LIVE. Section 1.1 introduces the project background, followed by Section 1.2 which presents the problem statement. Section 1.3 explains the Managerial Problem-Solving Method (MPSM) and its application in the thesis. Section 1.4 focuses on problem identification, including the problem cluster and the gap between current and desired conditions. The research objectives are outlined in Section 1.5 followed by Section 1.6 which presents the research questions in relation to MPSM steps. Finally, Section 1.7 summarizes the chapter and outlines the thesis structure.

1.1 Background

One of the crucial parts of construction industry is logistics operations, where minor improvements can directly affect costs and employee satisfaction. Bouwvervoer is a family-owned logistics company that successfully handles the transportation of construction materials with around 100 specialists (*Bouwvervoer BV*, 2025). Planners play a crucial role within operations; meeting delivery times and planning optimal routes. These steps are very important in a prosperous logistics business. To enhance the efficiency of its operations, Bouwvervoer has partnered with Softec AI to develop an advanced planning system in which route planning becomes more automated and optimized with the help of an AI-based system called Norma LIVE. This collaboration aims to switch from manual planning to automated route planning in which planners' constant attention is not demanded.

According to the Harrison and van Hoek (2002) the most important factor for successful logistic system is well defined decision-making structure. The current focus is to develop a system that optimizes decision-making processes. This project concentrates on specific aspect of this newly developed system, namely the creation of an algorithm that helps to decide whether it is more cost-effective for drivers to detour home/base for the night or stay overnight in the truck. These decisions depend on factors such as fuel cost, accommodation expenses, driver preferences, and regulatory compliance.

1.2 Problem Statement

The main challenge faced by Bouwvervoer is the lack of a data-driven decision-making algorithm. Currently, at the end of each workday, drivers must decide whether to stay in their trucks overnight or return to the base. However, this decision involves multiple factors, including fuel costs, delivery deadlines, rest requirements, and company policies, making it difficult for drivers to consistently choose the most optimal option. As a result, suboptimal decisions may lead to increased operational costs and driver dissatisfaction. To address this issue, Bouwvervoer aims to implement an algorithm-based decision-making system that can analyze relevant factors and provide optimal routing recommendations for drivers. This research will explore algorithmic models that enhance efficiency, reduce costs, and improve driver well-being, ensuring a more data-driven and consistent approach to decision-making.

1.3 Methodology

This sup-chapter is focused on explaining the chosen methodology to achieve results. For this thesis project the used methodology will be well known Managerial Problem-Solving Methods (MPSM). Chapter demonstrates the definition of different steps in MPSM and how they will be implemented in this thesis within compact table. Table 1 bellow shows each step of the MPSM with its definition and application in this project.

The MPSM is an organized approach broadly used in organizations to address complex decision-making challenges. (Heerkens & Van Winden, 2021) MPSM will be used in this thesis to systematically solve problems and develop an effective solution. This methodology is used to break down complex issues into manageable parts, enabling to analyse the root causes and develop algorithm.

The MPSM process consist of seven steps:

MPSM Steps	Definition	Application in Thesis
Problem Identification	Defining the core problem and understanding its impact on operations.	General introduction about the nature of the problem and its effects. Problem Cluster is made to identify core problem. The difference between norm and reality is discussed
Solution Planning	Creating a roadmap for solving the problem, setting objectives and constraints.	Key objectives are decided and divided into different categories. And guidelines for next steps are planned
Problem Analysis	Analysing data to understand the contributing factors and the scale of the problem.	Conducting literature reviews Decided on which factors should be considered to satisfy stakeholders
Solution Generation	Proposing potential solutions based on the problem analysis.	Shaping the company data to be used in algorithm building. How the logic of previously mentioned algorithmic models be used in

		solution generation
Solution Choice	Selecting the best solution from the available options.	Criteria is created to compare possible solutions and pick the most suitable and successful option
Solution Implementation	Implementing the chosen solution in the operational environment.	Making an outline of algorithm with inputs, calculations and outputs. Explaining steps of codes and logic behind the calculations
Solution Evaluation	Assessing the effectiveness of the solution and refining it as needed.	Evaluation of the final solutions and validations methods. Discussing iterations during the thesis project.

Table 1 MPSM Steps

1.4 Problem Identification

Switching between systems is not easy, and Norma Live should include different algorithms within their system to consider many other factors while planning. As mentioned in the previous paragraph, more than 100 drivers are working for Bouwvervoer, and almost every weekday, they are on delivery. Drivers must decide where to spend the night at the end of each day. They have options to either sleep in the truck or go to the closest base. In the current program, drivers decide to stay in the truck or go back to base for themselves; these decisions are based only on driver preference, where the cost factors and efficiency for the next delivery are not calculated. The increasing cost of sleepovers, inefficiency due to detours, and driver dissatisfaction are reasons to consider this matter a problem. Thus, the main task is to create an algorithm description that determines whether a driver is allowed to detour via their base and sleep at home, or if, given the conditions, it is more cost-effective to spend the night in the vehicle.

Problem Cluster and Core Problem

Problem cluster is a tool to identify the core problem of the main action problem. Solving core problems can automatically solve related problems if they are identified correctly. Figure 1 shows problem cluster to identify core problem.

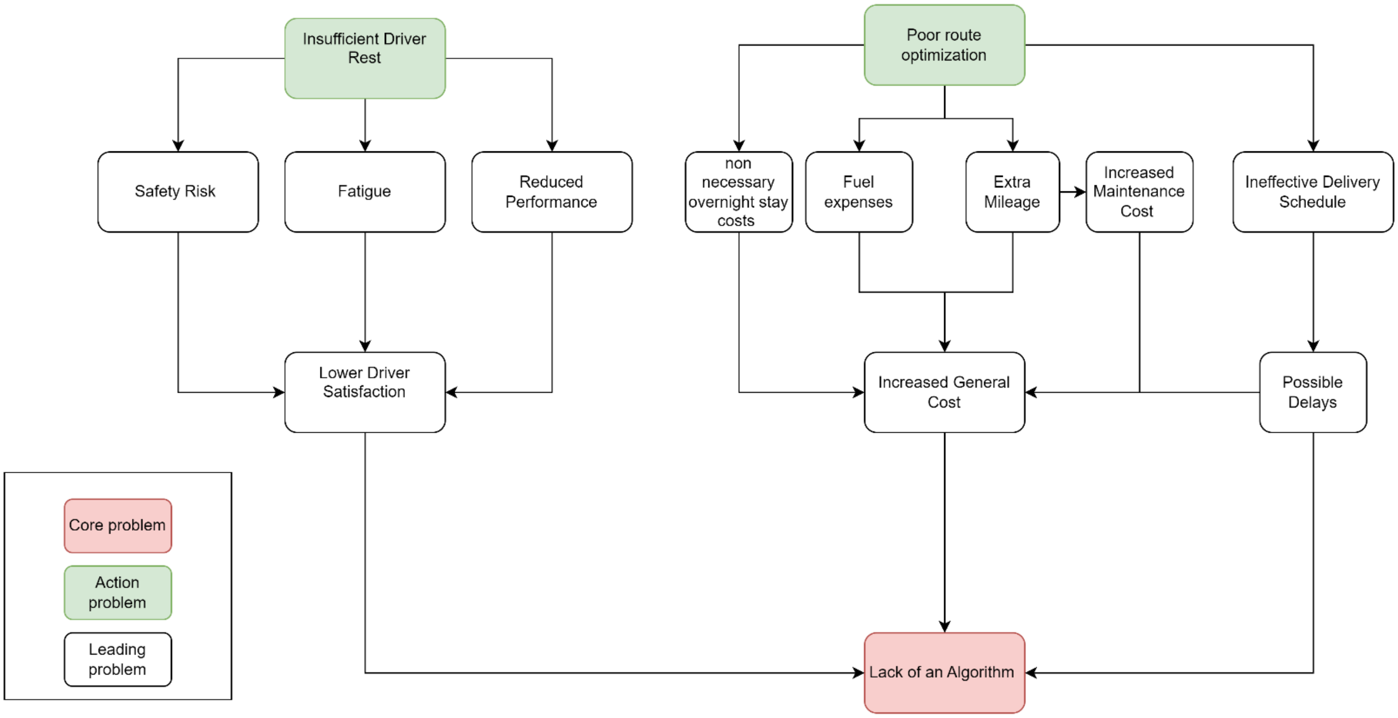


Figure 1 Problem Cluster

The problem cluster presents the cause-effect relationship between the core problem and its contributing issues. The root of the issue is the lack of algorithm, which directly leads action problems that impacts overall logistics efficiency.

At the root of the issue is the lack of an algorithm for efficient driver planning and routing. This core problem directly contributes to two significant action problems that impact overall transportation efficiency:

1. Insufficient Driver Rest – Poor scheduling and planning prevent drivers from getting adequate rest, leading to fatigue, reduced performance, and potential safety risks.
2. Poor Route Optimization – Inefficiencies in routing lead to delays, detours, and ineffective delivery schedules, reducing overall service efficiency. Without an optimized planning system, unnecessary mileage, fuel expenses, and overnight stays increase operational costs.

Desired solution of creating the algorithm for the operations of Bouwvervoer would assist the company to overcome the operational inefficiencies.

Norm and Reality

In order to understand the gap between norm and current practices, the norm and reality should be identified. Possible norm for this project is to have some kind of system which makes optimal decisions for drivers. More specifically norm in this case represent having an algorithm for decision-making that maximizes efficiency, minimizes costs, and ensures that drivers can return home when it is economically and practically viable. However, the current reality shows that the existing planning process may not meet these standards. The current system involves manual planning processes that are time-consuming and prone to human error, relying heavily on planners' experience and intuition. This can lead to inconsistencies, suboptimal decisions, and increased operational costs due to inefficiencies. Moreover, the manual nature of the existing planning process can lead to a lack of transparency and consistency, making it challenging to ensure consistent compliance with regulations and company policies. Changes in cost and driver satisfaction can be considered to measure this gap. Implementing a decision-making algorithm is crucial since more efficient planning will reduce costs and increase driver satisfaction. By systematically identifying and addressing the discrepancies between the norm and reality, the project aims to enhance operational efficiency, reduce costs, and improve driver satisfaction through more effective and data-driven planning processes. This approach will align Bouwvervoer's practices with norm and set a new benchmark for excellence in logistics planning and execution.

