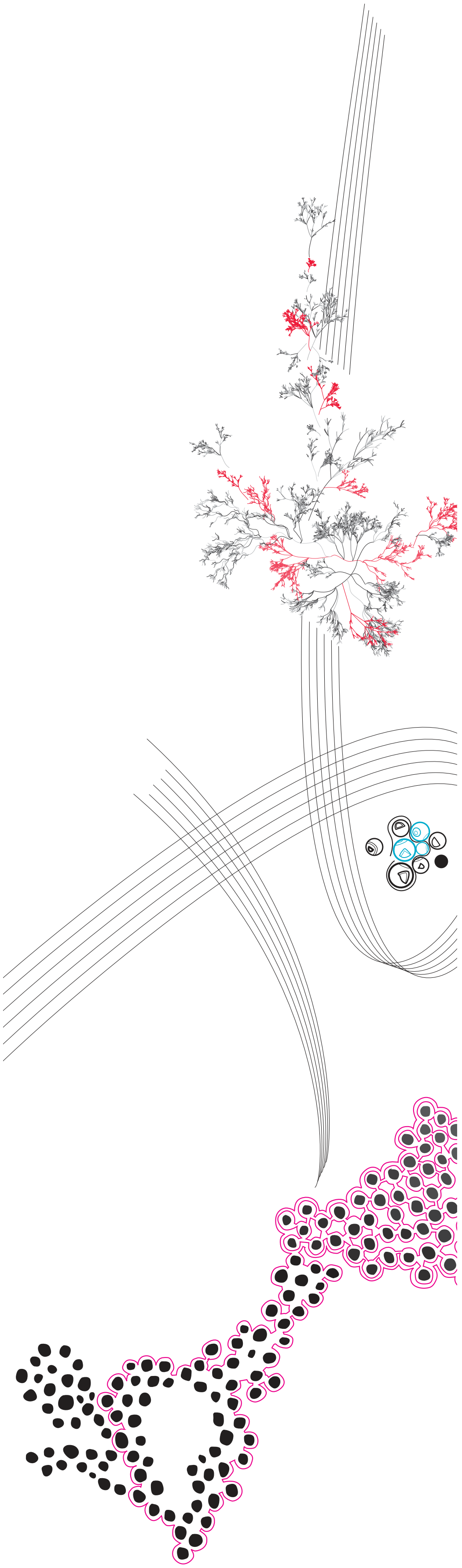
MSc Industrial Engineering
and Management
Graduation Thesis



# Deep reinforcement learning to support dynamic decision-making in a transport network amid travel- and handling time uncertainty

J.C. Kessels (Jasper)

Supervisor UT: prof.dr.ir. M.R.K. Mes (Martijn)
Daily Supervisor UT: ir. F.R. Akkerman (Fabian)
Supervisor BBI: N.G. Tijink (Niek)
Daily Supervisor BBI: S. Beeldman (Sander)

March, 2024

Faculty of Behavioral, Management
and Social sciences

UNIVERSITY OF TWENTE.

# Preface

This thesis is written as a graduation project for the master Industrial Engineering & Management, specialization in Production and Logistics Management, at the University of Twente and represents the end of my time as a student in Enschede.

Firstly, I would like to thank Bolk Business Improvement for allowing me to perform this thesis under their guidance. Special thanks to Sander Beeldman for his consistent involvement and support throughout the research process. His guidance and availability were important in navigating the daily challenges of the project. I also extend my appreciation to Niek Tijink for his oversight and assistance, in ensuring the smooth progress of the research during the more difficult moments.

I am thankful to Fabian Akkerman for his support during the challenges encountered in modeling and programming within DynaPlex. His assistance and feedback were also crucial in overcoming obstacles and refining the research methodology. Similarly, I appreciate Martijn Mes for his insightful feedback, which improved the overall quality of this research.

Finally, I would like to thank all my friends within “Magnus”, within the rowing association and my house for their support. As well as my family whom I could always give a call when I did not progress as I initially hoped. Last but not least, I want to thank my girlfriend for giving me the motivation to keep going and reminding me that there is a life besides graduating.

I hope you enjoy reading this thesis.

*Kind regards*,

*Jasper*
*Enschede, March 2024*

# Management Summary

This research has been conducted at Bolk Business Improvement (BBI) as part of the master's graduation assignment in Industrial Engineering and Management at the University of Twente. BBI is a consultancy company specializing in transport- and logistics projects. Additionally, this research is embedded in the DynaPlex project, which aims to develop a deep reinforcement learning toolbox to support data-driven logistics decision-making. Deep reinforcement learning is a research area within machine learning that combines reinforcement learning (RL) and deep learning.

## Problem Description

Gam Bakker, BBI's client, currently shares responsibility for all transport operations for Cargill within the Amsterdam area. Gam Bakker is proposing a new arrangement to Cargill, aiming to assume sole responsibility for all transport operations of Cargill's products in the Amsterdam area. This initiative seeks to establish an end-to-end (E2E) transport network, incorporating additional transport routes currently serviced by competitors of Gam Bakker. Transitioning towards an E2E transport network offers opportunities for better transport coordination, potentially leading to more consolidated transport flows and a reduced number of empty kilometers driven.

BBI anticipates complications in acquiring these additional transport routes. The main concern is that the route planning approach currently employed by Gam Bakker may lead to inefficient freight transportation within the proposed E2E transport network, thereby failing to leverage the advantages of having more control over transport operations.

The recent advancements in AI, particularly in deep reinforcement learning, hold promise for addressing such complex logistical challenges. Notably, Silver et al. [1] presented a deep reinforcement learning algorithm capable of high-level decision-making in complex and stochastic environments. Since then, there has been a growing body of literature focused on utilizing deep reinforcement learning in route optimization. BBI is interested in exploring the application and practical implementation of this technology. Therefore, the main research question addressed in this study is:

**How can deep reinforcement learning contribute to efficient route planning and truck scheduling in Gam Bakker's proposed transport network?**

To address this question, we started by modeling the end-to-end (E2E) transport network as an MDP. This mathematical framework allows for modeling decision-making scenarios in which outcomes are partly random and partly under the control of a decision-maker. We accessed the effectiveness of DRL within the route planning context by comparing the performance of an agent trained using DRL against a rule-based algorithm functioning as a benchmark. This comparison is done by using a simulator to generate trajectories of states, actions, and rewards. In this simulation, the trajectory of states can be seen as a realization of the transport operation of one day and the actions are taken by the decision-making agents. We proceeded to evaluate the performance of the two agents by analyzing key indicators, including the number of empty kilometers driven on days when transport activities occur, as well as the overall costs incurred during these operations. In short, we conducted a comparative analysis between the agents trained utilizing deep reinforcement learning and the rule-based algorithm specifically designed to mirror the currently used planning principles, thereby determining the effectiveness of DRL.

## Solution Methodology

After analyzing the current situation and performing a literature review, we proposed (i) to classify the problem and formulate the problem as an indefinite MDP, (ii) to validate the capability of neural networks to serve as decision-making agents using supervised learning and (iii) to compare the performance of neural network agents, trained using a deep reinforcement learning technique called deep controlled learning, against the performance of a rule-based algorithm based on current planning principles. These three phases of the solution methodology are further elaborated upon below.

### i. Problem formulation

Based on the problem description we identified the E2E transport network problem as a homogeneous capacitated multi-vehicle routing problem with pickup and delivery with time constraints and stochastic travel- and handling times. We formulated the transport network as an indefinite MDP and followed the sequential decision-making framework proposed by Powell [2]. This provided a mathematical framework for modeling decision-making scenarios where outcomes depend on stochastic factors. In this problem, the traveling- and handling times are the stochastic factors. Notably, reinforcement learning is capable of solving MDPs without explicit specifications of the transition probabilities. Furthermore, reinforcement learning can be combined with function approximators, such as neural networks, to address MDPs with a large number of states.

### ii. Validation neural network agents

Before we started training neural networks as agents using DRL, we aimed to validate whether neural networks can effectively serve as decision-making agents for this specific variant of the vehicle routing problem. Specifically, we sought to determine if the mathematical representation of the E2E transport network could be utilized as input for a neural network and if the neural network could adequately capture environmental conditions to generate logical output. To accomplish this, we sampled various states from the entire state space and labeled them using a rule-based algorithm. Subsequently, we trained multiple neural networks using supervised learning and assessed whether these networks could accurately replicate the output provided by the rule-based algorithm.

### iii. Simulation study

We trained various neural networks using a deep reinforcement learning technique called deep controlled learning, each neural network is trained on a specific problem instance based on historical data. Thereafter, we performed an extensive simulation study to evaluate the various routing solutions based on key performance indicators and general decision-making logic. Additionally, we investigated the influence of various cost parameters on the decision-making process of neural network agents trained using deep-controlled learning. Similarly, we examined the impact of varying the number of vehicles within the transport network on overall performance.

## Results

### i. Verification MDP implementation

We verified that the conceptual model was correctly implemented into the DynaPlex toolbox. We utilized automatic MDP unit testing to test large quantities of the state space. Similarly, we manually analyzed several states and state transitions to verify logical behavior within the E2E transport network. No illogical behavior was found.

### ii. Validation neural network agents

We generated a dataset containing 25000 state samples using a similar sampling technique employed in the first phase of deep controlled learning. We tested varying network architectures to find high accuracies, indicating well-trained neural network agents. The initial evaluation showed that the neural networks have an accuracy between 70% - 75%. Subsequently, we aimed to increase the performance of the supervised neural networks by employing data augmentation, hyperparameter tuning, and regularization on the best-performing neural network architecture. The accuracy did not improve despite these additional methods focused on increasing the performance. In summary, our findings indicate that the efficacy of supervised neural networks for dynamic decision-making in this E2E transport network is limited.

### iii. Simulation study

In all three problem instances, the neural network agents trained using the deep controlled learning algorithm (NNARL) seem to outperform the consolidation algorithm (CA). In Table 1, we show that the cost differences of the routing solutions of the NNARLs, compared to the consolidation algorithm, in the transport network with 3 vehicles, are respectively 8.0%, 6.8%, and 13.0% for problem instances 12 June 2023, 14 March 2023 and 28 August 2023. Averaging a cost difference of 9.1%. In Table 2, we show that the cost difference of the routing solutions of the NNARLs, compared to the consolidation algorithm, in the transport network with 5 vehicles are respectively 44.3%, 20.9%, and 56.1% for problem instances 12 June 2023, 14 March 2023 and 28 August 2023. Averaging a cost difference of 40.4%.

Table 1: Average costs and standard deviation for each real-sized problem instance using 3 vehicles utilizing two different decision-making policies. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** | **CA** | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 3320.1 ± 676.6 | 3610.5 ± 981.5 | 8.0% |
| 14 March 2023 | 5760.9 ± 2170.4 | 6184.3 ± 2260.2 | 6.8% |
| 28 August 2023 | 4224.7 ± 1131.4 | 4858.0 ± 1210.7 | 13.0% |

Table 2: Average costs and standard deviation for each real-sized problem instance using 5 vehicles utilizing two different decision-making policies. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** | **CA** | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 3772.0 ± 995.1 | 6767.8 ± 2230.3 | 44.3% |
| 14 March 2023 | 4967.4 ± 1331.8 | 6277.8 ± 1518.5 | 20.9% |
| 28 August 2023 | 4336.8 ± 1176.2 | 9877.6 ± 3209.7 | 56.1% |

The first key performance indicator for evaluation is the average number of empty kilometers driven within the routing solutions of the vehicle routing problem with pickup and delivery. In Table 3, we show that the relative reduction of empty kilometers under neural network agents trained using deep reinforcement learning compared to the consolidation algorithm are respectively 4.0%, 0.7%, and 21.0% for problem instances 12 June 2023, 14 March 2023 and 28 August 2023 in the configuration with 3 vehicles. Similarly, we show in Table 4, that these gaps are respectively 76.7%, 64.7%, and 83.9% for problem instances 12 June 2023, 14 March 2023 and 28 August 2023.

Table 3: Average number of empty kilometers for each policy and each real-sized problem instance using 3 vehicles. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** (km) | **CA** (km) | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 377.6 ± 49.5 | 393.5 ± 65.2 | 4.0% |
| 14 March 2023 | 443.5 ± 18.1 | 446.5 ± 17.4 | 0.7% |
| 28 August 2023 | 427.7 ± 25.7 | 541.4 ± 45.7 | 21.0% |

Table 4: Average number of empty kilometers for each policy and each real-sized problem instance using 5 vehicles. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** (km) | **CA** (km) | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 412.6 ± 18.6 | 1770.5 ± 356.6 | 76.7% |
| 14 March 2023 | 523.4 ± 17.7 | 1480.9 ± 415.7 | 64.7% |
| 28 August 2023 | 445.3 ± 16.9 | 2775.1 ± 420.7 | 83.9 % |

The second key performance indicator for evaluation is the average total waiting time within the routing solutions of the vehicle routing problem with pickup and delivery. In Table 5, we show that the relative reduction of total waiting time under neural network agents trained using deep reinforcement learning compared to the consolidation algorithm are respectively 3.0%, 0.1%, and 23.0% for problem instances 12 June 2023, 14 March 2023 and 28 August 2023 in the configuration with 3 vehicles. Similarly, we show in Table 6, that these gaps are respectively 86.0%, 75.0%, and 89.1% for problem instances 12 June 2023, 14 March 2023 and 28 August 2023.

Table 5: Average total waiting time summed over all vehicles for each policy and each real-sized problem instance using 3 vehicles. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** (min) | **CA** (min) | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 152.1 ± 36.7 | 156.8 ± 42.0 | 3.0% |
| 14 March 2023 | 153.0 ± 29.6 | 153.1 ± 29.6 | 0.1% |
| 28 August 2023 | 147.4 ± 32.4 | 191.3 ± 89.0 | 23.0% |

Table 6: Average total waiting time summed over all vehicles for each policy and each real-sized problem instance using 5 vehicles. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** (min) | **CA** (min) | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 142.4 ± 28.2 | 1019.8 ± 171.1 | 86.0% |
| 14 March 2023 | 189.1 ± 31.8 | 756.6 ± 193.3 | 75.0% |
| 28 August 2023 | 148.6 ± 30.1 | 1361.2 ± 195.2 | 89.1% |

## Conclusion

Conceptually and experimentally deep reinforcement learning has demonstrated its efficacy in training agents that contribute to efficient route planning within the model of Gam Bakker's proposed transport network. The numerical results show that the NNARLs are on average 9.1% more cost-efficient in routing problems with three available vehicles, and 40.4% more cost-efficient in routing problems with five available vehicles. We further conclude that the performance gap of the neural network agents compared to the rule-based planning algorithm is contingent on the number of vehicles employed in route planning. Therefore, while the rule-based algorithm demonstrates comparable route planning capability to the neural network agent in specific parameter configurations and problem instances, the deep controlled learning algorithm exhibits greater potential for generalizability.