1.5 Research Objectives

This thesis aims to develop an algorithm description to build an actual algorithm within Norma LIVE to provide cost-effective decisions on whether drivers should return home or stay overnight in their trucks. The algorithm description will account for cost optimization, driver well-being, and regulatory requirements, such as ensuring a minimum rest of nine hours.

By automating the decision-making process, the algorithm will replace the manual planning system, which can reduce the burden on human planners and eliminate human mistakes while providing transparent and consistent planning. In summary, the objectives are:

1.6 Research Questions

To achieve the objectives, the thesis will address the following research questions, and a literature review will be conducted on the second question. Since the chosen methodology is MPSM, the research questions are related to its steps. The table 2 demonstrates how research questions help to do each step of the MPSM.

MPSM Steps	Relation of Research question to MPSM steps	Research Question
Problem Identification	Identifying key variables that influence driver decisions.	Sub-Research Question 2
Solution Planning	Structuring the approach for designing the algorithm to address these inefficiencies.	Main Research Question
Problem	Investigating cost factors, operational	Sub-Research Question 2

Analysis	constraints, and driver preferences that impact these decisions. Understanding existing algorithmic models and decision-making frameworks	Sub-Research Question 1
Solution Generation	Exploring different algorithmic methods to integrate data inputs and optimize decisions.	Sub-Research Question 1
Solution Choice	Selecting the most suitable algorithm for implementation in logistics decision-making.	Sub-Research Question 1
Solution Implementation	Directly focuses on making an algorithm.	Main Research Question
Solution Evaluation	Assessing whether the developed algorithm achieves the balance between cost-effectiveness and driver well-being.	Main Research Question

Table 2 Relations between MPSM and Research Questions

Main Research Question:

How can a logistics company's decision-making algorithm be designed to optimize driver decisions, balancing operational cost-effectiveness and driver well-being?

Sub-Research Questions related to MPSM steps:

1. What algorithmic models can be developed to optimize decision-making in terms of cost-effectiveness, driver well-being, and compliance? The goal is to learn about different methods and propose algorithms that integrate data inputs from various sources and produce optimal decisions.
2. What factors influence a driver's decision to detour home or stay in the truck? This question will help identify the key variables that impact these decisions, such as cost factors and constraints.

In table 3 the main step of the research is outlined with what to do in each step.

Research Phase	Description
Research Approach	The MPSM framework is used to systematically analyze the problem and generate solutions
Data Collection Methods	Literature Review, Data Analysis and gathering insights from company employees
Solution Development	Algorithm Design, testing algorithm with sample data and comparing results.
Evaluation and Conclusion	Discussing the effectiveness of algorithm and providing recommendations for future implementations.

Table 3 Research Phases

1.7 Structure of Thesis

Chapter 1 has outlined the essential aspects of the research, including the challenges Bouwvervoer faces, the inefficiencies in manual decision-making processes, and the necessity of an algorithm for improving operations. By presenting a clear problem statement, objectives, and structured research questions, this chapter helps to go further for a detailed exploration of logistics optimization in the following chapters. The subsequent chapters will build upon this introduction, providing context, literature insights, to achieve the stated goals. Table 4 outlines the general structure of the thesis.

This thesis is structured as follows:

Chapter 1: Introduction	Discusses research background, problem statement and objectives. Detailed breakdown of each methodology step to achieve the result
Chapter 2: Context Chapter	Shows available materials and information about stakeholders
Chapter 3: Literature Review	Focuses on answering research questions and provides literature
Chapter 4: MPSM	Detailed breakdown of each methodology step to achieve the result
Chapter 5: Conclusion	Summarise key findings and limitation with future research directions,

Table 4 Structure of Thesis

Chapter 2: Context Chapter

Understanding the operational context is vital for developing an effective decision-making algorithm. This chapter explores the key stakeholders involved in the project and the data provided by Bouwvervoer to support the algorithm's design. Stakeholder analysis shows the different priorities and concerns of the logistics service provider, drivers, regulatory authorities, planners, and Softec Technologies. Additionally, an overview of available data sets and tools are discussed in this chapter. There are 2 main sub-chapter, section 2.1 outlines stakeholders and their roles. Following by section 2.2 focusing on the required and received data.

2.1 Stakeholders

There are different roles involved in this project; after a period of engagement with the company and its employees, this analysis of stakeholders was created. This stakeholder analysis is conducted with the information from the websites of the companies and discussions with the company employees.

Bouwvervoer (Logistics Service Provider)

The central entity of this logistics service is Bouwvervoer. From a management perspective, the main concern is optimizing their operations to reduce costs and improve efficiency. Bouwvervoer's concerns include minimizing operational expenses, ensuring timely deliveries, complying with driver agreements, and meeting regulatory requirements. Management of Bouwvervoer also want to maintain the well-being of their drivers to ensure their safety and job satisfaction.

Softec Technologies Group (AI solutions Provider)

Softec provides AI solutions, including the Advanced Planning System Norma LIVE, to Bouwvervoer. Softec's AI Center is an innovative scientific research department developing propriety algorithms, routing solutions, and machine learning capabilities for logistics, transportation, and last-mile delivery companies. They aim to enhance logistics operations' efficiency, consistency, and transparency. Using Norma LIVE, concerns revolve around developing a practical algorithm that aligns with Bouwvervoer's goals. They need to ensure that the software they provide addresses the specific challenges Bouwvervoer faces and delivers a solution that optimizes the decision-making process.

Drivers (Logistics Personnel of Bouwvervoer)

Drivers play a critical role in logistics operations. Their primary concern is their well-being and job satisfaction, including rest time and personal preferences regarding detours and overnight stays. Drivers are concerned about factors that impact their quality of life and work balance, such as rest time, the number of overnight stays, and how the algorithm's decisions affect their personal lives and comfort while on the road.

Regulatory Authorities (Transportation Regulators)

Regulatory authorities oversee and regulate the transportation and logistics industry to ensure safety and compliance with laws and environmental standards. They are concerned with the safety of drivers and other road users and want to ensure that logistics companies like Bouwvervoer comply with regulations related to rest time, driving hours, and environmental standards.

Planners (*Logistics Personnel of Bouwvervoer)*

Since the company is using a manual planning system, planners have an important role in making sure deliveries are one-time. They have to work on planning daily. They will have less work overload when the company moves to an AI planning system. This can negatively affect them by making them feel less valuable to the company.

Figure 2 Stakeholder Diagram

2.2 Data

In order to create algorithm description, inputs and calculations should be made. For this reason, data were requested from the company. In bellow table required data type, description of data and purpose of usage was stated.

Data Type	Description	Purpose/Usage
Historical Route Data	Trip start/end times, routes taken, km covered	Analyze routes, calculate trip duration/distance, identify inefficiencies
Driver Schedules and Preferences	Daily schedules, overnight stay records, preferences	Respect schedules/preferences, incorporate comfort/satisfaction
Cost Data	Fuel costs, accommodation costs	Calculate financial impact, optimize for cost-effectiveness
Operational Data	Delivery windows, penalties, recorded delays	Ensure delivery windows are met, understand delay causes
Regulatory and Compliance Data	Latest legal requirements and compliance records	Integrate regulatory compliance, avoid violating rest period regulations

Table 5 List of Requested Data

Data Received

1. **Driver Preferences** (pl-A-nn-I-ng Sleepovers Driver preferences 2024.xlsx):

Sleepovers	Remarks
4	All week sleepovers
0	
4	All week sleepovers
4	All week sleepovers
0-1-2	in consultation
0	
	preferably no Makz
0-1-2	in consultation
4	All week, a night at home no problem
0	
0-1-2-3-4	Sleepovers no problem
0-1-2	in consultation
0-1-2	in consultation
0	

Figure 3 Excel screenshot | Driver preferences.

Columns to focus on: driver name, shipment related remarks (not included for the ethical purposes), sleepover preferences (number of sleepovers allowed), and resource codes.This data will help to determine how many sleepovers a driver is allowed and whether any restrictions apply. 0-1-2 shows that the number of sleepovers is still in the discussion. In this project the maximum number was taken into account. If data shows 0-1-2, two sleepovers were considered for the driver, with the same logic 0-1-2-3-4, four sleepovers were considered.

2. **Trip and Operational Data** (pl-A-nn-I-ng Order Dataset twee weken 2024.xlsx):

Kind	Shipment	Status	Addressname	City	Year	Period	Actionkind	Actions	Arrivaldate	Arrivaltime	Departuredate	Departuretime	Duration (minutes)	Duration (calculated)	Duration representative
Route							Laden								
Stop	30000		Dycore BV Breda	BREDA	2024	7	LADEN		7/5/2024	7:58:03 AM	7/5/2024	8:53:04 AM	55	12:55:01 AM	Representatief
30000	804730	Uitgevoerd	Dycore BV Breda	BREDA	0		LADEN		7/5/2024	7:58:03 AM	7/5/2024	8:53:04 AM	55	12:55:01 AM	

Figure 4 Excel screenshot | Trip Data

Important columns: trip distance, status, loading/unloading times, city names, and vehicle data. This data will allow us to calculate trip distance, assess if a driver can return to base, and check the shipment status for operational planning. Picture below shows how the data looks like, some columns are altered and deleted to not reveal private data such as customer name and trip code.