Furthermore, there are several limitations regarding the problem formulation and the research design originating from causes like the scope of the research, the available research time, the lack of data representative of the proposed situation, and the overall simplifications made to model the E2E situation. We presented a list of limitations of the current model that led to suggestions for further theoretical research. Additionally, we made recommendations regarding the potential development of a planning tool based on the model presented in this thesis.

Theoretically, this research adds to the current body of literature and aligns with various recommendations outlined by leading papers in the field of routing problems regarding the utilization of DRL. This is the first research to discuss a homogeneous capacitated multi-vehicle routing problem with pickup and delivery, with time constraints, and stochastic travel- and handling times and combine this with training a neural network agent based on deep controlled learning.

Lastly, we provided BBI with recommendations about the possible implementation of this model in a planning tool and provided sufficient insight into the capabilities of AI, and specifically DRL, for application in the transportation sector.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| **ADP** | Approximate Dynamic Programming |
| **AI** | Artificial Intelligence |
| **API** | Application Programming Interface |
| **BBI** | Bolk Business Improvements |
| **CA** | Consolidation Algorithm |
| **DARP** | Dial-a-Ride Problem |
| **DCL** | Deep Controlled Learning |
| **DP** | Dynamic Programming |
| **DRL** | Deep Reinforcement Learning |
| **E2E** | End-to-End |
| **GB** | Gam Bakker |
| **HPC** | High Performance Cluster |
| **IRP** | Inventory Routing Problem |
| **KPI** | Key Performance Indicator |
| **MDP** | Markov Decision Process |
| **ML** | Machine Learning |
| **NN** | Neural Network |
| **NNARL** | Neural Network Agent trained using deep Reinforcement Learning |
| **PDP** | Pickup and Delivery Problem |
| **PP** | Production Plant |
| **PPO** | Proximal Policy Optimization |
| **RL** | Reinforcement Learning |
| **SARSA** | State-action-reward-state-action |
| **TH** | Terminal Hoogtij |
| **VRP** | Vehicle Routing Problem |
| **VRPPD** | Vehicle Routing Problem with Pickup and Delivery |
| **WH** | Warehouse Hoogtij |

# 1 Introduction

This chapter serves as an introductory framework, providing context for the identified problem, introducing the solution direction, and outlining the research motivation and design. In Section 1.1, we briefly provide background information and contextualize this research. In Section 1.2, we provide a brief description of the overarching project this research thesis is part of and the broader trend in the literature. In Section 1.3, we provide descriptions of the companies related to the research. In Section 1.4, we describe the research motivation and identify the problem. In Section 1.5 we describe the research design.

## 1.1 Background

This thesis revolves around the transportation operations conducted by Gam Bakker for its client, Cargill. Gam Bakker is a company specializing in logistical and transport services. Cargill is a company that manufactures chocolate products. Currently, various companies, including Gam Bakker, are involved in transporting freight between Cargill's production facility and different terminals and warehouses in Amsterdam. Gam Bakker is in the process of proposing a new arrangement, aiming to take sole responsibility for all transportation and storage of Cargill's products in the Amsterdam area. The objective is to establish an end-to-end (E2E) system, streamlining transport operations with the goal of cost and time savings. In this E2E system, Gam Bakker assumes complete responsibility for facilitating transport and warehousing for Cargill in the Amsterdam area, allowing Cargill to concentrate on its core business, while Gam Bakker oversees transportation and warehousing. The E2E system differs from the current situation where the responsibilities for transport and warehousing of Cargill's products are divided among multiple parties, who may have conflicting interests. This thesis focuses specifically on the transport network within this proposed E2E system, exploring efficient freight transportation across the Amsterdam area.

Currently, Gam Bakker's transportation operations are limited to moving freight between Hoogtij and Cargill's production facility. Hoogtij houses both a warehouse and a terminal of Gam Bakker, making transport planning between these locations relatively straightforward due to the limited number of possible decisions a planner can take. In contrast, the proposed E2E system introduces increased complexity, with a larger number of possible actions that can be taken at each moment a vehicle is available for transport. For instance, when facing a traffic jam between Hoogtij and the production facility, a decision can be made to perform another transport movement, postponing the trip between Hoogtij and the production facility to a more optimal time. Similarly, if a vehicle arrives at the production facility without available freight to transport back to Hoogtij, a decision in the E2E system can be made to satisfy a transport request to another terminal in Amsterdam for which the freight is ready. In contrast, in the current situation, this are no alternative transport requests, and either the vehicle has to wait until freight is ready to be transported back to Hoogtij or the vehicle has to drive back empty to Hoogtij.

In summary, transitioning towards an E2E system offers opportunities for better transport coordination, potentially leading to more consolidated transport flows and a reduced number of empty kilometers. However, with the increase in responsibility and the absorption of transport flows previously managed by other parties, a more intricate planning and scheduling process for the transportation of Cargill's products is required to leverage the expanded operation control. The exact configuration of the E2E system will be further discussed in Chapter 2.

Recognizing the increased complexity in planning and scheduling, Bolk Business Improvements (BBI), an advisory partner to Gam Bakker specializing in process optimization, has expressed interest in exploring a solution based on Deep Reinforcement Learning (DRL) to enhance transportation coordination in the proposed E2E system. BBI sees potential in this approach and wants to gain more practical experience utilizing DRL. This research aims to evaluate a DRL-based planning and scheduling approach focusing on optimizing the coordination of transportation activities for this specific use case.

## 1.2 Adoption of AI

In this section, we provide context to the adoption of AI. In Section 1.2.1, we specifically focus on the adoption of AI in the transport sector. In Section 1.2.2, we shortly elaborate upon the DynaPlex project.
Artificial Intelligence (AI) and Machine Learning (ML) algorithms increasingly contribute to different sectors of society, including finance, healthcare, and logistics [4]. In 2018, DHL, one of the largest logistical companies, and IBM, a large software corporation, published a report [5] on the development of AI, specifically tailored towards the logical sector. The report concluded that AI will transform the sector's approach to work into one that is more proactive, predictive, automated, and personalized.

In the same year, researchers at DeepMind published an article [1] presenting AlphaZero, an algorithm that was able to compete against established game engines in Chess, Go, and Shogi. AlphaZero was able to approximate effective actions in uncertain environments. Notably, AlphaZero achieved this performance without relying on hand-crafted features or evaluation functions, instead, it autonomously learned and improved without human intervention. Furthermore, the algorithm was very generalizable and could be used in many uncertain environments. This achievement marked a milestone in AI and exemplified its potential in any complex decision-making scenario.

The logistics sector recognizes the potential of this technology and aims to integrate AI into business processes for further business improvement. However, companies encounter challenges when attempting to translate the theoretical possibilities of AI into practical applications and leverage the technology effectively due to the required expertise.

### 1.2.1 Adoption of AI in transportation

A key goal driving research in the transportation domain is the realization of fully connected and autonomous mobility [6]. Consequently, there is a growing trend in the literature to reduce human involvement in decision-making processes and enhance AI capabilities.

The literature related to the implementation of AI in transportation has seen significant growth since the publication of Silver et al. [1], showcasing the potential of AI, particularly Deep Reinforcement Learning (DRL). Farazi et al. [6] conducted a systematic analysis of 150 studies related to applying DRL in transportation by the end of 2023. They identified and categorized seven diverse applications of DRL in transportation: (i) autonomous driving, (ii) adaptive traffic signal control, (iii) energy-efficient driving, (iv) maritime freight transportation, (v) route optimization, (vi) rail transportation, and (vii) traffic management systems.

Notably, the research presented in this thesis aligns with the category of route optimization outlined above. The application and adoption of DRL-based methods for routing optimization are demonstrated in varying use cases. For example, studies by Nazari et al. [7], Zhang et al. [8], and Chen et al. [9] investigate the effectiveness of DRL-based solutions for various multi-vehicle routing problems. Conversely, research conducted by Oda and Wong [10] and Singh et al. [11] focuses on taxi dispatching and optimizing routing decisions for the transportation of people. In summary, the group of route optimization papers explores challenges related to freight delivery or mobility services for the transportation of people.

An extensive literature review is performed in Chapter 3.

### 1.2.2 The DynaPlex project

As mentioned in Section 1.1, BBI has an interest in exploring the applicability DRL for transport coordination within Gam Bakker transport network, thereby aligning itself with its industry interests of utilizing AI.

Following the observation that the logistics industry had difficulties utilizing the theoretical capabilities of AI, researchers from Eindhoven University and the University of Twente aspired to make this technology more accessible and practicable. A project was started to develop a generic toolbox that could solve any sequential decision problem formulated as a Markov Decision Process (MDP). This toolbox is called DynaPlex [12]. DynaPlex serves as a versatile, generic solution to address various logistical challenges. It supports decision-making in planning, scheduling, allocation, and routing within transportation management, inventory management, and warehouse management and is based on DRL.

This research is embedded in the DynaPlex project, contributing to the ongoing development of the toolbox and validating its effectiveness, as well as assessing the capabilities of DRL itself for a use-case of the logistical industry.

## 1.3 Company Descriptions

This master's thesis is being conducted at Bolk Business Improvement (BBI), situated in Hengelo, the Netherlands. As already described in Section 1.1, there are multiple stakeholders in this research project besides BBI. We first discuss the relationships between the stakeholders in Section 1.3.1. Secondly, we discuss the stakeholders as individual entities in the subsequent subsections.

### 1.3.1 Stakeholders relationships

In Figure 1 the relationships between the stakeholders of this research project are visualized and described in Figure 1.

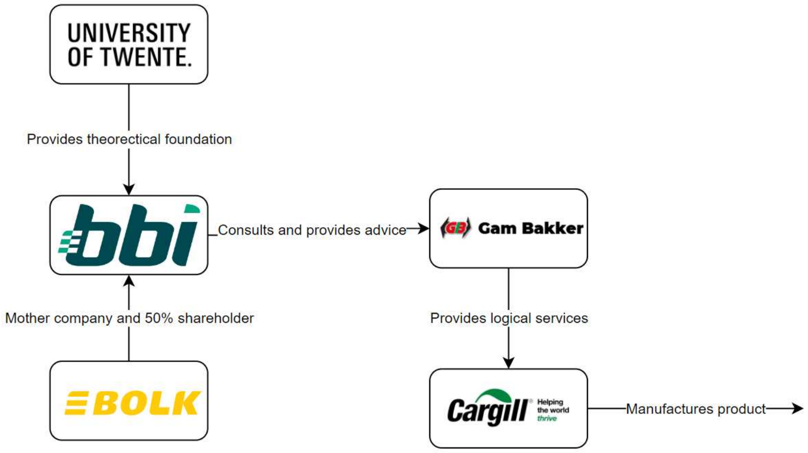

UNIVERSITY OF TWENTE.
Provides theorectical foundation
bbi
Consults and provides advice
GB Gam Bakker
Mother company and 50% shareholder
Provides logical services
BOLK
Cargill Helping the world thrive
Manufactures product

Figure 1: Organizational chart representing the relationships between stakeholders.

As described in Section 1.1, the problem centers around the transport network of a logistical provider and a production plant. The logistical provider is called Gam Bakker and the production plant is owned by Cargill Cocoa and Chocolate. The other three entities are indirectly involved.

Gam Bakker is a client of BBI. BBI serves as a consultant to Gam Bakker, assisting them in securing the transport tender from Cargill. This is not their first collaboration; BBI previously consulted Gam Bakker on another project, involving the construction of their new warehouse called Hoogtij.

Bolk Transport B.V. is the parent company of BBI. BBI operates with a relatively high degree of autonomy from Bolk, with the latter refraining from active participation or involvement in the specific details of BBI's projects. Bolk's focus does not extend to this research endeavor. Nonetheless, by allowing its subsidiary company to explore opportunities, it remains informed and engaged in the event of possibly impact-full technological developments.

The University of Twente is responsible for the theoretical development of the DynaPlex toolbox, as outlined in Section 1.2. This toolbox is designed to facilitate decision-making processes within the logistics sector. BBI has shown interest in integrating this toolbox into Gam Bakker's proposed transport network to enhance transport planning and scheduling.

### 1.3.2 Bolk Transport B.V.

Bolk Transport B.V. (Bolk) is a transportation company. Bolk is stable and highly innovative with diverse activities, clients, and collaborations [13]. The company engages in a wide range of activities. For example, Bolk offers both conventional and exceptional transport services. In recent years, Bolk has become internationally known for its specialization in the transportation of windmill turbines throughout the European continent. In addition to its core transportation services, Bolk also focuses on several other areas, including:

- Logistical services: activities related to warehousing.
- Container transport: the transportation of containers using different modal types.
- Consultancy services: providing advice both internal and external to improve business processes.

These consulting services are provided by their daughter company Bolk Business Improvement (BBI).

### 1.3.3 Bolk Business Improvement

Bolk Business Improvement (BBI) specializes in offering advice and developing (digital) tools for production and logistical services. Established in 2021 through a collaboration between two experienced process engineers and Bolk Transport B.V. (Bolk), BBI operates as a subsidiary of Bolk [14]. Conceptually, BBI serves as an internal consultancy firm embedded within the organizational framework of Bolk, aiming to enhance production- and logistical processes for Bolk. Furthermore, BBI has the flexibility to extend its consulting services to external companies facing similar logistical challenges as Bolk.

### 1.3.4 Gam Bakker

Gam Bakker is involved in offering transportation and logistical services to various clients, with Cargill being one of their prominent partners. Gam Bakker specializes in conditioned groupage transport for flowers, flower bulbs, plants, and seeds [15]. Their expertise also extends to fragile and refrigerated transport services. Gam Bakker has been growing and diversifying their offerings significantly. Notably, they have recently completed the construction of the Hoogtij warehouse, enhancing their capacity to serve clients more comprehensively through expanded warehousing capabilities.

### 1.3.5 Cargill Cocoa and Chocolate

Cargill Incorporation is an international corporation that trades, purchases, manufactures, and distributes agricultural commodities [16]. The upper management of the daughter company, Cargill Cocoa and Chocolate, claims that about 90% of all food in the Netherlands came into contact with the Cargill Incorporation.

Cargill Cocoa and Chocolate (Cargill), based in the Netherlands, specializes in the production of cocoa butter and cocoa press cakes derived from cocoa mass [16]. Their production facility is situated in Wormer, in the Amsterdam area. To meet the production demands, Cargill imports cocoa mass and additional cocoa butter from Africa via the ports of Rotterdam and Amsterdam. Subsequently, Cargill exports to various destinations worldwide, primarily within Europe, utilizing a combination of modal types for distribution.

## 1.4 Research Motivation

In this section, the research motivation and problem identification are discussed and elaborated upon. In Section 1.4.1, we elaborate upon the research motivation for each stakeholder. In Section 1.4.2 we identify and formulate the research motivation.

### 1.4.1 Motivation stakeholders

**Gam Bakker perspective**
Gam Bakker aims to secure Cargill's business by proposing a partnership wherein Gam Bakker assumes full responsibility for transporting Cargill's products in the Amsterdam area, as outlined in Section 1.1. With the acquisition of transport flows previously managed by other parties, the complexity of the transport network increases. Anticipating a need for a more sophisticated planning and scheduling approach, compared to the current planning and scheduling method, could enhance transport coordination effectiveness. Therefore, directly applying the current approach for planning and scheduling transport to the proposed E2E system is expected to underutilize the transport network's potential, missing opportunities for enhancing transportation efficiency.

**BBI perspective**
BBI has explicitly stated its interest in exploring an AI-based solution and the current capabilities of DRL. This demonstrates BBI's motivation not only to find an effective solution for this use case but also to better understand this preferred solution methodology. It is important to recognize that this preference limits the possible solution directions for the problem discussed in this thesis, which will be elaborated on in Section 1.4.2. Furthermore, it is worth noting that DRL may not always be the best choice for every route planning and scheduling problem. Its effectiveness depends on various factors. In some cases, traditional methods like dynamic programming or heuristics can be equally effective [17]. However, traditional scheduling methods often rely on predefined rules and assumptions, which may not hold in dynamic or uncertain situations. DRL models can adapt and learn from the environment, making them well-suited for making sequential decisions in uncertain conditions. A more detailed explanation of DRL and its theoretical applications is provided in chapter 3.

The remainder of this thesis is written with the understanding that the solution methodology will be based on DRL because BBI is specifically interested in exploring the capabilities of DRL and DynaPlex.

**Academic perspective**
As described in Section 1.2, the theoretical developments of AI and specifically DRL show potential to improve decision-making in uncertain environments. However, in many model instances, simple decision-making algorithms show similar performance. For this use case, it is interesting to research the effectiveness of DRL techniques relative to existing techniques commonly applied for planning and scheduling.

Furthermore, this research is embedded in the DynaPlex project. This research contributes to the ongoing development of the DynaPlex toolbox by possibly exposing practical challenges during the usage of the toolbox.

### 1.4.2 Problem Identification

In this section, we aim to identify and formulate the research problem. However, the methodology for problem identification deviates from the common practice in two aspects.

The research methodology, as outlined in Heerkens et al. [18], involves formulating and identifying a problem and subsequently searching for an appropriate solution approach. However, as discussed in Section 1.4, BBI's motivation is to gain insights into a solution based on DRL. Additionally, from an academic perspective, this research is embedded in the broader project of developing a generic toolbox for solving sequential decision-making problems. Consequently, a problem was identified that provides a sufficient framework for the validation of the preferred solution approach: deep reinforcement learning.

Secondly, the E2E transport network is currently a proposal and not a reality. It is crucial to acknowledge that we are identifying a problem for which there is no practical experience or a complete dataset available. It is essential to note that some degree of speculation should be taken into consideration.

**Current transport network - E2E transport network**

The exact differences between - and characteristics of the current situation and E2E situation will be explained in Chapter 2. To identify the problem it is key to understand that the proposed E2E transport network is responsible for transport requests between more locations relative to the current situation, including a warehouse in Amsterdam and a terminal in Amsterdam. This is visualized in Figures 2 and 3.

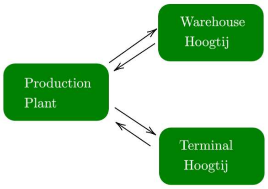


Figure 2: Schematic of Current Situation: Gam Bakker is responsible for transporting between three locations.

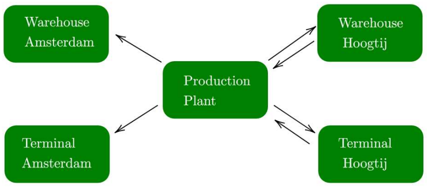


Figure 3: Schematic of E2E Situation: Gam Bakker is responsible for transporting between five locations.

In the current situation, the planning is static, with each vehicle receiving a predetermined sequence of transport requests at the beginning of the day that must be executed in order. The transport primarily takes place between Hoogtij and the production plant, as depicted in Figure 2. Consequently, in the current situation, there are no instances where alternative transport routes are considered during the execution of the transport requests because all transport takes place between Hoogtij and the production plant. In contrast, the proposed E2E transport network, illustrated in Figure 3, introduces transport requests that necessitate taking different routes. These different routes, present opportunities, as discussed in Section 1.1.

To illustrate these opportunities, consider a scenario where planners create static schedules for each vehicle in the E2E transport network. Suppose a specific vehicle is currently at the production plant, and its schedule dictates that it executes a transport request between the production plant and Hoogtij. Unfortunately, there is much traffic on the scheduled route from the production plant to Hoogtij, resulting in extended travel times, executing another transport request between the production plant and warehouse Amsterdam might be more efficient if the route is less congested. Therefore, if the deadlines for these orders allow for flexibility, rescheduling could be advantageous.

In essence, the core problem lies in Gam Bakker underutilizing the opportunities for efficient freight transport when it transitions towards the E2E transport network if the current planning approach is applied to the E2E transport network. Ideally, the planning team focuses more on dynamic decision-making rather than static decision-making.

**Problem cluster**

In accordance with Heerkens and van Winden's method for solving managerial problems [18] a problem cluster is constructed in Figure 4 to show cause-and-effect relationships between the action problem and the core problems. Furthermore, some dependent problems are discussed to better contextualize the problem.

It is essential to highlight that the problem cluster assumes that Gam Bakker has secured the transport tender and gained more operational control over Cargill's transport network. Despite this change in the situation, the existing method for route planning and truck scheduling is still in use in the problem cluster depicted in Figure 4.

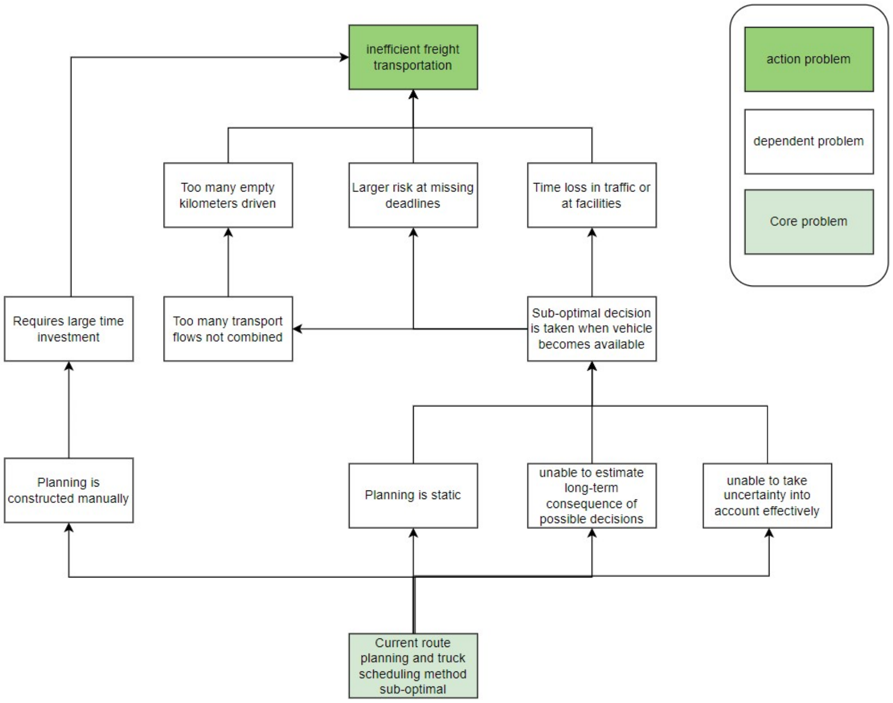

inefficient freight transportation
action problem
dependent problem
Core problem
Too many empty kilometers driven
Larger risk at missing deadlines
Time loss in traffic or at facilities
Requires large time investment
Too many transport flows not combined
Sub-optimal decision is taken when vehicle becomes available
Planning is constructed manually
Planning is static
unable to estimate long-term consequence of possible decisions
unable to take uncertainty into account effectively
Current route planning and truck scheduling method sub-optimal

Figure 4: Problem cluster overview.

**Core problem**

Adopting more operational control and employing the current planning and scheduling method introduces several interdependent problems, as illustrated in Figure 4. Generally, these issues stem from the heightened complexity of the E2E transport network compared to the current transport network. The solvable core problem is that the current route planning and truck scheduling method cannot be efficiently applied to the E2E transport network.

### 1.4.3 Research Problem

In accordance with Heerkens and van Winden [18] the research problem can be constructed from the problem identification in combination with the problem cluster. The research problem is formulated as follows:

**The current planning and scheduling method leads to inefficient freight transport in the proposed E2E transport network.**

Ideally, the DRL model can dynamically provide decision-making assistance and identify the best transport movement with the available information at that moment.

## 1.5 Research Design

As previously mentioned, this research framework is based on the methodology outlined by Heerkens and van Winden [18]. In Section 1.5.1, we explain the research goal and the desired results. In Section 1.5.2, we provide an overview of the research questions. In Section 1.5.3, we specify the approach to answer the research questions. Lastly, in Section 1.5.4, the scope of the research is explained.

### 1.5.1 Research Goal

The research goal is formulated in correspondence with the identified research problem discussed in Section 1.4.3, and is as follows:

**Gain insight into how deep reinforcement learning can contribute to efficient route planning and truck scheduling in Gam Bakker's proposed transport network.**

### 1.5.2 Research Questions

The main research question is derived from the stated research problem and research goal. The main research question focuses on solving the core problem and is defined as follows:

**How can deep reinforcement learning contribute to efficient route planning and truck scheduling in Gam Bakker's proposed transport network?**

Four different research questions and corresponding sub-questions have been designed to provide a more detailed breakdown of the main research question.

1. **What is the current situation of Gam Bakker's transport operations for Cargill?**
   (a) What characterizes the current transport network?
   (b) How many resources are used in the current transport network and how efficiently are they being used?
   (c) What are the differences between Gam Bakker's current transport network and the proposed E2E transport network?

2. **What solutions based on (deep) reinforcement learning have been proposed in the literature for similar transport network problems?**
   (a) What are the implementations of (Deep) Reinforcement Learning models in the literature for similar transport network problems?
   (b) What does this research contribute to the existing body of literature?
3. **How can we model the E2E transport network of Gam Bakker to evaluate the performance of deep reinforcement learning agents?**
   (a) How can Gam Bakker's sequential decision-making process be defined as a Markov Decision Process (MDP)?
   (b) How do we train agents utilizing deep reinforcement learning in the context of this problem?
4. **What experiments need to be performed to test the effectiveness of deep reinforcement learning techniques in the proposed E2E transport network model?**
   (a) Is the model correctly implemented in DynaPlex with respect to the conceptual model?
   (b) How effective are neural networks as decision-making agents in the proposed E2E transport network model?
   (c) How do we evaluate the performance of the neural network agents trained using deep reinforcement learning?
5. **What is the performance of the agents trained using deep reinforcement learning?**
   (a) How does the performance of a neural network agent compare against the performance of a rule-based algorithm within the E2E transport network?
   (b) What are the key differences between the actions recommended by the neural network agent and those taken by the rule-based algorithm?

### 1.5.3 Research Approach

The structure of the thesis adheres to a relatively straightforward framework. Each research question corresponds to a chapter.

In Chapter 2 the current situation is described. The focus here lies on context analysis of the current situation and the proposed E2E situation.

In Chapter 3 the literature is reviewed. The focus here lies on discovering how the research field has modeled and addressed similar problems. The goal of this chapter is to identify a gap in the literature. Thereby, further highlighting the academic relevance of this research project.

In Chapter 4 the design of the model is discussed. The focus here lies on the mathematical formulation of the E2E transport network.

In Chapter 5 we discuss the experimental design. The focus here lies on explaining the methodologies used to evaluate the performance and capabilities of DRL.

In Chapter 6 the performance of the model is assessed. The focus here lies on evaluating and presenting the numerical results.

In the last chapter the conclusions, both theoretical and practical contributions and both theoretical and practical recommendations are presented. The focus here lies on presenting the key findings and insights of this thesis and highlighting potential new research directions.

### 1.5.4 Scope

The implementation and integration of DRL models for operational-level route planning within a transport network is challenging. This challenge is particularly pronounced in the logistics sector, where the adoption of machine-learning techniques is still relatively limited. To ensure that the thesis is completed within a single academic semester and produces valuable and reliable results, it is important to define the scope of his thesis.

This research exclusively focuses on the scoped transport network of Gam Bakker in the Amsterdam area. Although Gam Bakker sometimes transports between other warehouses and terminals for Cargill, i.e. Rotterdam and Antwerp, these specific transport routes are excluded. Including them would introduce considerable complexity into the formulation of the sequential decision-making problem. Additionally, BBI's interest lies in a proof-of-concept, favoring a practical basic implementation over a potentially less efficient complex one to show the potential of applying DRL on problems within the logistics sector.