3. **Fuel Consumption Data** (pl-A-nn-I-ng Order Dataset twee weken 2024 Fuelconsumption.xlsx):

Ø-consump. (l/100km)	Ø-speed (km/h)	Total consumption in l
14.65	63.50	780.11
15.35	55.67	473.49
20.55	68.34	18.65
20.85	62.52	692
21.42	64.65	477
21.43	58.68	617
21.93	57.46	679
22.20	63.91	847.5
22.84	57.53	911.5
23.19	66.77	1100
23.24	61.96	1252.5

Figure 5 Excel screenshot | Fuel Consumption

Relevant columns: vehicle number, fuel efficiency (liters per 100 km), and total fuel consumption. (License plates are not included in the picture due to ethical reasons)

This data helps calculate the cost of detours based on fuel consumption rates.

4. **Sleepovers** (pl-A-nn-I-ng Order Dataset twee weken 2024 Overnachtingen.xlsx):

This excel file demonstrates when drivers used sleepovers from first of July till 12th. Driver names are not included in the screenshot for safety reasons.

1-Jul									12-Jul
ma	di	wo	do	vr	ma	di	wo	do	vr
x	x	x	x		x	x	x	x	
x	x		x			x	x	x	
		x							
x	x				x			x	
			x		x	x			
x									
						x			
x	x	x			x	x	x	x	
					x	x	x		
x	x	x	x		x	x	x	x	
x	x	x	x		x	x	x		
x	x	x	x		x	x	x	x	
x	x	x	x		x	x	x	x	
			x			x	x		
	x	x	x		x	x	x		
		x	x		x	x			
	x	x	x				x		
						x			
						x			

Figure 6 Excel screenshot | Sleepovers

2.3 Conclusion

Chapter 2 has provided a detailed analysis of the stakeholders and their roles, alongside a comprehensive review of the data necessary for algorithm development. By understanding the concerns of each stakeholder and utilizing the diverse data sets, the project can address challenges effectively. These insights serve as a crucial basis for the next chapters, and data sets will be used in the next chapters for calculations and algorithm

Chapter 3: Research Questions and Literature Review

This chapter will conduct a literature review to fill the knowledge gap; as discussed in Research Questions, there are two sub-research questions. The literature review will follow a simple version of the PRISMA framework, including search strings, databases, inclusion and exclusion criteria, a PRISMA flow diagram. However, not all the answers to research questions will come through a literature review; some will require data analysis, discussions, and existing information from the company. Chapter starts with explaining literature review methodology in 3.1. This chapter is focused on getting answers for the research questions 1 and 2 in 3.2 and 3.3 respectively.

3.1 Literature Review Methodology

PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) is a framework for conducting and reporting systematic reviews. It ensures transparency in the research process by systematically identifying, screening and picking relevant studies. The key steps of the PRISMA are defining inclusion and exclusion criteria, creating search strategy with multiple databases, removing duplicates, screening articles and selecting final studies. PRISMA flow diagram visually shows this process with detailed number of studies filtered, excluded and included in the review. Bellow in this chapter Table 6 demonstrates inclusion and exclusion criteria followed by Table 7 with search strings databases. Lastly combined flow diagram is provided for given search strings.

Inclusion and Exclusion Criteria

Criteria	Inclusion Criteria	Exclusion Criteria
Peer Review	Articles published in peer-reviewed journals	Articles not peer-reviewed
Focus	Studies focusing on the algorithm making or decision-making	Studies not addressing the decision-making models
Relevance	Research addressing the integration of these algorithms into planning systems	Research not related to planning decisions or planning systems
Language	Papers published in English	Non-English publications
Date	Publications after 2005	Studies published before 2005

Table 6 Inclusion and Exclusion Criteria for LR

Search Strings

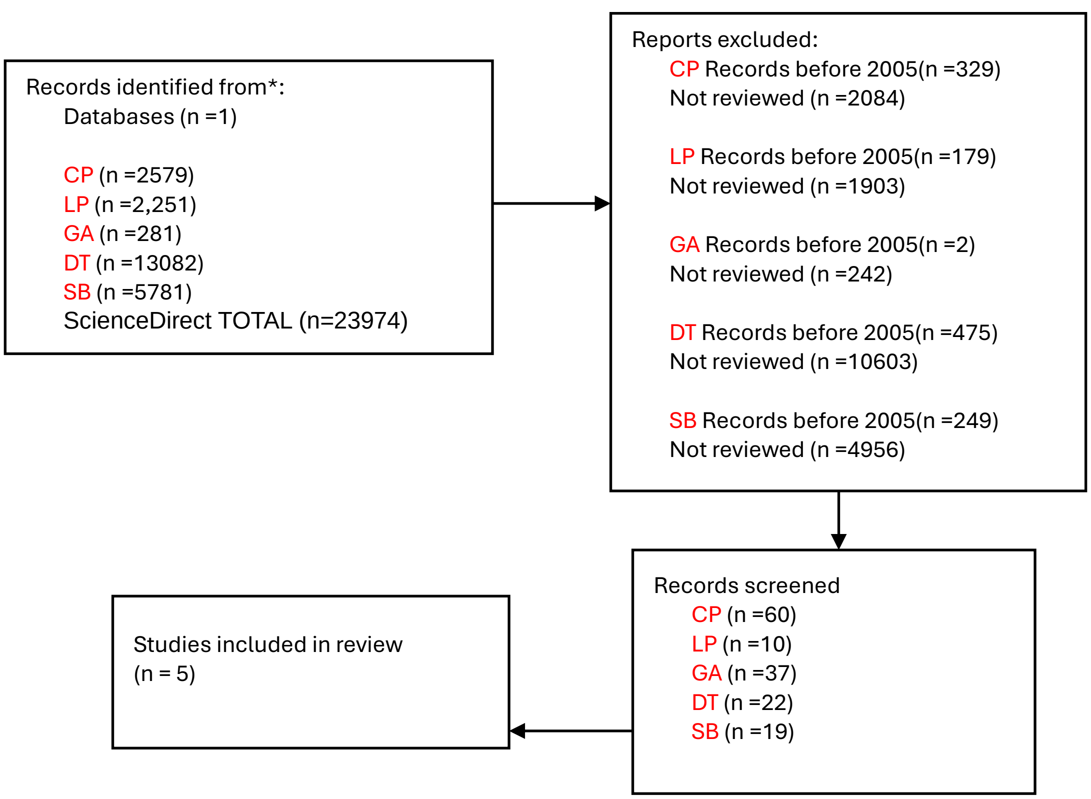
Models	Search Stings
Constraint Programming	("constraint programming" AND "artificial intelligence") OR "logistics optimization"
Linear Programming	"linear programming" AND “logistics "AND “cost minimization"
Genetic Algorithms	"genetic algorithm" AND "optimization efficiency" AND "logistics"
Decision Trees	"decision trees" AND "logistics" AND "decision-making"
Simulation-Based	"simulation-based" AND "algorithm" AND "logistics"
Machine Learning-Based Models	"machine learning in logistics"

Table 7 Search strings for LR

The main academic databases to be used are:

- Scopus
- ScienceDirect

Combined Flow Diagram



3.2 Research Question 1

Research Question:

What algorithmic models can be developed to optimize decision-making in terms of cost-effectiveness, driver well-being, and compliance? The goal is to learn about different methods and propose algorithms that integrate data inputs from various sources and produce optimal decisions.

This literature review aims to examine a range of algorithmic models that can be applied to optimize decision-making. Each model will be analyzed for its potential benefits and limitations. Understanding these models will allow us to pick the most suitable approach and fill the knowledge gap for the second sub-research question

Introduction to Decision Models

Decision-making is a critical process that directly influences operational efficiency, cost management, and regulatory compliance in logistics and transportation. Companies in these sectors must navigate a complex mix of variables, such as fuel prices, delivery deadlines, etc. The growing complexity of these variables makes it hard to rely solely on human planners for optimal decision-making. Algorithmic decision models have become an essential tool in logistics to address this problem.

Decision models are structured algorithms that can guide decision-makers in picking the best possible outcome from different alternatives. Depending on the model, it can leverage data inputs from multiple sources to analyze, predict, and optimize the decision. (Sebata, 2024) In logistics, these models are mainly used to streamline route planning, minimize costs, and enhance satisfaction. Automating decisions can decrease the load on human planners, and consistent, transparent decision-making processes can be made.

According to Lombardi et al. (2016), there are several different models for building decision-making algorithms. In this literature review, search strings will be prepared for each model, and research will be conducted on each model. After getting enough information, the right model to build this algorithm will be chosen based on the evaluation criteria that was created.

Constraint Programming (CP)

Constraint Programming is a strong technique for solving combinatorial problems by stating constraints that must be met in solution. The basic logic of this model is to define constraints for decision variables and look for feasible solutions that satisfy the constraints. In logistics operations, CP is used for scheduling, resource allocation, vehicle routing, etc. Regardless of this model being resource-intensive for data, it excels in scenarios with multiple rules and complex relationships (Rossi et al., 2008).

Linear Programming (LP)

Linear programming is a mathematical optimization technique that finds the best outcome for linear objective functions, such as minimum cost or maximum profit. Mainly used for route and scheduling optimizations and cost minimization. If constraints are defined correctly LP finds the most efficient solution, however, cannot handle non-linear relationships. (Clautiaux & Ljubić, 2024).

Genetic Algorithms (GAs)

GAs are search heuristic based on the principles of natural selection and genetics. The main characteristic of GAs is to evolve potential solutions through iteration by mimicking processes like crossover and mutation. Finding an optimal path by iteratively improving on previous solutions can be a good example of use of GAs in fleet optimization. (Alexakis et al., 2024).

Decision Trees

Decision Trees are hierarchical models that lay out decision paths based on conditions. There are internal nodes, branches, and leaf nodes. Fundamentally, internal(root) nodes represent decisions, branches appear as outcomes, and leaf nodes are decisions. Besides determining optimal routes and cost management strategies, this model is usually helpful for visualizing the possible outcomes in logistics operations. It can become complex if there are many variables and branches. (Bhowmik et al., 2024).