Moreover, this research focuses on operational-level route planning. As mentioned earlier, Gam Bakker and Cargill are exploring the potential for E2E logistics, and this thesis constitutes a component of this broader transportation tender. It is important to recognize that this thesis should refrain from focusing on challenges associated with the tactical-level aspects of the transportation network.

# 2 Current Situation

This chapter describes the current situation and functions as context analysis. In Section 2.1, we explain the current transport network in detail and analyze patterns and the performance of the current network. In Section 2.2, we elaborate upon the current planning procedure. Lastly, in Section 2.3, we compare the current situation and the proposed E2E situation.

## 2.1 Transport network

In this section we discuss the current transport network and key performance indicators. In Section 2.1.1, we describe the transport flows and the relevant geographical locations in detail. In Section 2.1.2, we describe limitations regarding the available data. In Section 2.1.3, we discuss the general performance metrics of the current transport network.

### 2.1.1 Description

Figure 5 illustrates the current flows of goods among Cargill's production plant, warehouse Hoogtij, terminal Hoogtij, warehouse Amsterdam, and terminal Amsterdam. This diagram shows inbound flows at both terminals. The production plant is responsible for manufacturing cocoa powder and cocoa butter from cocoa mass. In Figure 5, cocoa butter is represented as "CL Butter" cocoa mass as "Semi-Finished Products", and cocoa powder as "US Powder". All quantities of goods are measured in metric tons (MT). For instance, flow 1 corresponds to 85 thousand metric tons of cocoa mass, equivalent to 85 million kilograms of cocoa mass received at terminal Hoogtij per year. It is worth noting that this system does not account for outgoing flows of goods, and is purely focused on the flows around the Amsterdam area. Remark that Cargill also imports cocoa powder and cocoa butter. This is primarily due to the fact that Cargill sells more products than it can produce. Therefore, Cargill closes the gap between its production capacity and sales by importing additional quantities.

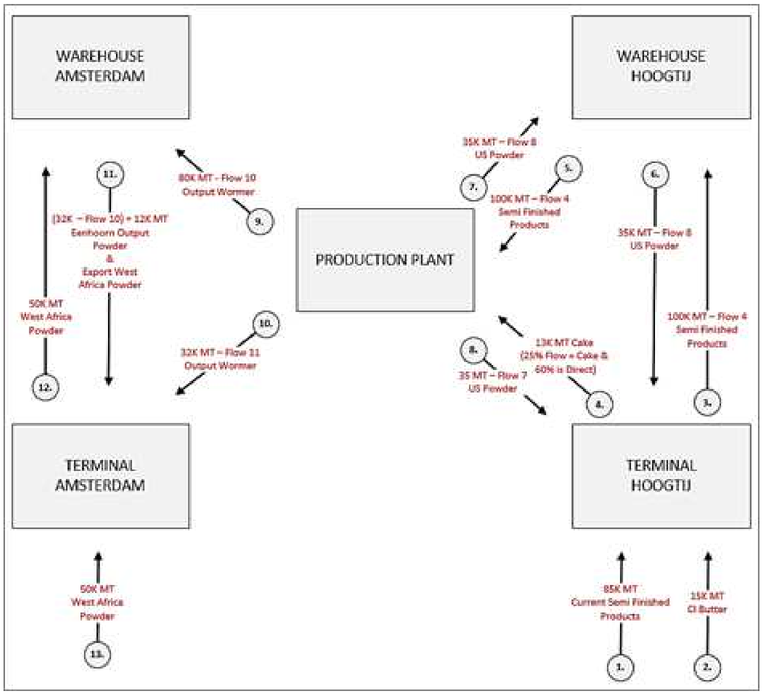

WAREHOUSE AMSTERDAM
WAREHOUSE HOOGTIJ
PRODUCTION PLANT
TERMINAL AMSTERDAM
TERMINAL HOOGTIJ
11.
(32K – Flow 10) + 12K MT Eenhoorn Output Powder & Export West Africa Powder
80K MT - Flow 10 Output Wormer
9.
50K MT West Africa Powder
12.
10.
32K MT – Flow 11 Output Wormer
35K MT – Flow 8 US Powder
7.
5.
100K MT – Flow 4 Semi Finished Products
6.
35K MT – Flow 8 US Powder
100K MT – Flow 4 Semi Finished Products
3.
8.
13K MT Cake (25% Flow = Cake & 60% is Direct)
35 MT – Flow 7 US Powder
4.
50K MT West Africa Powder
13.
85K MT Current Semi Finished Products
1.
15K MT Cl Butter
2.

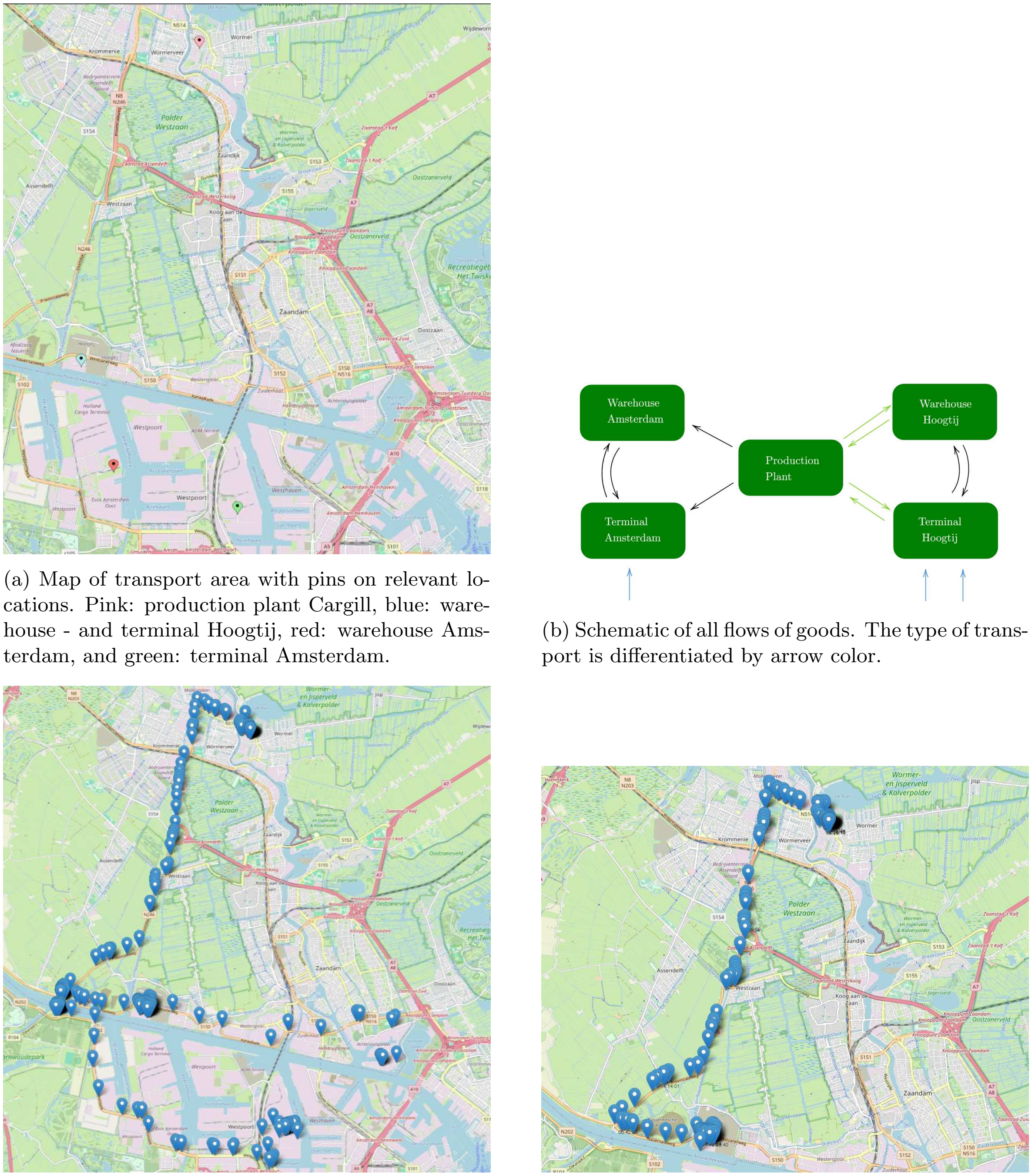
Figure 5: Schematic provided by BBI of all flow of goods and quantities transported for Cargill in the current situation by several transport companies.

Figure 6a shows the geographical map of the area in which the transport occurs for Cargill's product in Amsterdam. The pink pin represents the production plant, the light-blue pin represents Hoogtij, the red pin represents warehouse Amsterdam, and green pin represents terminal Amsterdam. Remark that Hoogtij is located north of the river and Amsterdam is south of the river. The transportation between warehouse Amsterdam, terminal Amsterdam, and Cargill generally requires the trucks to take the Coentunnel. In the current situation, the trucks that are responsible for transport between warehouse Amsterdam warehouse, terminal Amsterdam, and the production plant are from van Zandbergen, a competitor of Gam Bakker. In Figure 6a the Coentunnel is located in the bottom-right corner. Currently, the Coentunnel is one of the busiest highways in the Netherlands, averaging 159.400 vehicles per day in 2019 [19]. Logically, the Coentunnel heavily influences the traveling time between Amsterdam and Cargill in case of congestion.

Figure 6b similarly depicts all flows of goods as in Figure 5. However, this figure focuses on the type of transportation used per flow. The six arcs between the five nodes depict transport by truck. The curved arcs between warehouses and terminals represent transport done by terminal trucks. The light-blue arcs represent the incoming transport overseas. Lastly, the light-green arcs represent the current transport done by Gam Bakker for Cargill. It is important to recognize that in the current situation, van Zandbergen (competitor) is responsible for transportation between warehouse Amsterdam, terminal Amsterdam, and the production plant. In case the transport tender is won, Gam Bakker will also become responsible for these two transport movements.

Figure 6c presents a geographical map depicting transport movements and historical data from a specific truck, This truck, 77-BLN-2, was tracked on 14 September 2023. The location markers on the map represent data sampled from the truck's onboard computer. The onboard computer transmits its current location at differing intervals. On this day, the truck made two trips between Hoogtij and Hoogtij, and another trip between Hoogtij and Terminal Amsterdam. It's noteworthy that, on this occasion, the Coentunnel was not utilized, as the planning team opted to use the ferry.

Figure 6d presents a geographical map depicting the transport movements and location markers based on historical data from a specific truck. This truck, 14-BFB-6, was tracked on the date of July 20, 2023. The location markers on the map represent data sampled from the truck's onboard computer. The onboard computer transmits its current location at differing intervals. On the particular day in question, the truck made three round trips between warehouse Hoogtij and the production plant. Further explanation of onboard data will be done in 2.1.2.



(a) Map of transport area with pins on relevant locations. Pink: production plant Cargill, blue: warehouse - and terminal Hoogtij, red: warehouse Amsterdam, and green: terminal Amsterdam.

(b) Schematic of all flows of goods. The type of transport is differentiated by arrow color.

(c) Map of transport movements between Cargill's production plant and Gam Bakker's warehouse.

(d) Map of transport movements between Cargill's production plant and Gam Bakker's warehouse.

Figure 6: Four sub-figures detail the transport network on different levels and provide context to the above-mentioned schematic in Figure 5.

### 2.1.2 Data limitations

The data analysis serves two purposes for this research. Firstly, by gaining insights into the performance and behavior of the current transport network. In this case by asking the questions: how much handling time is spent waiting at depot, terminals, and production plant, how much time is required to travel between these locations, and how many empty kilometers are driven? Secondly, by finding relevant parameters for later usage in the model formulation.

Gam Bakker's available data can be divided into two categories. (1) Planning-related data generated using the Navitrans logistics software and (2) onboard data which tracks the movement of vehicles in real-time.

**Challenges of analysis**

The first challenge is based on the fact that transport movements for Cargill are not performed in a vacuum during the day in which transport takes place. Figures 7, 8, and 9 depict instances where a significant portion of the day was dedicated for transportation of freight for Cargill, while another portion of the day was dedicated to other clients. This means that while the scoped current situation seems relatively straightforward, the currently available onboard data is more complex and also contains transport operations for other clients. This means that optimizing the scoped situation could be beneficial but in actuality could compromise the total operation because it does not consider other clients.

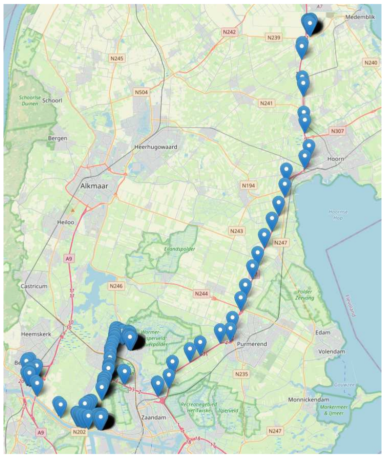


Figure 7: Location tracker 14-BFB-6 2023-06-07.

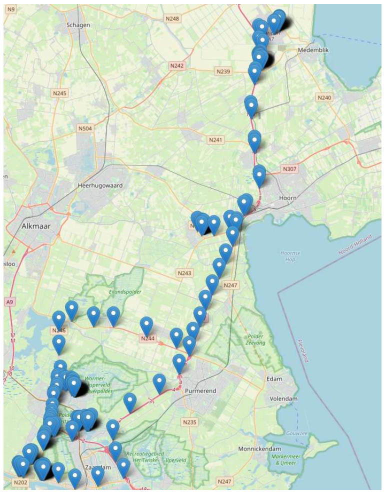


Figure 8: Location tracker 14-BFB-6 2023-09-04.

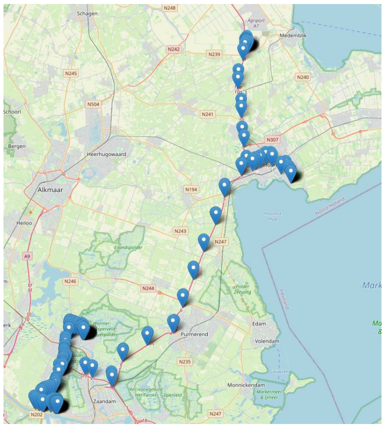


Figure 9: Location tracker 71-BGK-7 2022-08-24.