Simulation-Based Approaches

This approach involves creating digital replicas of the real-word logistics systems to study its behavior and predict different scenarios. The effects of different decisions and variables can be analyzed by running simulations. Modeling simulations with key variables and interactions is first step to do for scenario and data analysis. For example, can predict the impact of warehouse layout changes on order picking efficiency. Building detailed simulations can take significant time and resources. (Rabia & Bellabdaoui, 2022).

Rule-Based Systems

Rule-based systems work on predetermined rules that dictate how the system should behave. Basically, a simple list of if-then conditions is used to determine the appropriate action or decision. As an example, a rule-based system can be used for reordering stock after inventory levels fall below certain thresholds each time to automate order processing.

Evaluation Criteria to Pick a Model

In order to connect these models to thesis topic and pick the related models evaluation criteria was made based on three main categories of algorithm logic.

Cost Effectiveness:

In logistics, minimizing cost is a primary goal since it directly impacts a company's profitability. Reducing fuel, vehicle maintenance, and routing expenses are the main parts of minimizing costs. Also, because it aligns with the primary goal of this thesis, cost-effectiveness will be an important criterion. How each model helps to optimize cost will be considered while evaluating models.

Compliance with Regulations:

It is essential to adhere to regulations, such as the EU rules on driver hours, mandatory rest periods, and safety standards for legal compliance and avoiding fines. Thus, while evaluating the model, their ability to consistently produce solutions that comply with the legal and operational constraints.

Driver Well-Being:

Ensuring drivers' well-being is important for maintaining high performance and improving job satisfaction. This paper will examine how each model can consider driver preferences, rest requirements, and well-being in its decision-making process.

Model Analysis

When considering different logistics decision-making models, each has pros and cons regarding cost, regulation compliance, and driver well-being. CP is suitable for keeping costs low while handling tricky rules. It is excellent for meeting strict regulations and ensuring drivers get their rest, though setting it up can be challenging. LP is perfect for straightforward cost-cutting and legal rules, but it struggles when things get non-linear. GAs are like multitaskers—they are flexible and do a solid job finding almost the best solutions for problems with multiple goals. However, they can be slow, need much computing power, and sometimes bend the rules. Decision Trees offer clear choices but cannot handle many factors or shifting conditions. Then there are simulation-based approaches, which are great for testing different scenarios but take time and need accurate data to be helpful. Machine Learning models excel at spotting trends and adapting to new patterns, but they require a lot of data and aren't the best at following strict rules. Rule-based systems are simple and easy to use for keeping costs down and staying compliant, but they aren't so good when things change. Table 8 outlines pros and cons of the different models.

Models	Advantages	Disadvantages
Constraint Programming	• Handles Complex Constraints • Flexible and Scalable	○ High Complexity ○ Resource intensive
Linear Programming	• Efficient for Linear Problems	○ Limited to Linear problems ○ Complex Setup
Genetic Algorithms	• Adaptable • Wide Solution Space	○ Difficult to Interpret
Decision Trees	• Easy to interpret with categorical and numeric data	○ Becomes complex with datasets
Simulation-Based Approaches	• Visualize • Scenario Analysis	○ Time Consuming
Machine Learning	• Adapts to Changes • Handles Complex Data	○ Data-Intensive ○ High Complexity
Rule-Based Systems	• Simple to Implement • Low Cost	○ No flexibility ○ Difficult to manage with many rules

Table 8 Summary of algorithmic models

3.3 Research Question 2

What factors influence the decisions between a driver detouring home or staying in the truck?

Driver Preferences

In order to make decisions on detouring or staying in the truck, relative factors in the decision-making process should be analyzed. One of the factors is the driver agreements between Bouwvervoer and its drivers. These agreements cover the availability of sleepovers and drivers' specific needs and requirements. For example, some drivers prefer to sleep at home, while others are in the truck for personal reasons. Balancing costs with driver preferences can lead to higher morale and reduced turnover while improving drivers' comfort, rest quality, and overall job satisfaction. Drivers are sometimes fatigued after sleeping in their trucks, especially on consecutive nights. Ensuring drivers have adequate rest is essential for both safety and productivity.

Cost Parameters

Another essential variable to consider is cost parameters. There are several costs incurred while making decisions about detouring or going back to the base. Since the distance that a truck travel changes with the decision the driver makes, fuel cost should be considered as a part of the cost parameters. It is possible that drivers work additional hours as a result of a decision made. Hence, driver wage expenses linked to additional working hours can be one of the factors related to cost. Additionally, extended distance and time can affect the maintenance of the trucks. Therefore, maintenance costs for vehicles could be mentioned as a cost factor.

Rest time requirement

Monitoring working and rest times according to EU regulations is critical. Considering EU driving and rest regulations, drivers must follow strict rules regarding driving hours and rest periods. If a driver has already driven near the maximum allowed hours, staying in the truck might be the only legal option. Regulatory and safety-related constraints that determine the minimum required rest time for drivers.

Delivery Windows

The company values specific time frames within which shipments must be delivered to customers or recipients. Tight delivery deadlines might make returning home infeasible, as it could lead to delays. Staying in the truck near the delivery location may be necessary to meet scheduling demands. Detouring off the main route to return home can create inefficiencies in route planning, making the detour costly and potentially delaying future pickups or deliveries.

Other Potential Factors

Traffic Conditions: Factors related to road congestion, weather conditions, and traffic patterns.

Environmental Considerations: Factors related to emissions and sustainability.

Many other factors can impact cost and planning optimization. However, these three main route optimization metrics can be used to consider only the most influential measures to assess the effectiveness of detours and route choices with calculations. These metrics are distance traveled on each decision, time spent on the road, and cost-effectiveness of route choices based on fuel costs.

These are the possible factors based on the common logic and discussions with the company. Further in research which factors can be used in calculations and creation of algorithms will be reasoned and discussed. To sum up this research questions, list of important variables are as follows:

Driver Agreements	Availability of sleepovers Driver Specific requirements and needs
Cost Parameters	Fuel costs associated with detours and overnight stays. Driver wage expenses linked to additional working hours. Maintenance costs for vehicles affected by extended detours. Penalties for missed delivery windows.
EU Regulations	Legally decided rest periods between driving shifts. Minimum rest time needed to ensure driver well-being and regulations. Factors influencing driver fatigue and the need for rest.
Delivery Windows	Customer-specific delivery window constraints. Time-sensitive nature of cargo, impacting the importance of on-time deliveries. Penalties for failing to meet delivery windows.
Route Optimization Metrics	Distance travelled during detours. Time spent on detours. Cost-effectiveness of route choices.
Other Potential Factors	Real-time traffic data that affects the feasibility of detours. Weather conditions impacting road safety and travel time. Environmental impact of extended detours and idling vehicles. Environmental regulations and standards.

Table 9 List of important factors

3.4 Conclusion

This chapter explored the two sub research questions with reviewing existing literature. For the first research question, *"What algorithmic models can be developed to optimize decision-making in terms of cost-effectiveness, driver well-being, and compliance?"*, different models were examined. Each model was analysed based on its strengths and limitations in handling costs, regulatory compliance, and driver well-being. CP and DT are the most suitable models as structured decision-making is possible while considering multiple constraints. These models will serve as the basis for the algorithm development in the following chapter.

For the second research question, *"What factors influence the decisions between a driver detouring home or staying in the truck?"*, several decision-making variables are decided. These include driver agreements, cost parameters, regulatory constraints and. These insights will be integrated into the algorithm.

Chapter 4: MPSM

Chapter 4 follows steps of MPSM in detail, after understanding the problem and identifying relevant factors, based on solution design the steps to complete a working code with algorithm logic is described. Section 4.1 describes solution planning with key objectives and factors. Followed by section 4.2 problem analysis with the creation of data. Section 4.3 shows algorithm design with its steps. Solution choice and implementation is discussed in sections 4.4 and 4.5. Lastly section 4.6 demonstrates on evaluation of results.

4.1 Solution Planning

Key Objectives

Decision-making of driver detours: The main scope of this project is to develop an algorithm that supports the decision-making process of driver detours. It involves deciding whether a driver should sleep the night in the truck or go to the base.

Cost-Effectiveness: An important goal is to improve the cost by making decisions that consider the unnecessary costs of overnight stays, lengthy detours, and other factors. It also includes fuel costs.

Driver Agreements: Bouwvervoer makes commitments for its drivers to consider drivers' agreements. Such as the maximum number of overnight stays per week agreed with drivers. Drivers can have different employment agreements based on specific needs.

Rest Time Compliance: Making sure the well-being of drivers is crucial. The project should consider rest time requirements, ensuring drivers have enough time to rest and meet the EU regulations standards. (The European Commission (2020) regulates the driving time and rest periods for drivers in the EU.")

Key Factors

Below is the list of key factors should be included in the data collection step to use in the calculations.

Driver Information	• Names of drivers. • Truck details, including license plates and fuel efficiency.
Daily Locations	• Last city visited at the end of the day. • First city to visit the next day.
Distance and Time	• Distance and time from the last city to the base and from the base to the next city. • Direct distance and time from the last city to the next city.
Sleepover Availability	• Per-driver maximum sleepover count allowed based on agreements.

	• Sleepover usage per day to track current totals.
Fuel Data	• Fuel prices during the relevant period. • Fuel consumption efficiency for each truck.

Table 10 Key factor to include in algorithm

Selection of Decision-Making Models

Constraint Programming is chosen as the modeling approach because:

1. CP allows the inclusion of soft constraints, making it possible to prioritize driver rest and satisfaction while still focusing on cost optimization. It provides a structured framework to enforce constraints like sleepover limits.
2. CP directly integrates constraints into the decision-making process, ensuring that all regulatory requirements are strictly followed. It efficiently handles optimization (minimizing costs) while respecting feasibility rules.
3. It is capable of balancing multiple factors, such as fuel costs and sleepover expenses, to minimize overall operational costs efficiently. It can be extended with additional constraints (e.g., other operational factors) if needed.

Planning the CP-Based Solution

Decision to Optimize: Should the driver stay in the truck or go to the closest base at the end of the day?

Key Actions:

- Calculate the cost of staying in the truck (fuel cost for traveling directly from the last city to the next city).
- Calculate the cost of returning to the base and then traveling to the next city the next day.
- Compare the costs and ensure the decision complies with sleepover limits.

Variables

Decision variables:

$$D \in \{0,1\}, where \ :$$

D=0: Driver stays in the truck.

D=1: Driver goes to the base

Cost variables:

C_{direct}: Fuel cost for traveling directly from the last city to the next city.

C_{base}: Fuel cost for traveling from the last city to the base and then to the next city.

Sleepover Counter:

S_{used}: Cumulative sleepovers used by the driver so far.

S_{max}: Maximum allowable sleepovers per week as per agreement.

Fuel Efficiency:

F_{eff}: Fuel efficiency of the truck (liters per kilometer).

P_{fuel}: Fuel price (cost per liter).

Distance:

D_{direct}: Distance between the last city and the next city.

D_{base}: Distance from the last city to the base.

$D_{base\_to\_next}$: Distance from the base to the next city.

Constraints

Sleepover Limit:

$$S_{used} + D \leq S_{max}$$

if D=0, the sleepover counter S_{used} is incremented *by 1, and the total remains below* S_{max}

Cost Feasibility

$$C_{direct} = D_{direct} / F_{eff} \times P_{fuel}$$

$$C_{base} = (D_{base} + D_{base\_to\_next}) / F_{eff} \times P_{fuel}$$

Logical Constraints: Only one decision can be chosen: D=0 or D= 1

4.2 Problem Analysis

This section of thesis is about how the provided data was analysed and how the key factor is structured. As it was mentioned in the context chapter the company has provided four excel files to help with the solution phase. From this files, relevant data was extracted and organized into a one final table (note that Driver names and License Plates was not included in the screenshot due to ethical reasons, but they are part of the data), focusing on key parameters necessary for decision-making and calculations. The extracted data spans July 1-12, with mainly completed data available for July 1-4 and July 8-11 because of weekend. The extracted table includes:

Last City	Distance L-B-F	Time L-B-F	First City Next Day	Distance L-F	Time L-F	Avaiab	Used Sleep	Fuel Efficiency
KONINGSBOSCH	235.0	160.0	SASSENHEIM	215.0	135.0	0	0	14.65
OOSTERHOUT	160.0	110.0	ANTWERPEN	67.0	42.0	4	0	26.34
BEST	160.0	115.0	VAASSEN	110.0	75.0	2	1	23.35
			XANTEN					25.90
GAMEREN	80.0	70.0	VEGHEL	50.0	45.0	2	1	30.93
WEERT	140.0	110.0	VEGHEL	60.0	50.0	0	0	28.71
HAARLEM	205.0	150.0	RHEDEN	120.0	80.0	2		26.45
TILBURG	120.0	100.0	MIERLO	60.0	50.0	4	0	32.10
RHEDEN	100.0	80.0	LELYSTAD	80.0	60.0	0	1	27.66
APELDOORN	140.0	110.0	AALST	150.0	105.0	4	0	25.02
OOSTERHOUT	170.0	110.0	ZANDHOVEN	60.0	50.0	4	0	26.42
VEGHEL	200.0	150.0	YERSEKE	150.0	105.0	0	0	28.95
KONINGSBOSCH	310.0	210.0	ROTTERDAM	180.0	120.0	4	0	29.10
ZWARTEMEER	150.0	110.0	WAGENINGEN	120.0	90.0	2	1	30.86
AALST	200.0	160.0	VARSSEVELD	100.0	80.0	0	1	28.27
VEGHEL	150.0	125.0	PAAL-BERINGEN	100.0	80.0	0	1	26.98
HAARLEM	265.0	190.0	WANSSUM	180.0	120.0	0	1	26.64
WESTERVOORT	40.0	40.0	WESTERVOORT			2	1	26.94
						0	1	24.89
OOSTERHOUT	160.0	120.0	LOCHRISTI	100.0	80.0	4	0	24.05
DILSEN-STOKKEM	300.0	190.0	HAARLEM	200.0	130.0	0	0	28.26
OOSTERHOUT	110.0	80.0	RIDDERKERK	50.0	40.0	0	0	26.16
XANTEN	255.0	210.0	VOORBURG	150.0	105.0	0	0	23.19
GHLIN						4	0	27.78
BREMEN	200.0	120.0	OTTERSBERG	100.0	60.0	4	0	29.20
DILSEN-STOKKEM	300.0	190.0	HAARLEM	200.0	130.0	0	0	22.84
			TILBURG			0	1	36.81
						2	0	
HAARLEM	180.0	140.0	WAGENINGEN	100.0	80.0	0	0	23.24
BREDA	260.0	180.0	ENSCHEDE	180.0	120.0	0	1	27.96
RHEDEN	140.0	120.0	ALMELO	80.0	60.0	4	0	26.48
VEGHEL							1	21.43
HAARLEM	260.0	180.0	WANSSUM	180.0	120.0	0	0	23.26
DILSEN-STOKKEM	300.0	190.0	HAARLEM	200.0	130.0	0	0	28.17
WESTERVOORT	80.0	80.0	DEVENTER	40.0	40.0	0	0	28.01

Figure 7 Created data set from the company data

Driver Information: Names of the drivers, license plate and relevant fuel efficiency of trucks

Daily Locations: Last City visited each day, and the first city visited the next day

Distance and Time to the next city: Calculated distance and time it takes from last city to base and from based to first city next day or distance and time from last city to the first city next day. The kilometres and approximate times were taken from Google Maps.

Sleepover Availability and Usage: Available sleepover count per driver and usage on each day

Fuel data: Approximate constant value for fuel prices in those periods to be used in calculations

Assumptions and general rules

Sleepover Costs:

- Fixed cost of €40 for the first night.
- Incremental cost of €59.40 for consecutive nights.

Fuel Price: Fuel prices and sleepover costs may vary but are treated as constants for simplicity.

Compliance Requirements:

- A minimum of 9 hours of rest daily.
- Cannot exceed maximum driving hours.

Driver Well-Being:

- Maximum number of sleepovers agreed upon by each driver is incorporated into the extracted data.
- Sleepover agreements must be respected during decision-making.

Limitations of the Data:

- Data for some days is missing, potentially causing gaps in decision-making. Unfortunately, this is the case for all days, there is no single day with full data. Therefore, either empty cells will be filled with assumed data or calculations will be done only for drivers with full data in solution generation steps.
- In reality there are different bases for drivers to stay but for simplicity and data security this project only considers one warehouse base in Wageningen. Which makes impossible to get real life result to compare with historical data for validation.

4.3 Solution Generation

The Figure 9 demonstrates algorithm design with inputs and outputs

Algorithm Design

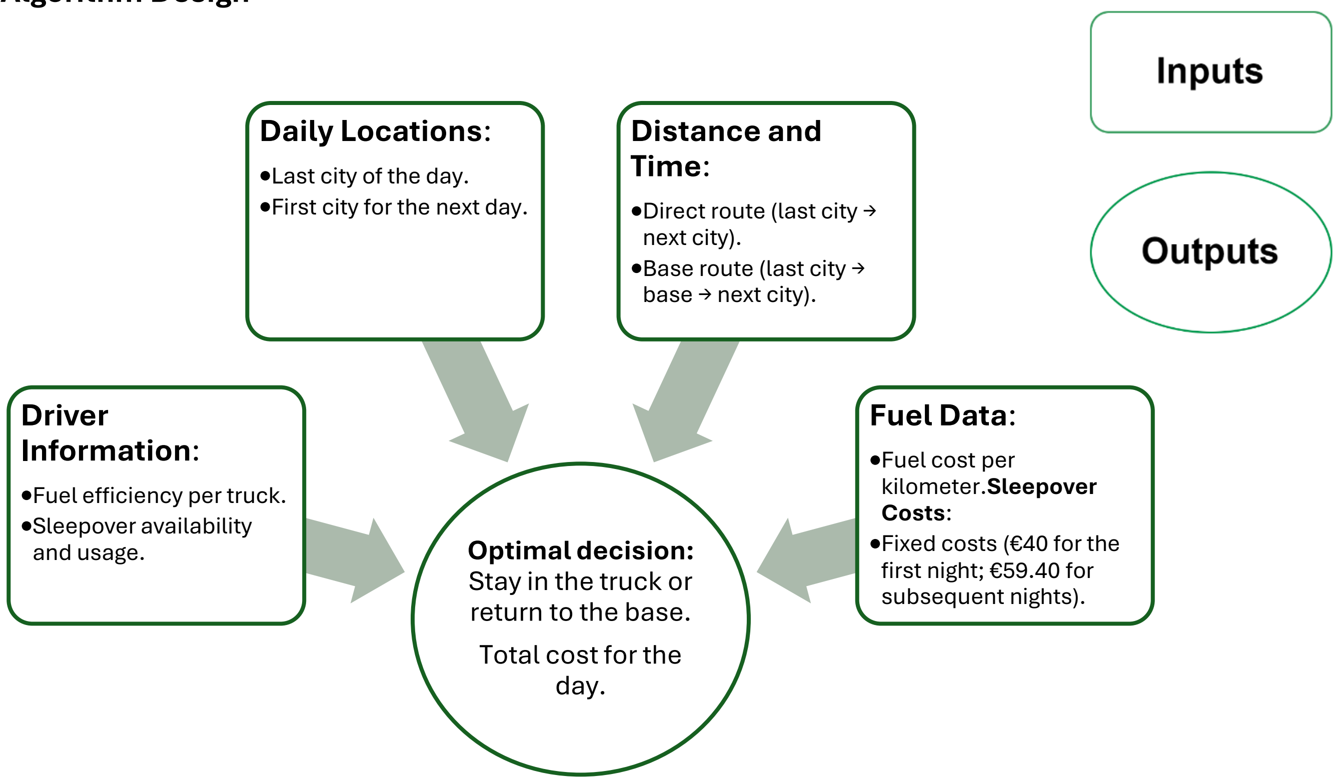


Figure 8 Algorithm Design

In the table 11, the steps of the algorithm are outlined with description for each step.