The second challenge stems from the fact that chauffeurs have different stations at which they start and end the working day based on their personnel contracts. This means that in some cases vehicles begin their day in Middenmeer, headquarters of Gam Bakker, and must travel to Hoogtij to initiate their tours for Cargill. This can also be seen in Figures 7, 8, and 9, where Middenmeer is located at the top-right and Cargill and bottom-left. In contrast to the chauffeurs stationed at Middenmeer, the chauffeurs stationed in Hoogtij do not have to travel this distance to start their operations. This means that a significant portion of empty drives is unavoidable in case the chauffeur is based in Middenmeer.

The last challenge is that the planning team only schedules and documents non-empty trips. This means that required empty drives between assignments are not documented in Navitrans. Consequently, the number of empty kilometers can only be deducted from the analysis of the onboard data. Large-scale analysis of the onboard data requires significant time investment because onboard data can only be analyzed per vehicle per individual day. This extensive analysis falls outside the scope of this thesis because well-defined and substantiated assumptions are sufficient for modeling the scoped transport network.

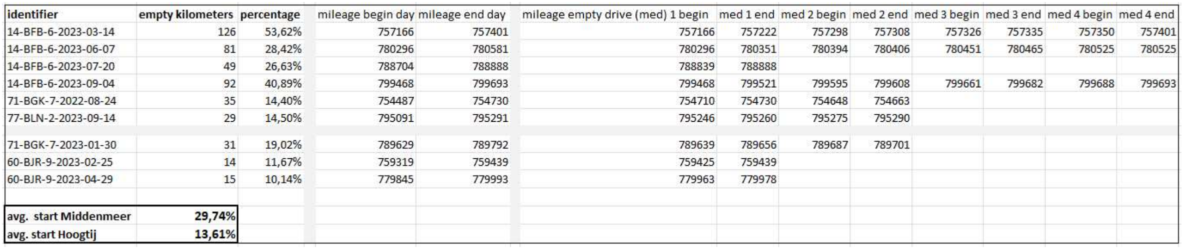
### 2.1.3 Performance

**Generic statistics**

To provide better insight into the number of orders Gam Bakker manages, it is helpful to start with simple statistics to contextualize the scale of the transport operations. These statistics are presented in Appendix F.

### Empty kilometers

The choice is made to perform a small analysis of several representative samples of onboard data containing days where vehicles primarily transported for Cargill. Onboard data contains activity information of the vehicle and the current mileage driven. The empty kilometers are deducted based on an analysis technique that determines whether a transport operation between nodes is with or without freight and uses the begin and end number of mileage for that transport movement to calculate the number of empty kilometers for the entire day. See Figure 10.



| identifier | empty kilometers | percentage | mileage begin day | mileage end day | mileage empty drive (med) 1 begin | med 1 end | med 2 begin | med 2 end | med 3 begin | med 3 end | med 4 begin | med 4 end |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 14-BFB-6-2023-03-14 | 126 | 53,62% | 757166 | 757401 | 757166 | 757222 | 757298 | 757308 | 757326 | 757335 | 757350 | 757401 |
| 14-BFB-6-2023-06-07 | 81 | 28,42% | 780296 | 780581 | 780296 | 780351 | 780394 | 780406 | 780451 | 780465 | 780525 | 780525 |
| 14-BFB-6-2023-07-20 | 49 | 26,63% | 788704 | 788888 | 788839 | 788888 | | | | | | |
| 14-BFB-6-2023-09-04 | 92 | 40,89% | 799468 | 799693 | 799468 | 799521 | 799595 | 799608 | 799661 | 799682 | 799688 | 799693 |
| 71-BGK-7-2022-08-24 | 35 | 14,40% | 754487 | 754730 | 754710 | 754730 | 754648 | 754663 | | | | |
| 77-BLN-2-2023-09-14 | 29 | 14,50% | 795091 | 795291 | 795246 | 795260 | 795275 | 795290 | | | | |
| 71-BGK-7-2023-01-30 | 31 | 19,02% | 789629 | 789792 | 789639 | 789656 | 789687 | 789701 | | | | |
| 60-BJR-9-2023-02-25 | 14 | 11,67% | 759319 | 759439 | 759425 | 759439 | | | | | | |
| 60-BJR-9-2023-04-29 | 15 | 10,14% | 779845 | 779993 | 779963 | 779978 | | | | | | |
| avg. start Middenmeer | 29,74% | | | | | | | | | | | |
| avg. start Hoogtij | 13,61% | | | | | | | | | | | |

Figure 10: Empty kilometers calculation based on activity status and mileage from the onboard data.

In figure 10, a distinction is made between two groups of vehicles: those based at Middenmeer (the top six) and those based at Hoogtij (the bottom three). All analyzed vehicles are primarily responsible for fulfilling Cargill's orders for the day. On average, each vehicle handles eight Cargill orders and one or two for other clients. For vehicles based at Middenmeer, they drive an approximate average of 29.74% of their total kilometers empty, while for vehicles based at Hoogtij, the average approximate number of empty kilometers is 13.61%.

Additionally, another critical indicator related to empty drives is the reuse percentage. As the name suggests, the reuse percentage reflects whether an arriving vehicle at Cargill with inbound cargo is subsequently used to transport outbound cargo to warehouses or terminals. In an ideal scenario, the reuse percentage would be 100%, signifying that all incoming vehicles are effectively repurposed to transport outgoing cargo, eliminating empty drives for Cargill. In Appendix F, we show data related to the reuse percentage.

### Transport times

The transport times between various locations have been determined through an analysis of the onboard data. We have sufficient data to calculate transport times between Hoogtij and Cargill, as presented in table 7. However, when it comes to travel times between Cargill and both warehouse Amsterdam and terminal Amsterdam, the onboard data is quite limited. This limitation arises because Gam Bakker's vehicles rarely operate between these locations.

In table 7, the average travel times between different locations are displayed, derived from onboard data. The values with asterisks denote approximations since they are based on limited onboard data collected during exceptional instances when Gam Bakker's vehicles traveled between warehouse Amsterdam and the production plant. In the context of this study, these values are considered as estimates, serving as a reference for subsequent modeling purposes. Furthermore, the last column in the table provides the travel times between these locations based on Google Maps. It is worth mentioning that the onboard data shows that the trucks travel at a slower pace compared to the estimated times provided by Google Maps. This difference can be explained by the fact that larger vehicles typically have slower travel speeds than regular cars on which the times in Google Maps are based.

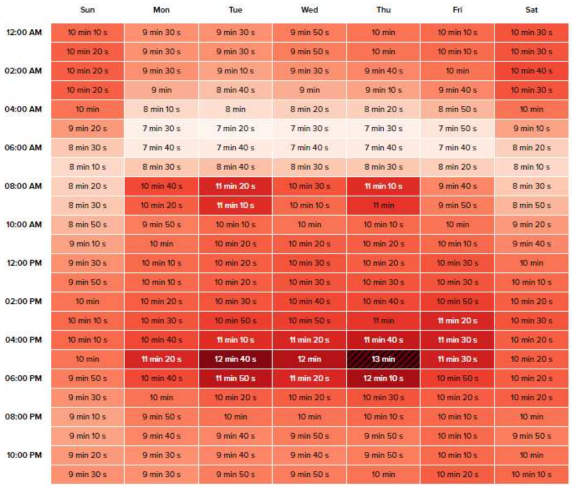
In addition, these projected travel times maintain a consistent ratio between journeys from Hoogtij to the production plant, similar to the approximate travel times derived from the available data. This consistency adds a degree of validity to the transport time approximations between the production plant and warehouse Amsterdam.

Publicly accessible data regarding traffic conditions in the Westzaan, Amsterdam region can also help illustrate and provide insight into the traffic conditions and seasonality throughout the week. Another

| Origin | Destination | Travel time ($min$). | Std duration ($min$) | Time Google ($min$) |
|---|---|---|---|---|
| PP | Hoogtij | 19.02 | 1.44 | 16 |
| Hoogtij | PP | 19.07 | 2.24 | 16 |
| PP | WA | 30.57* | - | 25 |
| WA | PP | 29.11* | - | 24 |
| PP | TA | - | - | 24 |
| TA | PP | - | - | 23 |

Table 7: Comparison of travel times based onboard data and mapping service. Travel times between Amsterdam and the production plant are based on the fact that the Coentunnel is the standard transport route.

company, specializing in location technology and GPS systems, released traffic reports that included traffic density data for each hour and day. While these reports do not provide specific travel times between the locations of the transport network, they do offer a more generalized overview of the entire region, which is valuable for accurately depicting daily and weekly traffic patterns and average travel times over a ten-kilometer distance. Figure 11 shows logical daily patterns. The daily commute of workers seems to be the most influential on the average travel times. The relative differences between hours during the day can be used to model travel times and their time dependency.



| | Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|---|
| 12:00 AM | 10 min 10 s | 9 min 30 s | 9 min 30 s | 9 min 50 s | 10 min | 10 min 10 s | 10 min 30 s |
| | 10 min 20 s | 9 min 30 s | 9 min 30 s | 9 min 50 s | 10 min | 10 min 10 s | 10 min 30 s |
| 02:00 AM | 10 min 20 s | 9 min 30 s | 9 min 10 s | 9 min 30 s | 9 min 40 s | 10 min | 10 min 40 s |
| | 10 min 20 s | 9 min | 8 min 40 s | 9 min | 9 min 10 s | 9 min 40 s | 10 min 30 s |
| 04:00 AM | 10 min | 8 min 10 s | 8 min | 8 min 20 s | 8 min 20 s | 8 min 50 s | 10 min |
| | 9 min 20 s | 7 min 30 s | 7 min 20 s | 7 min 30 s | 7 min 30 s | 7 min 50 s | 9 min 10 s |
| 06:00 AM | 8 min 30 s | 7 min 40 s | 7 min 40 s | 7 min 40 s | 7 min 40 s | 7 min 40 s | 8 min 20 s |
| | 8 min 10 s | 8 min 30 s | 8 min 40 s | 8 min 30 s | 8 min 30 s | 8 min 20 s | 8 min 10 s |
| 08:00 AM | 8 min 20 s | 10 min 40 s | 11 min 20 s | 10 min 30 s | 11 min 10 s | 9 min 40 s | 8 min 30 s |
| | 8 min 30 s | 10 min 20 s | 11 min 10 s | 10 min 10 s | 11 min | 9 min 50 s | 8 min 50 s |
| 10:00 AM | 8 min 50 s | 9 min 50 s | 10 min 10 s | 10 min | 10 min 10 s | 10 min | 9 min 20 s |
| | 9 min 10 s | 10 min | 10 min 20 s | 10 min 20 s | 10 min 20 s | 10 min 10 s | 9 min 40 s |
| 12:00 PM | 9 min 30 s | 10 min 10 s | 10 min 20 s | 10 min 30 s | 10 min 20 s | 10 min 30 s | 10 min |
| | 9 min 50 s | 10 min 10 s | 10 min 20 s | 10 min 30 s | 10 min 30 s | 10 min 30 s | 10 min 10 s |
| 02:00 PM | 10 min | 10 min 20 s | 10 min 30 s | 10 min 40 s | 10 min 40 s | 10 min 50 s | 10 min 20 s |
| | 10 min 10 s | 10 min 30 s | 10 min 50 s | 10 min 50 s | 11 min | 11 min 20 s | 10 min 30 s |
| 04:00 PM | 10 min 10 s | 10 min 40 s | 11 min 10 s | 11 min 20 s | 11 min 40 s | 11 min 30 s | 10 min 20 s |
| | 10 min | 11 min 20 s | 12 min 40 s | 12 min | 13 min | 11 min 30 s | 10 min 20 s |
| 06:00 PM | 9 min 50 s | 10 min 40 s | 11 min 50 s | 11 min 20 s | 12 min 10 s | 10 min 50 s | 10 min 20 s |
| | 9 min 30 s | 10 min | 10 min 20 s | 10 min 20 s | 10 min 30 s | 10 min 20 s | 10 min 20 s |
| 08:00 PM | 9 min 10 s | 9 min 50 s | 10 min | 10 min | 10 min 10 s | 10 min 10 s | 10 min |
| | 9 min 10 s | 9 min 40 s | 9 min 40 s | 9 min 50 s | 9 min 50 s | 10 min 10 s | 9 min 50 s |
| 10:00 PM | 9 min 20 s | 9 min 30 s | 9 min 40 s | 9 min 40 s | 9 min 50 s | 10 min 10 s | 10 min |
| | 9 min 30 s | 9 min 30 s | 9 min 50 s | 9 min 50 s | 10 min | 10 min 20 s | 10 min 10 s |

Figure 11: Average travel times in Amsterdam metropolitan area over ten kilometers for each hour and day.

**Handling times**

The handling time, also referred to as the turnaround time, represents the duration required for loading and/or unloading a vehicle. Similar to the travel times, the handling times for different locations can be determined through the analysis of onboard data. There is sufficient data available to compute handling times at Hoogtij and Cargill. However, it is important to note that there is a lack of data concerning the Amsterdam warehouse and terminal. Table 8 presents the average handling times, with the noteworthy observation that performing a single activity takes half as much time as executing both unloading and loading tasks.

## 2.2 Current Planning and Scheduling Approach

Gam Bakker's planning relies exclusively on client orders, operating on a pull system basis. This implies that orders are considered for transport only upon receiving an order from a client. Gam Bakker has an agreement with its clients stipulating that orders for the following day must be placed before 14:00.

| Location | Average handling time ($min$) | Activity |
|---|---|---|
| Cargill | 20:03 | Load or Unload |
| Cargill | 41:08 | Load and Unload |
| Hoogtij | 29:52 | Load or Unload |
| Hoogtij | 47:48 | Load and Unload |

Table 8: The handling times in the current transport network. Handling time is split based on whether the vehicle is only loaded or unloaded or both.

The planning team at Gam Bakker follows a one-day-ahead planning and scheduling approach for their operations. In practice, this means that between 14:00 and 17:00 the planning is made for the next workday. The objective is to create a truck schedule for the following day based on the current list of orders and the availability of trucks. The schedule is uploaded to Navitrans for the chauffeurs, providing the chauffeurs with the sequence of orders at the starting time of their workday. Consequently, the chauffeurs execute the orders in the sequence provided to them via Navitrans.