Algorithm Steps

Steps	**Descriptions**
Initialization	Input data for all drivers and their respective routes. Initialize cumulative costs and sleepover usage.
Calculating Costs	*Option 1 (Stay in Truck):* $C_{direct} = D_{direct} / F_{eff} \times P_{fuel}$ *Option 2 (Return to Base):* $C_{base} = (D_{base} + D_{base\_to\_next}) / F_{eff} \times P_{fuel}$ Adding sleepover costs for the driver if applicable.
Applying Constraints	Ensuring $S_{used} + D \leq S_{max}$ Checking the decision complies with rest requirements and driving time limits.
Decision-Making	Compare C_{direct} and C_{base} Select the option with the lowest cost that satisfies constraints.

Updating Records	If staying in the truck, increment S_{used} for the driver Update cumulative costs for the day.
Repeat for All Drivers	Applying the algorithm for each driver's data daily.

Table 11 Algorithm Steps

4.4 Solution Choice

Criteria for choosing solution

Category	Criteria
Cost Minimization	The solution must reliably identify the cheaper option between staying in the truck and returning to the base.
Compliance with Constraints	Must respect sleepover agreements. Adhere to rest and driving time regulations.
Scalability	The solution should handle multiple drivers and large datasets efficiently.
Ease of Implementation	Should integrate with existing systems (e.g., data input/output via Excel)
Flexibility	Ability to adapt to changes, such as dynamic fuel prices or sleepover costs.

Table 12 Criteria for solution choice

Potential Solutions

Solution 1: Manual Framework

In this option drivers or planners manually calculate costs based on provided data and rules. Which is simple to implement without technical expertise, but it is prone to have errors and time intensive. Also, it would be difficult to work with large datasets.

Solution 2: Spreadsheet-Based Automation

Solution 2 is based on using Excel formulas or macros to calculate and compare costs and applying constraints. Pros of this solution is that it is easy to implement and understand and excel is usually familiar tool for most users. However, the complex logic could still require manual handling with increasing datasets.

Solution 3: Python-Based Algorithm

Python-based solution uses coding to automate calculations and apply constraints to generate results. There are many advantages to this option such as handling large data efficiently, fully automated and minimum errors, adaptable to changes. Requiring programming skills would be biggest challenge and disadvantage for this option

Decision-making process paragraph

After analyzing all options according to the evaluation criteria in Table 10, it is clear that the most suitable choice is Solution 3. Therefore, the Table 11 gives the overview for the selected solution.

Category	**Criteria**
Cost Minimization	The Python algorithm uses exact calculations for cost comparison.
Compliance with Constraints	Sleepover and rest constraints are incorporated directly into the logic.
Scalability	The script can handle hundreds of drivers and records efficiently.
Ease of Implementation	Outputs are user-friendly such as CSV files (which are suitable for easy analysis
Flexibility	Easy to adjust parameters like fuel prices or sleepover costs.

Table 13 Overview of selected solution

4.5 Solution Implementation

To implement the algorithm design there are several steps to focus. This sub-chapter divides the written code into important parts and explain those parts based on screenshots from Python script.

First Version of Code

During coding phases the different versions of codes are built, the first version of code is simpler than the final version. The difference between versions is explained at the end of chapter.

Loading Data

As it was described in Figure 7, Data from the company was combined with only necessary columns into the one final excel file called "logistics_data.csv". The empty cells were deleted before starting next step to prevent errors and get clear result. After deleting drivers with empty data, the code was run with only 28 drivers. With the aim of loading excel file in the python script necessary libraries were downloaded as it shows on Figure 9 bellow.

```
import pandas as pd

# Load the data from  CSV file
data = pd.read_csv("logistics_data.csv")

print(data.columns)
```

Figure 9 Loading data from Excel file | Screenshot

Defining Constants

The approximate fuel price for July 2024 can be assumed based on [Dieselprijzen Nederland | evofenedex] and marked as constants in the code bellow. The constant number can be changed easily for the future calculations or more detailed fuel costs can be added to each driver to get more accurate results. In this project for simplicity constant number is used.

```
# Constants
FUEL_PRICE = 1.542  # €/liter
```

Figure 10 Fuel Price Constant

Initialize

A list called results is created to store the decisions, costs, and other outputs for each driver. This will later be converted into a table and exported.

```
# Initialize results storage
results = []
```

Figure 11 Initialization

Looping

Since calculations should be for all drivers not only one the loop is created to go through each driver in the data.

```
# Loop through each driver in the data
for _, row in data.iterrows():
    # Extract relevant information from the row
    driver = row['Driver']
    distance_L_B_F = row['Distance L-B-F']  # Distance from last city to base to first city
    distance_L_F = row['Distance L-F']  # Direct distance from last city to first city
    available_sleepover = row['Available Sleepover']
    used_sleepover = row['Used Sleepover']
    fuel_efficiency = row['Fuel Efficiency']  # km/l
```

Figure 12 Loop Through each driver | Screenshot

Decision-Making Logic

The first check ensures that the driver has enough available sleepovers before even considering staying in the truck. If the driver has available sleepovers, the algorithm will compare the costs of staying in the truck and returning to the base. If no sleepovers are available, the decision defaults to "Return to Base." In case sleepovers are available, the algorithm compares the two options and the option with the lower cost is selected as the decision. Secondary Constraint checks whether incrementing the used_sleepover count will still respect the sleepover limits: If adding 1 to used_sleepover exceeds the available_sleepover, the decision is overridden, and the driver must "Return to Base". This ensures no invalid sleepover counts are recorded, even if an earlier condition allows the "Stay in Truck" option.

```
# Decision-making logic
if used_sleepover + 1 <= available_sleepover:  # Check if sleepovers are within limits
    if cost_L_F <= cost_L_B_F:
        decision = "Stay in Truck"
        # Verify again before incrementing used_sleepover
        if used_sleepover + 1 <= available_sleepover:
            used_sleepover += 1  # Increment used sleepover count
        else:
            # If incrementing exceeds the limit, change the decision
            decision = "Return to Base"
            total_cost = cost_L_B_F
    else:
        decision = "Return to Base"
        total_cost = cost_L_B_F
else:
    decision = "Return to Base"
    total_cost = cost_L_B_F
```

Figure 13 Decision-making logic | Screenshot

Cost Calculations

```
    # Calculate fuel costs for both options
    cost_L_B_F = distance_L_B_F / fuel_efficiency * FUEL_PRICE
    cost_L_F = distance_L_F / fuel_efficiency * FUEL_PRICE
```

Figure 14 Cost Calculations Code

Storing Results

Combines the decision, cost, and updated sleepover count for the driver to the results list

```
    # Store the result for this driver
    results.append({
        'Driver': driver,
        'Decision': decision,
        'Total Cost (€)': round(total_cost, 2),
        'Updated Used Sleepover': used_sleepover
    })
```

Figure 15 Storing Code | Screenshot

Exporting Results

This part of the code converts the results list into a Data Frame and exports the results to a CSV file named decision_results.csv. Lastly prints a confirmation message in the terminal.

```python
# Convert results to a DataFrame for analysis
results_df = pd.DataFrame(results)

# Save the results to a new CSV file
results_df.to_csv("decision_results.csv", index=False)

print("Decisions and costs saved to 'decision_results.csv'")
```

Figure 16 Exporting Results | Screenshot

Final version of code

Data for 2 days

In the final version code uses two separate datasets for each day. Column names should match to verify the connection between 2 excel file, this enables processing different days.

```python
# Load data for both days
# Specify header=0 to ensure the first row is treated as the header
day1_data = pd.read_csv("day1_logistics_data.csv", header=0)
day2_data = pd.read_csv("day2_logistics_data.csv", header=0)

# Print column names to verify correctness
print("Day 1 Columns:", day1_data.columns)
print("Day 2 Columns:", day2_data.columns)

# Ensure column names are consistent by stripping spaces
day1_data.columns = day1_data.columns.str.strip()
day2_data.columns = day2_data.columns.str.strip()

# Rename columns explicitly to match the dataset
expected_columns = ['Driver', 'Last City', 'Distance L-B-F', 'Time L-B-F',
                    'First City Next Day', 'Distance L-F', 'Time L-F',
                    'Available Sleepover', 'Used Sleepover', 'LicensePlate',
                    'Fuel Efficiency']
day1_data.columns = expected_columns
day2_data.columns = expected_columns
```

Figure 17 Loading data for 2 days

Sleepover cost

According to the algorithm design in 5.4, the input of sleepover costs is defined in the screenshot of code below. Also, now sleepover costs are part of total cost.

```
FIRST_NIGHT_COST = 40  # € for the first night of sleepover
CONSECUTIVE_NIGHT_COST = 59.40  # € for consecutive nights

# Initialize results for both days
results_day1 = []
results_day2 = []

# Function to calculate total cost with sleepover
def calculate_total_cost(stay_cost, used_sleepover):  2 usages
    if used_sleepover == 0:  # First night
        sleepover_cost = FIRST_NIGHT_COST
    else:  # Consecutive nights
        sleepover_cost = CONSECUTIVE_NIGHT_COST
    return stay_cost + sleepover_cost
```

Figure 18 Sleepover Cost

Updating used sleepover

The main difference between day 1 and day 2 datasets is that in the second day the used sleepover column is empty and gets filled after processing day 1. Staying in the truck in the first day increases used sleepover count in the second day

```
# Save Day 1 results and update Day 2's Used Sleepover column
results_day1_df.to_csv("day1_results.csv", index=False)
day2_data['Used Sleepover'] = results_day1_df['Updated Used Sleepover']
```

Figure 19 Updating Constraints

Time Constraints

Minimum of 9 hours of rest daily and not exceeding maximum driving hours constraints were considered in the new version of code

```
if total_time_stay <= MAX_DRIVING_HOURS and total_time_stay + MIN_DAILY_REST <= 24 * 60:
    # No time constraint violated
    constraint_violated = "None"
else:
    # Time constraint violated
    constraint_violated = "Violated"
```

Output

As a result, for final version of code two output files expected. Each file contains driver names, decisions, total cost and updated used sleepovers.

```
# Convert results of Day 2 to DataFrame
results_day2_df = pd.DataFrame(results_day2)

# Save Day 2 results
results_day2_df.to_csv("day2_results.csv", index=False)

print("Day 1 and Day 2 decisions saved to 'day1_results.csv' and 'day2_results.csv'")
```

Figure 20 Converting results to Data frame

4.6 Solution Evaluation

<u>Evaluation for the first version of code</u>

The python script resulted in working code and as a result it gives excel file called “decision_results” with driver names (was not included in the screenshot due to ethical reasons), recommended decisions, total cost and updated used sleepover columns. In order to evaluate this result five main category for evaluation criteria is decided. Those are:

Cost Effectiveness: Does the algorithm pick the cheaper option?