In the current situation, creating a routing plan satisfying Cargill requested transport movements is relatively easy. This is because 75.9% of transport is between requests of Cargill are between Hoogtij and the production plant. Shuttling between two locations is efficient because of the high reuse potential. There are already arrangements between Cargill and Gam Bakker to stimulate the reuse of vehicles that deliver cargo to the production plant. Cargill aims to have outbound cargo ready to transport back to the warehouse Hoogtij at the same time of product delivery, thereby reusing the vehicle that was just delivered.

Modifications to the schedule during the day are rarely performed and typically only occur in response to feedback from proactive chauffeurs who identify more efficient options. For example, the chauffeur notices that the queue at the production plant is large while another client does not have a queue, by executing the order for Cargill at a later moment when the queue is shortened the chauffeur spends his time more efficiently. Generally, the schedule remains fixed (static) once established and undergoes minimal modification during the day. As mentioned in the problem identification this is one of the problems that lead to inefficient route planning and truck scheduling because this approach does not consider and reacts minimally to transport network disruptions.

Lastly, Gam Bakker's planners do not schedule breaks and mandatory stops for vehicles that primarily travel between the production plant and Hoogtij during the day. Given that the handling times are typically equal to or longer than the travel times, the required downtime during loading and unloading operations adequately accounts for mandatory stopping times throughout the day. Moreover, there is a strong motivation for continuous work because once all the daily orders are completed, the drivers have finished their working day.

## 2.3 Transport Tender and the proposed End-To-End Situation

In the broadest sense, a transport tender represents a structured approach for selecting transportation services from a specific provider, considering factors such as cost, service quality, reliability, availability, and other relevant criteria important for the company seeking transport services. It can be likened to a competitive bidding process.

In this use-case Cargill has set up a transport tender and Gam Bakker's objective is to secure more of Cargill's business operations by proposing an End-To-End (E2E) system in which Gam Bakker assumes responsibility for transportation and warehousing activities in the Amsterdam area. Transitioning to an E2E model has the potential to increase Gam Bakker's efficiency and reduce total costs. As previously mentioned the efficiency increase is expected to come from the increased possibilities for transport coordination and warehouse coordination. This thesis specifically focuses on the increase in transport coordination.

In the E2E situation, the concept of increased transport coordination translates into the consolidation of transport movements. Six pairs of orders have been identified for potential consolidation, thereby facilitating more efficient vehicle dispatching and coordination. The combinations of these orders are visually represented in Figure 12. Each pair consists of two transport movements that Cargill routinely orders. The strategic merging of these individual orders leads to a reduction in empty kilometers, both to and from the production plant.

In the current situation the pink, dark blue, and yellow combinations in Figure 12 are not possible to execute. This restriction arises because Gam Bakker's competitor is currently executing orders between the production plant and warehouse Amsterdam or terminal Amsterdam.

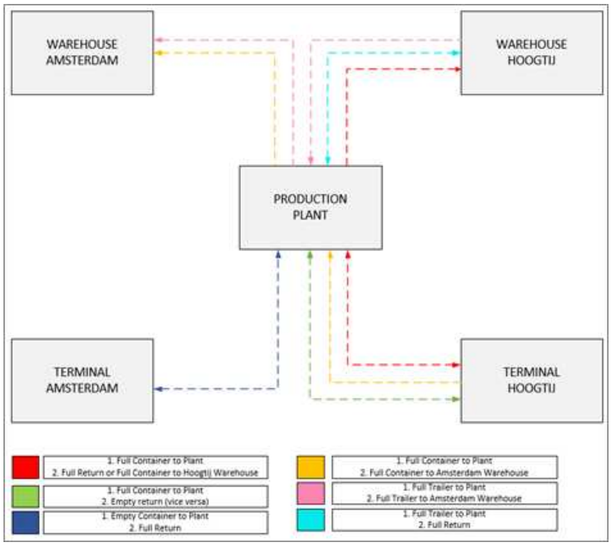

WAREHOUSE AMSTERDAM
WAREHOUSE HOOGTIJ
PRODUCTION PLANT
TERMINAL AMSTERDAM
TERMINAL HOOGTIJ
1. Full Container to Plant
2. Full Return or Full Container to Hoogtij Warehouse
1. Full Container to Plant
2. Empty return (vice versa)
1. Empty Container to Plant
2. Full Return
1. Full Container to Plant
2. Full Container to Amsterdam Warehouse
1. Full Trailer to Plant
2. Full Trailer to Amsterdam Warehouse
1. Full Trailer to Plant
2. Full Return

Figure 12: Schematic visualizing of the six possible transport combinations in the envisioned E2E network. Schematic provided by BBI.

## 2.4 Conclusion

In this chapter, the research question "*What is the current situation of Gam Bakker's transport operations for Cargill?*" is answered. We described the transport operations performed for Cargill in the Amsterdam area and presented schematics illustrating the transport flows and general statistics regarding the number of empty kilometers, traveling times, and handling times in the current situation. We discussed the current planning and scheduling approach and highlighted that Gam Bakker operates a pull system and currently utilizes a static planning approach. Lastly, highlighted the fact that we are not concerned with solving and improving the transport operations in the current situation. This research is focused on the decision-making in the hypothetical E2E transport network.

# 3 Literature

In this chapter we discuss the literature related to the deep reinforcement learning in route optimization. In Section 3.1 we introduce the vehicle routing problem and the specific variant relevant to this research. In Section 3.2 we elaborate upon the historical context of the routing problem relevant for this research and review the literature in which (deep) reinforcement learning is applied in the context of route optimization. Lastly, in Section 3.3 we conclude this chapter and answer its respective sub-question introduced in Chapter 1.

## 3.1 Background

This section aims to provide background information relevant to this research. In Section 3.1.1, we introduce the vehicle routing problem and a classification system for its variants. In Section 3.1.2, we classify the proposed E2E transport network following the framework, based on the information provided in Chapter 2. In Section 3.1.3, we discuss the literature related to classified variant of the vehicle routing problem.

### 3.1.1 Vehicle Routing Problems

The Vehicle Routing Problem (VRP) is a combinatorial optimization problem first introduced by Dantzig et al. [20] Its objective is to determine the most efficient routes for a fleet of vehicles, ensuring that all customers are visited, and that the vehicles start and end at the specified depot. Konstantakopoulos et al. [21] reviewed the developments in the VRP research area and stated that since its formulation this research literature area has been rapidly growing. VRPs and all its variants find a lot of practical applications for real-world problems and are important for supply chain operations where the model aims to optimize both distribution costs and customer satisfaction. Furthermore, because of its relevance an applicability in different scenarios a lot of different variants VRP have been developed. Ojeda Rios et al. [3] present the taxonomy and classification elements of a large set of common VRP variants. These classification framework is presented in Figure 13

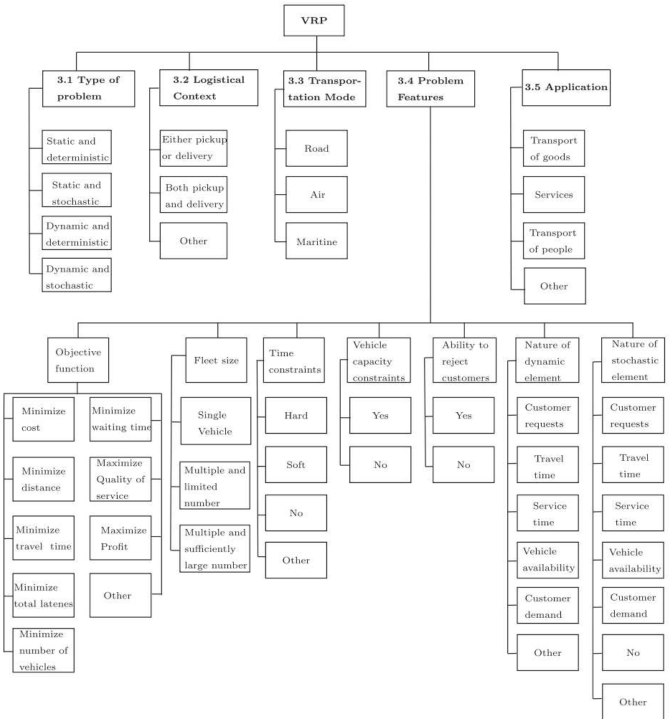


Figure 13: Classification framwework of vehicle routing problems [3].

Finding the optimal solution for the VRP is NP-hard, therefore the VRP cannot be solved to optimality in polynomial time [22]. A lot of research is centered around the development of algorithms to solve various VRP variants [21]. The most important distinction between solution methodologies is whether the algorithm produces an exact or approximate solution. The advantage of approximate methods is that they limit computational complexity. As a general rule, as more problem features are introduced or the scale of the VRP expands, the computational demands tend to increase. Consequently, in such scenarios, exact methods may become infeasible. Therefore, given the continuous development of the vehicle routing area, characterized by the incorporation of additional problem features, the development of approximate solution techniques becomes more relevant. In Figure 14 different solution methodologies are categorized.

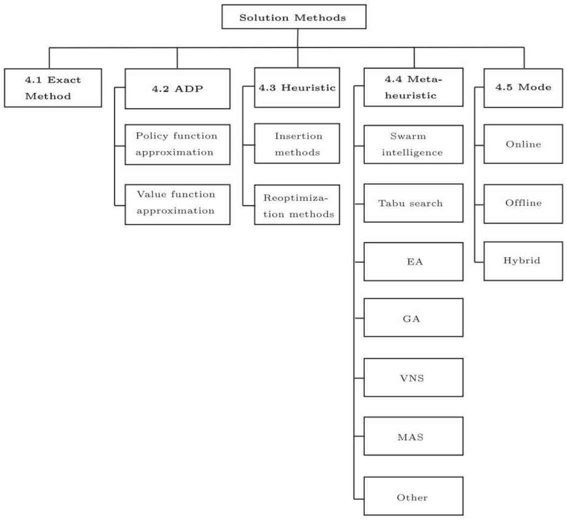


Figure 14: Classification framework of the vehicle routing problem solution methods [3].

### 3.1.2 Contextualization research problem

In this section, our objective is to contextualize the research problem within the existing body of literature by classifying the E2E transport network according to the descriptions provided in Chapter 2 and the elements delineated in Figure 13.

(i) **Type of problem**: dynamic and stochastic. Each route segment between locations will have varying travel times that change throughout the day. Similarly, handling times are also dynamic and stochastic.

(ii) **Logistical context**: both pickup and delivery. Gam Bakker will operate in a pull-based system where clients' transport requests specify both pickup and delivery locations.

(iii) **Transportation mode**: road. The E2E transport network will exclusively consider road transportation.

(iv) **Application**: transport of goods. In the case of the E2E transport network, Gam Bakker will be dealing with the transportation of goods, specifically chocolate products.

(v) The problem features are quite comprehensive. We list them as follows:

- **Objective function**: The Gam Bakker planning team will have multiple objectives. The most important objectives are to minimize costs, waiting time, total lateness, and empty kilometers.
- **Fleet size**: Gam Bakker will employ a variable number of vehicles each day to satisfy transportation requests.
- **Time constraint**: Gam Bakker is allowed to arrive past the deadline, but this incurs negative operational consequences for the customers.

- **Vehicle capacity constraint**: The vehicles utilized by Gam Bakker have a limited capacity.
- **Ability to reject customers**: In the context of the proposed E2E transport network, customers can be seen as transport requests. Gam Bakker does not have the capability to reject transportation requests.

Following the classification methodology outlined by Ojeda Rios et al. [3], the most logical option is to classify the routing problem, based on the elements discussed above, as a vehicle routing problem with pickup and delivery, with soft time constraints, stochastic and dynamic travel- and handling times, and a capacitated homogeneous fleet.

The goal of this section is to contextualize the proposed E2E transport network problem within the body of research related to vehicle routing problems. Remark that in Chapter 4 we mathematically define the E2E transport network and make simplifications in the formulation for effective modeling.

### 3.1.3 Vehicle Routing Problem with Pickup and Delivery

The vehicle routing problem with pickup and delivery (VRPPD) is a variant of the VRP that considers finding the optimal route and schedule to satisfy a set of transportation requests between two locations with a vehicle fleet and minimize operational costs or maximize a certain performance statistic [23]. This variant of the VRP introduces a more complex scenario where each transportation request involves both a pickup and a delivery location, and these locations have a precedence relationship, meaning that the pickup must occur before the delivery. In Figure 15 an example of the VRPPD is presented.

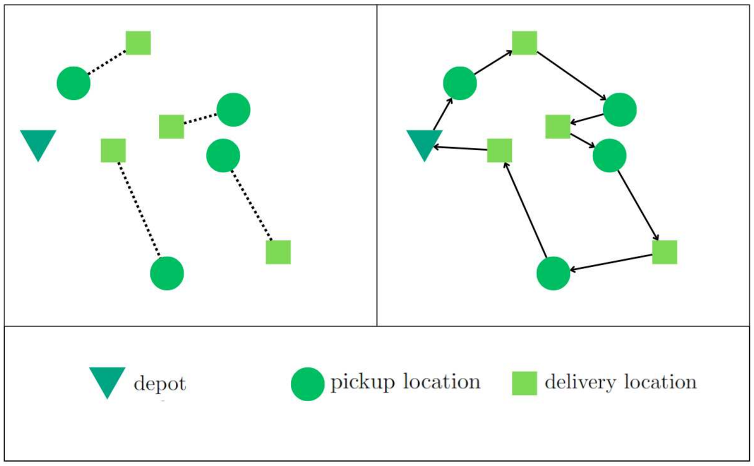


Figure 15: Illustration of the vehicle routing problem with pickup and delivery. On the left, is an overview of four transport requests. On the right, is the solution that minimizes travel distance and satisfies all four transport requests.

Furthermore, there are two problems closely related to the VRPPD: the vehicle routing problem with simultaneous pickup and delivery (VRPSPD) and the dial-a-ride problem (DARP).

**Vehicle routing problem with simultaneous pickup and delivery**
In the VRPSPD, a fleet of vehicles serves multiple customers, each with delivery and pickup demands. All delivery items originate from and all pickup items are destined for the depot. This means that the VRPSPD is mainly concerned with respecting the vehicle capacity and preventing preemptively loading to much freight before delivery. The VRPSPD is a more commonly researched problem in the literature compared to the VRPPD. Even though the names of the VRPPD and VRPSPD are similar, it is crucial to distinguish between them. To illustrate: in the VRPPD, customers typically request transportation from a pickup location *a* to a delivery location *b*, with distinct pickup and delivery locations specified as presented in Figure 15. In contrast, in the VRPSPD, customers request

a quantity to be picked up at location $a$ and a quantity to be delivered at the same location $a$.

**Dial-a-ride problem**
The DARP considers producing routes and schedules for persons who specify pickup and delivery requests between origins and destinations. The goal is to design a route that satisfies all requests respecting a set of ride-sharing and time constraints while maximizing a certain performance statistic. The DARP is specifically focused on the transportation of people. One of the most common applications is found in door-to-door transportation for elderly or disabled people [24]. In contrast to the inherent modeling differences between VRPPD and VRPSPD, the formulation of the VRPPD and the DARP is almost identical, with mainly the nature of the requests differing: transportation of people instead of transportation of goods.

## 3.2 Literature review

In this section we examine the literature related to the application of (deep) reinforcement learning in the field of route optimization. We specifically elaborate upon the literature related to solving the VRPPD and its variants. In Section 3.2.1 we provide insights into the historical development of these problems and specifically discuss the solution methodologies employed to solve the VRPPDs, VRPSPDs, and DARPs. In Section 3.2.2 we present the recent advancements in applying (deep) reinforcement learning in route optimization. Lastly, in Section 3.2.3 we elaborate upon the contributions of this research to the literature.

### 3.2.1 Historical development of VRPPD, VRPSPD, and DARP

This section is structured into three distinct periods to provide context for the historical development of these problems. We discuss (i) the time of their inception, (ii) literature over a decade old to highlight the complexity of the problem formulation and solution methodologies during that period, and (iii) recent advancements made in the context of these problems. For clarification, in this literature review the decision is made that advancements are considered recent if they are less than a decade old.

**Introduction problems**
The first papers proposing the vehicle routing problem with simultaneous pickup and delivery (VRPSPD) and the dial-a-ride problem (DARP) were Wilson et al. [1] [25] (1971) and Min [26] (1989). Wilson et al. proposed a single-vehicle uncapacitated model where transport requests between the pickup location and destination of passengers were known beforehand, this was the first version of the DARP. Min was the first to recognize that the traditional VRP in freight transport is considered a pure delivery or pickup problem. In many practical instances, a vehicle is often required to simultaneously drop off and pick up goods at the same location. Min proposed the first version of the vehicle routing problem with simultaneous pick and delivery (VRPSPD) where a fleet of vehicles has to satisfy a set of transport requests and minimize or maximize some objective.

**Further development**
Following the introduction of the discussed problems in these two papers, we transition to more recent publications. Psaraftis et al. [27] (2016) showed in their literature review that the vehicle routing problem received substantial research attention since its introduction due to its growing significance in supply chain operations and the increased trend of globalization. This also holds to various papers and overviews discussing the VRPPD and its variants.

Desaulniers et al. [23] (2002) provided an extensive overview of the body of research related to the VRPPD. They explained the mathematical formulation of the VRPPD with time windows (VRPPDTW). Furthermore, they elaborated extensively on various heuristics and exact solution methodologies. They concluded that many practical instances of the VRPPD are large-scale and that researchers favored heuristic approaches. Most commonly seen where insertion and local search improvement

[1] In Section 3.2.1 and Section 3.2.2, reference statements include the time of publication to clarify historical context.

heuristics. Attanasio et al. [28] (2004) proposed and compared various implementations of tabu search heuristics tailored for a dynamic DARP. In dynamic DARP scenarios, transportation requests emerge throughout the day, and the objective is to accommodate as many of these requests as possible. Their computational results demonstrated that the proposed algorithms could effectively fulfill a substantial percentage of dynamically revealed orders. Flatberg et al. [29] (2005) emphasized the need for a shift in focus within the context of VRPs. They argued that the conventional emphasis on static and deterministic modeling should transition towards a more dynamic and stochastic perspective on vehicle routing. They pointed out that the assumption of transport systems being static and deterministic rarely holds in real industrial settings. In response to this insight, Flatberg et al. extended a generic VRP solver to handle the challenges of dynamic and stochastic scenarios. Beaudry et al. [30] (2008) analyzed and solved a patient transportation problem arising in large hospitals. The problem was initially modeled as a DARP where the goal was to provide an efficient transport service for patients between several location within the hospital. The paper proposed a heuristic that combines insertion and tabu search to generate feasible solutions. The results showed that the algorithm was capable of handling dynamic transport requests. Zhang et al. [31] (2012) developed two approaches for the stochastic travel time vehicle routing problem with simultaneous pickup and delivery (STT-VRPSPD). This vehicle routing problem is comparable to the discussed VRPPD but travel times between nodes are stochastic. The first solution approach employed is called scatter search (SS) and the second approach is a generic genetic algorithm (GA). The computational results suggested that the SS solutions outperformed the GA solutions.

In summary, these older papers on VRPPD and its variants mainly discuss the inclusion of stochastic and dynamic elements. The papers typically propose a solution methodology based on local search procedures or genetic algorithms. The experimental results in the papers showed that dynamic decision-making generally resulted in decreased total costs and more robust outcomes. Overall it could be stated that during this developmental phase, the main aim was to research the mathematical formulation and solution methodologies related to VRPPD with an emphasis on stochastic and dynamic information.

**Recent advancements**

Hornstra et al. [32] (2020) proposed an adaptive large neighborhood search metaheuristic to solve the vehicle routing problem with simultaneous pickup and delivery and handling costs (VRPSPD-H). In this variant of the VRPSPD handling operations are more explicitly modeled and only the last loaded item is accessible. Koulaeian et al. [33] (2015) proposed a new mathematical model for a multi-depot vehicle routing problem with simultaneous pickup and delivery. In this model, a heterogeneous fleet of vehicles is employed to service customers with pickup and delivery demands. Two different solution methodologies are used: the first based on imperialist competitive algorithm (ICA) and the second based on genetic algorithm (GA). Zhang et al. [8] (2019) introduced a new VRP variant that encourages the reutilization of collected items. The paper addresses a multi-commodity many-to-many vehicle routing problem with simultaneous pickup and delivery (M-M-VRPSPD) for a fast fashion retailer. What sets this model apart is its distinctive features: it encourages the reallocation of collected products from customers to different locations, and it deals with multiple types of commodities. Olgun et al. [34] (2020) proposed a hyper heuristic based on iterative local search and variable neighborhood descent to solve the green vehicle routing problem with simultaneous pick and delivery (G-VRPSPD). This paper aims to minimize fuel consumption costs while satisfying customer pickup and delivery demands.

Rist and Forbes [35] (2020) propose a mixed integer programming formulation and branch-and-cut algorithm to solve the DARP. This paper introduces a new formulation for the DARP based on route segments called restricted fragments. These restricted fragments are effectively building blocks allowing the model more efficient route planning. Ackermann et al. [36] (2021) analyze new metrics for optimization guidance for the dynamic DARP. These metrics allow better analysis of the insertion potential of dynamically revealed customers. Furthermore, this paper presents a MDP-based approach of modelling the DARP to enable an agent trained by reinforcement learning. Dong [37] (2021) explores new formulations of the DARP and developed multiple solution methods based on local search, column generation, metaheuristics and reinforcement learning for solving large-scale DARPs. The author conducted case studies based on data extracted from NYC taxi services. Liang et al. [38]

(2020) proposed an optimization model that maximizes the profit of the automated taxi system. This model considers the effect of traffic congestion in the DARP. The solution methodoly is based on the lagragian relaxation algorithm.

In summary, the more recent papers addressing VRPPD and its variants emphasize the incorporation of various problem features. Hornstra et al. concentrated on modeling realistic handling operations, Koulaeian et al. tackled the challenge of modeling a heterogeneous fleet, Zhang et al. formulated a mathematical model including multiple commodities, and Olgan et al. optimized routes with a focus on fuel consumption. These problem features hold significance in diverse industrial contexts, contributing to the growing body of research. It is worth mentioning that all four papers employed various OR tools, meta-heuristics, and local search algorithms to tackle their respective VRPPD variants, this shows that basic heuristics are still commonly used as the preferred solution methodology for VRPPDs, similar to the early 2000s. Conversely, the research field of DARP places a stronger focus on different solution methodologies. Ackermann et al. and Dong et al, in particular, stand out in the context of this thesis as they harness forms of reinforcement learning to address the optimization problem.

### 3.2.2 Recent advancements of (deep) reinforcement learning in route optimization

In this section we discuss the recent advancements regarding the application of (deep) reinforcement learning in the broader context of routing problems.

Raza et al. [39] (2022) published a comprehensive review paper in which the recent advancements in solving VRPs using RL are explored. In this paper, Raza et al. studied a multitude of papers and summarized several issues and challenges when incorporating RL algorithms as solution methods for VRPs. The following list highlights challenges that could also affect the model that will be proposed in this thesis:

- The state space increases exponentially with the increase in problem features. In general, finding effective policies requires much computational effort compared to effective policies in less complex routing problems. Therefore, incorporating more variables, in variants such as the VRPPD, requires more computational efforts.
- The solutions of the VRP and its variants require distance matrix calculations. In case of dynamically changing VRPs this can lead to large computational requirements.
- There is a shortage of VRP-related studies focusing on real-life characteristics like dynamic traffic environment, time windows and service times.

Despite these challenges, it is worth noting that VRPs are currently conventionally solved using mixed-integer programming (MIPs), adaptive large neighborhood searches (ALNS), and genetic algorithms [39]. Simple heuristics often yield computational effective solutions with minimal optimality gaps for many problem instances, circumventing the need for complex RL-based models.

Nevertheless, since the introduction of the DRL solution methodology of Silver et al. [1] (2018), there has been an increase in studies applying (deep) reinforcement learning in various fields related to decision-making in operations management. Table 9 provides a selection of papers relevant to this research project, mostly focusing on route optimization using various DRL-based methods to solve the problems outlined in their respective studies.
Nazari et al. [7] (2018) presented a framework based on RL for solving various VRPs. They demonstrated that their model outperformed classical heuristics on the capacitated VRP. The framework also had the potential to be applied to other variant of the VRP such as the stochastic VRP. There is no specific mention of the DARP or the VRPPD. Kalakanti et al. [48] (2019) proposed a solver based on reinforcement learning for the VRP. The problem is formulated as a Markov Decision Process (MDP). The simulation results suggest that the proposed method is able to obtain better and similar results compared to the Clarke-Wright saving heuristic and the sweep heuristic. Geng et al. [?] (2020) designed a route planning algorithm based on deep reinforcement learning (DRL) to minimize pedestrian travel time. The agent learns through competing deep Q networks and aims to learn strategies to avoid congested roads. Simulation results show that the algorithm significantly outperforms more traditional

Table 9: Overview of studies applying DRL-based methods for various routing problems.

| Context | Study | Problem type | DRL Algorithm |
|---|---|---|---|
| Routing optimization | Nazari et al. [7] (2018) | Various VRPs | Actor-Critic |
| Routing optimization | Zhang et al. [8] (2020) | Multi-VRP with soft time windows | Multi-agent DRL with attention |
| Pedestrian flow planning | Geng et al. [40] (2020) | DARP | DDQN |
| Routing optimization | Kullman et al. [41] (2019) | VRP with stochastic service requests | D3QN |
| Routing optimization | Zhao et al. [42] (2021) | VRP with time windows | Actor-Critic |
| Inventory allocation problem | van Steenbergen et al. [43] (2021) | Stochastic dynamic inventory allocation problem | DL-VFA and NN-VFA |
| Route optimization | Balaji et al. [44] (2019) | Stochastic VRP | DQN |
| Route optimization | Iklassov et al. [45] (2023) | VRP with stochastic demand | Attention model |
| Route optimization | Li et al. [46] (2021) | PDP | Attention model |
| Route optimization | Soroka et al. [47] (2023) | Capacitated PDP with time windows | Attention model |
| Route optimization | Kalakanti et al. [48] (2019) | Capacitated VRP with stochastic demand | 2-phase solver based on DQN |
| Route optimization | Dong et al. [37] (2022) | Various DARPs | Metaheuristics combined with Thompson Sampling |
| Route optimization | Ackermann et al. [36] (2022) | Dynamic DARP | DQN |

heuristics such as the shortest-path heuristic. Van Steenbergen et al. [43] (2023) studied dynamic routing methods and the deployment of unmanned aerial vehicles as a potential solution for humanitarian relief distribution. The paper evaluates dynamic solutions generated by deep reinforcement learning approaches. The paper contributed to the field of vehicle routing with travel time uncertainty by analyzing a heterogeneous fleet problem variant with continuous travel time distributions. Furthermore, in the paper two different RL methods are compared to four benchmarks. The results show that the use of dynamic planning methods successfully mitigated travel time uncertainties in humanitarian operations.

In summary, the papers discussed above highlight the effectiveness of DRL compared to traditional algorithms. Generally, three main DRL algorithms are utilized for training neural network agents: (i) deep Q-learning and its variants, (ii) actor-critic models, and (iii) attention models. Furthermore, the types of routing problems discussed are very divergent. No paper concretely discusses a vehicle routing problem with pickup and delivery, with stochastic and dynamic travel- and handling times.

### 3.2.3 Contributions

To the best of our knowledge and after analysis of the citations of Temizoz et al. [49], no papers have been published that utilize deep controlled learning in the context of route optimization. This DRL algorithm is currently mainly utilized in the field of inventory control. Similarly, to the best of our knowledge no paper has discussed the VRPPD variant formulated in this research and proposed a methodology for solving this routing problem. In their review, Raza et al. [39] highlighted that further research should focus on the incorporation of additional problem features and more complex variants of the VRP, specifically mentioning the VRPPD.