Constraints: For example, does the algorithm ensure that used sleepovers never exceeds the available sleepovers?

Accuracy: Are results matching with manual calculations according to the example scenarios?

Usability: Does results look easy to understand and analyse the situation?

Decision	Total Cost (â‚¬)	Updated Used Sleepover
Return to Base	24.74	0
Stay in Truck	3.92	1
Stay in Truck	7.26	2
Stay in Truck	2.49	2
Return to Base	7.52	0
Return to Base	11.95	
Stay in Truck	2.88	1
Return to Base	5.57	1
Return to Base	8.63	0
Stay in Truck	3.5	1
Return to Base	10.65	0
Stay in Truck	9.54	1
Stay in Truck	6	2
Return to Base	10.91	1
Return to Base	8.57	1
Return to Base	15.34	1
Stay in Truck	6.41	1
Return to Base	16.37	0
Return to Base	6.48	0
Return to Base	16.96	0
Stay in Truck	5.28	1
Return to Base	20.25	0
Return to Base	11.94	0
Return to Base	14.34	1
Stay in Truck	4.66	1
Return to Base	17.24	0
Return to Base	16.42	0
Return to Base	4.4	0

Figure 21 File output of version1 code

Example Scenario

To compare code results with manual calculations data for first 5 drivers is calculated manually and will be used for validation. For calculations same logic of the python code is used.

Driver	Last City	Distance L-B-F	Time L-B-F	First City Next Day	Distance L-F	Time L-F	Available Sleepover	Used Sleepover	Fuel Efficiency
1	KONINGSBOSCH	235	160	SASSENHEIM	215	135	0	0	14.65
2	OOSTERHOUT	160	110	ANTWERPEN	67	42	4	0	26.34
3	BEST	160	115	VAASSEN	110	75	2	1	23.35
4	GAMEREN	80	70	VEGHEL	50	45	2	1	30.93
5	WEERT	140	110	VEGHEL	60	50	0	0	28.71

Table 14 Example Scenario

Formula for calculating cost:

$$Cost = \frac{Distance}{Fuel\ Efficiency} \times Fuel\ Price$$

Drivers	Go to base cost	Stay in truck cost	Decision	Updated Sleepover
Driver 1	24.76	22.63	Base	0
Driver 2	9.37	3.93	Truck	1
Driver 3	10.57	7.26	Truck	2
Driver 4	3.99	2.49	Truck	2
Driver 5	7.52	3.22	Base	0

Table 15 Calculations of example scenarios

Based on the example scenarios available sleepover rule is applied and working since for the Driver1 staying in the truck is the cheaper option but because the max sleepover is decided to be zero the decision is going to base. Additionally updated sleepovers are matching with de output of the python code. Lastly having exactly same numbers for costs demonstrates that calculation is done correctly.

Evaluation for the final version of code

The output of the final version is fairly similar to the first version, since the numbers are same the calculations are working properly. However, there is a problem with the final version. Since staying in the truck has a cost of:

$$Cost = \left(\frac{Distance}{Fuel\ Efficiency} \times Fuel\ Price\right) + Sleepover\ Cost$$

And in comparison, going to the base has only fuel cost in all cases at least for the first 2 days going to base is always cheaper option. For that reason, output has “go to base” decision in all cells. There can be few reasons for that, firstly as it was mentioned extra cost of min 40 euro, and secondly since kilometers are very close to each other, only considering fuel cost can not exceed 40 euros of sleepover cost. If different bases are added and there is a big difference in kilometers code can result in different outcomes.

Evaluation Criteria of Algorithm

Both versions of code meet the criteria that are decided on the beginning of this sub-chapter however their differences are summarized in the table below.

First Version	**Last version**
Simple and effective	Handles more constraints such as sleepover and time constraints
Does not include sleepover cost	Sleepover costs have a negative impact on decisions
Only considers 1 day	Considers 2 days and can easily be updated for more days if datasets are ready.

Table 16 Evaluation Summary of algorithms

Validation:

It is important to mention that the logic of the whole algorithm was discussed few times with the stakeholders at Bouwvervoer and as a result including the first city next day was decided to make comparison more logical. Secondly the situation where the comparison between costs is 50-50 the personal choice of driver should count; however, this was not implemented in the project due to complexity.

4.7 Conclusion

This chapter demonstrated a breakdown of the problem-solving process and the steps taken to develop a decision-making algorithm for optimizing driver detours in. The chapter began by analyzing the nature of the problem, identifying key influencing factors such as fuel costs, driver agreements, regulatory constraints, and sleepover expenses. These factors were then systematically incorporated into the algorithm design to ensure that decision-making aligns with cost-effectiveness, compliance, and driver well-being.

The algorithm logic was structured and implemented in Python, with detailed steps outlining how the model processes data, applies constraints, and generates optimal recommendations. The first version of the script provided a basic yet functional decision-making tool, focusing primarily on cost calculations. The final version enhanced the model by integrating multi-day planning, and improved sleepover cost handling.

To assess the effectiveness of the developed algorithm, an evaluation was conducted based on predefined criteria, ensuring that the system differentiates between staying in the truck and detouring to the base. The results confirmed that the algorithm minimizes operational costs however, some limitations were identified, such as the high fixed cost of sleepovers impacting decision outcomes and the assumption of a single base location.

Chapter 5: Conclusion

5.1 Main Results

This thesis focused on building an algorithm description to address inefficiencies in logistics operations by automating decision-making for truck drivers. The study followed the MPSM to systematically identify the core problem, analyze relevant variables, and develop solution to optimize driver decisions. Based on the problem analysis and requirements, an algorithm was designed to balance cost efficiency, driver well-being. The algorithm incorporates constraints such as fuel costs, sleepover expenses, and driver agreements to generate optimal routing recommendations. A Python script was developed to implement the algorithm, going beyond just theoretical modeling. The code structures decision-making, with different versions evaluated based on predefined criteria. The first version focused on fundamental cost calculations and constraint handling, while the final version incorporated additional factors such as multi-day planning, and sleepover agreements. The primary research question—How can a logistics company's decision-making algorithm be designed to optimize driver decisions, balancing operational cost-effectiveness and driver well-being? has been addressed through the development of a constraint-based optimization model. The algorithm achieves cost savings by minimizing unnecessary detours and ensuring optimal overnight stays based on predefined constraints. Additionally, the inclusion of driver preferences and sleepover agreements helps maintain driver satisfaction.

5.2 Contributions

This thesis makes several key contributions to logistics optimization and decision-support systems. Firstly, it presents a structured approach to solving driver detour decision-making challenges by leveraging algorithmic automation ensuring a balance between cost efficiency, compliance with regulations, and driver well-being. Another significant contribution of this study is its practical implementation using real company data from Bouwvervoer. The algorithm was designed based on actual operational constraints, making it directly applicable within the industry. The study provides a clear framework for integrating AI-driven decision-making models into logistics and transportation management systems, serving as a reference for further advancements in AI-based route planning. Additionally, the thesis considers multiple decision-making criteria. Unlike conventional models that prioritize cost alone, this project incorporates driver preferences, regulatory constraints, and operational flexibility, leading to a more comprehensive and human-centric approach to transportation logistics. The ability to evaluate different versions of the algorithm further strengthens its practical application.

5.3 Limitations

1. Data assumptions for simplification: Only one based was used in the algorithm which made impossible to compare outputs with the real-world historical data. Data gaps in data made very difficult to include all 12 days of data since data for one driver is not consistent every day.
2. Sleepover Cost: Even it is better to include this important constraint in the implementation of code, the algorithm outputs are heavily affected by high fixed cost of sleepover
3. No real time factors: Factors like weather, traffic, road conditions were not included.
4. Static fuel prices: Creating list of fuel prices for specific time periods and truck types can increase the accuracy of the calculations, however for simplicity and lack of data constant variable was decided on fuel price for all the trucks.
5. Single Base Location: The algorithm assumes only one warehouse base, making it difficult to compare outputs with real-world historical data. In reality, drivers may have multiple base options, and incorporating these would enhance decision accuracy.
6. Driving Hours: The total number of hours a driver works in a day can be quantified, and if returning to the base results in exceeding the permissible limit, the algorithm can take this into account. However, in this project, this calculation was not performed due to the unavailability of necessary data.

5.4 Recommendation

The decision-making algorithm can be improved by some strategic recommendations. A major improvement to this includes real base locations, where drivers are assigned a base closest to their location. This change would substantially increase the accuracy of calculations, making them better representative of actual operations. Also, it would resolve the present issue in which the algorithm usually favored “go to base” as a decision to output, which results in a less balanced and more realistic output. Dynamic cost adjustments need to be introduced to optimize the calculation of costs. The algorithm should account for real-time market changes in fuel pricing and time-dependent unit costs. This will enhance the precision of financial planning, thereby increasing the quality of decision-making. A key recommendation is the incorporation of real-time traffic and weather information. Incorporating such external factors into the algorithm would enable route planning to be more responsive and adaptable to changing road conditions. This would make sure that decision-making is made in a cost-effective manner while accounting for practical limitations such as time lost in traffic or bad weather. Even the implementation of some of these improvements would greatly increase the reliability and usability of the algorithm, which is a valuable asset in logistics optimization.

5.5 Future Research

In order to extend the algorithm and improve its capabilities. For example, one potential direction is the use of heuristic methods that can perform a simplifying way of making decisions. CP guarantees optimality, but these heuristic methods provide promising solutions with flexible computations when optimization isn't feasible due to computational limitations. Testing the algorithms for longer periods would also help. Its performance can be analyzed in more detail by running the algorithm on a larger dataset over a longer time period. Moreover, machine learning techniques can be integrated to further increase the adaptability of the algorithm. Instead of static, rule-based constraints, using a historical-data-trained machine learning model could make for more adaptive, data-driven decision-making. Finally, the addition of a driver satisfaction metric to measure driver well-being would be useful to show how well the algorithm balances cost-effectiveness with driver satisfaction. This would help refine decision-making parameters for operational efficiency to not come at the cost of employee satisfaction by analyzing driver feedback and preferences over time. Such future enhancements could evolve the algorithm to be a more sophisticated and scalable decision-making tool for logistics optimization.

References

Harrison, A., & Van Hoek, R. I. (2008). *Logistics Management and strategy: Competing Through the Supply Chain*. Pearson Education.

Sebata, D. (2024). Investigating how logistics operations affect an organization's financial performance. In *IntechOpen eBooks*. https://doi.org/10.5772/intechopen.1004304

Lombardi, M., Milano, M., & Bartolini, A. (2016). Empirical decision model learning. *Artificial Intelligence*, *244*, 343–367. https://doi.org/10.1016/j.artint.2016.01.005

Rossi, F., Van Beek, P., & Walsh, T. (2008). Chapter 4 Constraint Programming. In *Foundations of artificial intelligence* (pp. 181–211). https://doi.org/10.1016/s1574-6526(07)03004-0

Clautiaux, F., & Ljubić, I. (2024). Last fifty years of integer linear programming: A focus on recent practical advances. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2024.11.018

Alexakis, K., Benekis, V., Kokkinakos, P., & Askounis, D. (2024). Genetic algorithm-based multi-objective optimisation for energy-efficient building retrofitting: A systematic review. *Energy and Buildings*, 115216. https://doi.org/10.1016/j.enbuild.2024.115216

Bhowmik, O., Chowdhury, S., Ashik, J. H., Mahmud, G. I., Khan, M. M., & Hossain, N. U. I. (2024). Application of artificial intelligence in reverse logistics: A bibliometric and network analysis. *Supply Chain Analytics*, *7*, 100076. https://doi.org/10.1016/j.sca.2024.100076

Rabia, M. a. B., & Bellabdaoui, A. (2022). Simulation-based analytics: A systematic literature review. *Simulation Modelling Practice and Theory*, *117*, 102511. https://doi.org/10.1016/j.simpat.2022.102511

Mahesh, B. (2020). Machine Learning Algorithms - a review. *International Journal of Science and Research (IJSR)*, *9*(1), 381–386. https://doi.org/10.21275/art20203995

Heerkens, H., & Van Winden, A. (2021). Solving managerial problems systematically. In *Routledge eBooks*. https://doi.org/10.4324/9781003186038

Bouwvervoer BV. (2025). Bouwvervoer.nl. https://bouwvervoer.nl/

Appendix 1

```
import pandas as pd


data = pd.read_csv("logistics_data.csv")

print(data.columns)

FUEL_PRICE = 1.542

results = []

for _, row in data.iterrows():
    driver = row['Driver']
    distance_L_B_F = row['Distance L-B-F']
    distance_L_F = row['Distance L-F']
    available_sleepover = row['Available Sleepover']
    used_sleepover = row['Used Sleepover']
    fuel_efficiency = row['Fuel Efficiency']

    cost_L_B_F = distance_L_B_F / fuel_efficiency * FUEL_PRICE
    cost_L_F = distance_L_F / fuel_efficiency * FUEL_PRICE

if used_sleepover + 1 <= available_sleepover:
    if cost_L_F <= cost_L_B_F:
        decision = "Stay in Truck"
        if used_sleepover + 1 <= available_sleepover:
            used_sleepover += 1
        else:
            decision = "Return to Base"
            total_cost = cost_L_B_F
    else:
        decision = "Return to Base"
        total_cost = cost_L_B_F
else:
    decision = "Return to Base"
    total_cost = cost_L_B_F

    results.append({
        'Driver': driver,
        'Decision': decision,
        'Total Cost (€)': round(total_cost, 2),
        'Updated Used Sleepover': used_sleepover
    })

results_df = pd.DataFrame(results)
results_df.to_csv("decision_results.csv", index=False)
print("Decisions and costs saved to 'decision_results.csv'")
```

Figure 22 Full Code for first version

```python
import pandas as pd
day1_data = pd.read_csv("day1_logistics_data.csv", header=0)
day2_data = pd.read_csv("day2_logistics_data.csv", header=0)
day1_data.columns = day1_data.columns.str.strip()
day2_data.columns = day2_data.columns.str.strip()
expected_columns = ['Driver', 'Last City', 'Distance L-B-F', 'Time L-B-F',
                    'First City Next Day', 'Distance L-F', 'Time L-F',
                    'Available Sleepover', 'Used Sleepover', 'LicensePlate',
                    'Fuel Efficiency']
day1_data.columns = expected_columns
day2_data.columns = expected_columns
FUEL_PRICE = 1.542  # €/liter
FIRST_NIGHT_COST = 40  # € for the first night of sleepover
CONSECUTIVE_NIGHT_COST = 59.40  # € for consecutive nights
MIN_DAILY_REST = 9 * 60  # 9 hours in minutes
MAX_DRIVING_HOURS = 10 * 60  # 10 hours in minutes
results_day1 = []
results_day2 = []
def calculate_total_cost(stay_cost, used_sleepover):  2 usages
    if used_sleepover == 0:
        sleepover_cost = FIRST_NIGHT_COST
    else:
        sleepover_cost = CONSECUTIVE_NIGHT_COST
    return stay_cost + sleepover_cost
for _, row in day1_data.iterrows():
    driver = row['Driver']
    distance_L_B_F = row['Distance L-B-F']
    distance_L_F = row['Distance L-F']
    available_sleepover = row['Available Sleepover']
    used_sleepover = row['Used Sleepover']
    fuel_efficiency = row['Fuel Efficiency']
    total_time_stay = row['Time L-F']
    total_time_return = row['Time L-B-F']
    cost_L_B_F = distance_L_B_F / fuel_efficiency * FUEL_PRICE
    cost_L_F = distance_L_F / fuel_efficiency * FUEL_PRICE
    time_constraint = "None"
    if used_sleepover + 1 <= available_sleepover:
        total_cost_stay = calculate_total_cost(cost_L_F, used_sleepover)

        if total_time_stay <= MAX_DRIVING_HOURS and total_time_stay + MIN_DAILY_REST <= 24 * 60:
            if total_cost_stay < cost_L_B_F:
                decision = "Stay in Truck"
                used_sleepover += 1
                total_cost = total_cost_stay
            elif total_cost_stay > cost_L_B_F:
                decision = "Return to Base"
                total_cost = cost_L_B_F
            else:
                decision = "Driver's Choice"
                total_cost = total_cost_stay
        else:
            decision = "Return to Base"
            total_cost = cost_L_B_F
            time_constraint = "Violated"
    else:
        decision = "Return to Base"
```

Figure 23 Full code for final version(part1)

```
            total_cost = cost_L_B_F

        results_day1.append({
            'Driver': driver,
            'Decision': decision,
            'Total Cost (€)': round(total_cost, 2),
            'Updated Used Sleepover': used_sleepover,
            'Time Constraint': time_constraint
        })

    results_day1_df = pd.DataFrame(results_day1)
    results_day1_df.to_csv("day1_resultsfinal.csv", index=False)
    day2_data['Used Sleepover'] = results_day1_df['Updated Used Sleepover']
    for _, row in day2_data.iterrows():
        driver = row['Driver']
        distance_L_B_F = row['Distance L-B-F']
        distance_L_F = row['Distance L-F']
        available_sleepover = row['Available Sleepover']
        used_sleepover = row['Used Sleepover']
        fuel_efficiency = row['Fuel Efficiency']
        total_time_stay = row['Time L-F']
        total_time_return = row['Time L-B-F']

        cost_L_B_F = distance_L_B_F / fuel_efficiency * FUEL_PRICE
        cost_L_F = distance_L_F / fuel_efficiency * FUEL_PRICE

        time_constraint = "None"
        if used_sleepover + 1 <= available_sleepover:
            total_cost_stay = calculate_total_cost(cost_L_F, used_sleepover)

            if total_time_stay <= MAX_DRIVING_HOURS and total_time_stay + MIN_DAILY_REST <= 24 * 60:
                if total_cost_stay < cost_L_B_F:
                    decision = "Stay in Truck"
                    used_sleepover += 1
                    total_cost = total_cost_stay
                elif total_cost_stay > cost_L_B_F:
                    decision = "Return to Base"
                    total_cost = cost_L_B_F
                else:
                    decision = "Driver's Choice"
                    total_cost = total_cost_stay
            else:
                decision = "Return to Base"
                total_cost = cost_L_B_F
                time_constraint = "Violated"
        else:
            decision = "Return to Base"
            total_cost = cost_L_B_F
        results_day2.append({
            'Driver': driver,
            'Decision': decision,
            'Total Cost (€)': round(total_cost, 2),
            'Updated Used Sleepover': used_sleepover,
            'Time Constraint': time_constraint
        })
    results_day2_df = pd.DataFrame(results_day2)
    results_day2_df.to_csv("day2_resultsfinal.csv", index=False)
    print("Day 1 and Day 2 decisions saved to 'day1_resultsfinal.csv' and 'day2_resultsfinal.csv'")
```

Figure 24 Full code for final version(part2)