We contribute to the existing literature with a solution methodology based on deep controlled learning [49] for dynamic decision-making in our routing problem. This includes an MDP formulation of the vehicle routing problem with pickup and delivery, with soft time constraints, stochastic and dynamic travel- and handling times, and a capacitated homogeneous fleet, following the framework proposed by Powell [2], we elaborate further on deep controlled learning in Chapter 4. Further contribution lies in the fact that we employ modeling data based on direct observations from a logistical company, increasing the overall validity of our modeling results.

## 3.3 Conclusions

In this chapter, the research question "*What solutions based on (deep) reinforcement learning have been proposed in the literature for similar transport network problems?*" is answered. In Section 3.1, we provided background information regarding the vehicle routing problem with pickup and delivery and classified the problem presented in Chapter 2. In Section 3.2, we discussed both the historical development of the VRPPD and the solution methodologies employed to solve various VRPPD variants. Furthermore, we also reviewed the current solution methodologies based on DRL that solve various routing problems. Based on the review we can conclude that most vehicle routing problems utilize either (i) deep Q-learning and its variants, (ii) actor-critic models, or (iii) attention models as their underlying DRL solution methodology. Lastly, to the best of our knowledge, there are no papers which utilize deep controlled learning in the context of route optimization. Similarly, there are no papers discussing the exact routing problem considered in this research.

# 4 Model design

This chapter discusses the model design. Firstly, in Section 4.1 the modeling goals are discussed. In Section 4.2 the modeling assumptions are stated. In Sections 4.3 and 4.4 the model notation and formulation of the model are discussed. In Section 4.5 the algorithmic framework is outlined. Finally, in Section 4.6 the sub-question related to overall the model design is answered.

As outlined in Section 3.2.3, our contribution to the body of literature involves researching the effectiveness of (deep) reinforcement learning techniques for the vehicle routing problem with pickup and delivery. Therefore, accurately modeling the transport network as an MDP is crucial, and requires defining state variables, decision variables, exogenous information, a transition function, and a cost function. Consequently, this chapter primarily focuses on formulating the E2E transport network as a Markov Decision Process with stochastic transition probabilities.

There are several reasons to follow the sequential decision-making framework by Powell [2] for this use case. These reasons are similar to the reasons presented in the paper of Steenbergen et al. [43] where reinforcement learning is used for humanitarian relief distribution under stochastic travel times.

- The framework is tailored towards RL methods. The framework is a suitable foundation for the (deep) reinforcement learning methods we employ to address sequential decision problems. Stochasticity allows for exploration by encouraging the agent to try varying actions.
- The transition function in the framework of Powell [2] explicitly enables sampling state transitions. This is done by defining a post-decision state and outcome spaces. This methodology deviates compared to the more classical MDP framework, for example outlined by Puterman et al. [50] where post-decision states are not explicitly modeled.
- We aim to model stochastic travel- and handling times. Therefore, the outcome- and state spaces are intractable. Consequently, the explicit calculation of transition probabilities becomes infeasible to perform, and using approximations is necessary.

## 4.1 Modeling Goals

The modeling goals and the capabilities of the model are formulated as follows:

- The stochastic program formulation represents the proposed E2E system and includes relevant problem features stated by Gam Bakker.
- The model can compare policies on varying realistic-sized problem instances based on key performance indicators.
- The model provides insight into the actions it takes in each decision-making moment.
- The model efficiently trains neural networks and can utilize the trained neural network as an agent.

## 4.2 Model Assumptions

To model the proposed E2E transport network of Gam Bakker and be able to formulate the network the following assumptions are made:

- We do not consider orders from other clients of Gam Bakker. This means only orders for Cargill are considered.
- We do not consider the kilometers driven, at the beginning of the simulation towards the pickup location of the first order and at the end of the day away from the delivery location of the last order, as empty kilometers. Vehicles always start at the production plant and can terminate at every location. VRPs with this property are called open-VRPs.

- A list of orders is known before starting the simulation. There are no new orders dynamically revealed.
- Each order only specifies a pickup location, a delivery location, a release time, and a deadline.
- Each order requires exactly one truck to be carried out. Remark that cargo is not modeled. This implicates the assumption that orders provided by clients respect truck capacity.
- Orders have a 100% success rate. We do not model disruptions such as the mechanical failures of a vehicle.
- Once an order has been assigned, it must be completed before executing another order (no preemption).
- All vehicles have equal travel and handling times distributions.
- All locations have operational time windows. Working outside these windows leads to an increase in waiting time or penalty costs.
- Traffic conditions are stochastic, time-dependent, and i.i.d. This means that travel time between locations is variable.
- The workload intensity at the warehouses, terminals, and production plant are stochastic, time-dependent, and i.i.d. This means that handling times are variable.

## 4.3 Case description and notation

In this section we describe the variant of vehicle routing problem with pickup and delivery, with soft time constraints, stochastic travel- and handling times, and a capacitated homogeneous fleet. This description is based on the current situation described in Chapter 2 and takes the model assumption from Section 4.2 into consideration. The problem setting is the proposed E2E transport network in the Amsterdam area where Cargill's products are transported. Cargill requests transport movements between their production plant and various warehouses and terminals in Amsterdam. In this VRPPD variant we consider fixed facility locations and fleet size.

First, we mathematically introduce the problem. Consider $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as a graph with a set of nodes (vertrices) $\mathcal{V}$ and a set of arcs (edges) $\mathcal{E}$. The set of nodes $\mathcal{V} = \{0, 1, 2, 3, 4\ \}$ represents the production facility, the warehouse Hoogtij, terminal Hoogtij, warehouse Amsterdam, and terminal Amsterdam respectively. Additionally, set $\mathcal{E} = \{(i, j) : i, j \in \mathcal{V}\}$ contains all arcs between the nodes. Therefore, graph $\mathcal{G}$ represents the transport network layout.

Furthermore, let $T^{travel}_{v,i,j,t}$ be the random variable representing the travel times for vehicle $v \in \mathcal{F}$ traveling from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$ over arc $(i,j) \in \mathcal{E}$ during time $t \in \mathcal{T}$. Similarly, let $T^{handle}_{v,i,t}$ be the random variable representing the handling times for vehicle $v \in \mathcal{F}$ that is handled at location $i \in \mathcal{V}$ during time $t \in \mathcal{T}$. The set $\mathcal{F}$ represents the fleet that is responsible for transport within the E2E network. The set $\mathcal{T} \subset \mathbb{R}_{\geqslant 0}$ represents the time since the start of the day and is bounded by $t_{max}$.

We consider a set of transport requests $\mathcal{O}$. Each transport request, or order (this terminology is used interchangeably in this thesis), $o \in \mathcal{O}$ is characterized by an origin and a destination given as $ori_o, dest_o \in \mathcal{V}$ representing the pickup point and the delivery point respectively. An order is satisfied when a vehicle picks up the container at the origin and delivers it to the destination. Additionally, all orders have delivery windows during which orders must be satisfied. The release time is denoted as $rt_o \in \mathbb{R}_{\geqslant 0}$ and the deadline is denoted as $dl_o \in \mathbb{R}_{\geqslant 0}$. The set of vehicles $\mathcal{F}$ aims to satisfy the set of orders $\mathcal{O}$ in the most effective manner, according to some weighted cost functions. The costs are dependent on several cost parameters
Each location in set $\mathcal{V}$ also operates within a certain time window. The node $i \in \mathcal{V}$ has an operational window $open_i, close_i \in \mathbb{R}_{\geqslant 0}$.

Unlike common VRPs with pickup and delivery, in this problem we do not model capacity. This is because we assume that one transport request from Cargill completely satisfies the capacity of the truck, therefore only one order per vehicle can be executed at each time. Subsequently, the decision space and transition dynamics are simplified because after an order is assigned another order cannot be assigned simultaneously.

## 4.4 Formulation

In this section, we formulate the E2E transport network as an indefinite MDP. All elements relevant to the formulation of the MDP are separately discussed from Section 4.4.1 to Section 4.4.7. To clarify, an indefinite MDP refers to a variant of an MDP, in which there is no defined horizon length. It suggests uncertainty or lack of explicit knowledge about the number of states it will visit. Termination in this indefinite MDP is determined by two conditions: the time of day and the number of orders available.

### 4.4.1 Stages

We identify that decisions are only needed to be made the moment at which a vehicle becomes available and a new order can be assigned. However, the specific time between adjacent decision epochs is not known beforehand because the availability of vehicles is subject to stochasticity. Stochastic travel- and handling times can lead to continuous and unbounded state- and outcome spaces making it difficult to discretize. The stochasticity originates from the stochastic handling time and stochastic travel time. Therefore, the decision is made to define the stages (decision epochs) in a relatively uncommon way.

We define the decision epoch k $\in \mathcal{K} = \{$ 0, 1, ..., K $\}$, where K is a random variable dependent on several model parameters and the travel- and handling time realizations within time window [0, $t_{max}$].

### 4.4.2 States

The state $S_k$ encapsulates all relevant information to model the transport network and enables making feasible decisions, calculating costs, and computing transitions. The state vector can be categorized into three sub-vectors. The resource vector $R_k$ which captures the information about all vehicles $\mathcal{F}$. The node vector $N_k$ which captures the relevant information about the nodes $\mathcal{V}$. The order vector $O_k$ which captures the relevant information about the orders.

The resource vector is formally formulated as follows:

$$R_k = \Big(\hat{v}_k, w_{k,v}, q_{k,v}, B_{k,v}, I_{k,v}, J_{k,v}, D_{k,v}, t^{\text{dep}}_{k,v}, t^{\text{fin}}_{k,v}, t_k\Big) \, \forall v \in \mathcal{F} \tag{1}$$

State variable $\hat{v}_k$ shows the vehicle availability in the current stage. $w_{k,v}$ indicates whether vehicle $v \in \mathcal{F}$ is currently waiting and $w_{k,v}$ indicates whether vehicle $v \in \mathcal{F}$ is currently terminated. In the state the begin node $B_{k,v}$ of the vehicle, the origin $I_{k,v}$ of the order the vehicle is currently performing, and the destination $J_{k,v}$ of the order the vehicle is currently performing are tracked. Remark that there is a specific difference between the begin node and origin. The begin node is the location where the vehicle becomes available (where the previous order ended) and the origin the node at which the available vehicle is loaded with the new order. The begin node and origin can be identical but this is not necessary. The time window in which the vehicle $v \in \mathcal{F}$ must deliver its current order is captured with state variable $D_{k,v}$ meaning deadline. Furthermore, the time elapsed since departure is stored by variable $t^{\text{dep}}_{k,v}$ and the expected time until the order is finished for each vehicle is stored in $t^{\text{fin}}_{k,v}$.

Secondly, the node vector is formally formulated as follows:

$$N_k = (St_{k,i}, E_{k,i}) \, \forall i \in \mathcal{V} \tag{2}$$

Information regarding the time windows in which the nodes are operational is captured by variable $St_{k,i}$ which represents the time until the start of the time window, and $E_{k,i}$ which represents the time until the end of the time window.

Lastly, vector $O_k$ is a vector containing the availability of all orders. $A_{k,o}$ represents whether an order is currently available to perform. The cardinality of $A_{k,o}$ is equal to the total number of orders in the problem instance. This order vector is formally formulated as follows:

$$O_k = (A_{k,o}) \, \forall o \in \mathcal{O} \tag{3}$$

Consequently, by combining these three vectors and adding the variable representing the current time $t_k$ the state vector is created. The state variable can be formally formulated as follows:

$$S_k = (R_k, N_k, O_k, t_k) \, \forall k \in K \tag{4}$$

Table 10: Overview of state variables including notations and descriptions.

| State variables | | Description |
|---|---|---|
| $\hat{v}_k$ | $\in \mathcal{F}$ | Current available vehicle at decision epoch $k, k \in \mathcal{K}$ |
| $w_{k,v}$ | $\in \{0,\, 1\}$ | Current waiting status vehicle $v \in \mathcal{F}$ at decision epoch $k, k \in \mathcal{K}$ |
| $q_{k,v}$ | $\in \{0,\, 1\}$ | Current termination status vehicle $v \in \mathcal{F}$ at decision epoch $k, k \in \mathcal{K}$ |
| $B_{k,v}$ | $\in \mathcal{V}$ | Begin node of vehicle $v \in \mathcal{F}$ at decision epoch $k, k \in \mathcal{K}$ |
| $I_{k,v}$ | $\in \mathcal{V}$ | Origin of vehicle $v \in \mathcal{F}$ at decision epoch $k, k \in \mathcal{K}$ |
| $J_{k,v}$ | $\in \mathcal{V}$ | Destination of vehicle $v$ at decision epoch $k, k \in \mathcal{K}$ |
| $D_{k,v}$ | $\in \mathbb{R}_{\geqslant 0}$ | Deadline order assigned to vehicle $v$ at decision epoch $k, k \in \mathcal{K}$ |
| $t^{\text{dep}}_{k,v}$ | $\in \mathbb{R}_{\geqslant 0}$ | Time since departure of vehicle $v \in \mathcal{F}$ at decision epoch $k, k \in \mathcal{K}$ |
| $t^{\text{fin}}_{k,v}$ | $\in \mathbb{R}_{\geqslant 0}$ | Expected time until vehicle $v \in \mathcal{F}$ at destination at decision epoch $k, k \in \mathcal{K}$ |
| $t_k$ | $\in \mathbb{R}_{\geqslant 0}$ | Current time at decision epoch $k, k \in \mathcal{K}$ |
| $St_{k,i}$ | $\in \mathbb{R}$ | Time until start of time window of node $i \in \mathcal{V}$ at decision epoch $k, k \in \mathcal{K}$ |
| $E_{k,i}$ | $\in \mathbb{R}$ | Time until end of time window of node $i \in \mathcal{V}$ at decision epoch $k, k \in \mathcal{K}$ |
| $A_{k,o}$ | $\in \{0, 1\}$ | Availability status of order $o \in \mathcal{O}$ at decision epoch $k$, $k \in \mathcal{K}$ |

### 4.4.3 Decision variables

At every stage/decision epoch $k \in \mathcal{K}$ with current time $t_k$, one vehicle $\hat{v}_k$ is marked as available because the vehicle just finished its transport request.

Decision variable $x_k$ represents the decision that the agent takes in epoch $k \in \mathcal{K}$. Three types of decisions can be taken at each moment: (i) assigning a new order $o \in \mathcal{O}$ to the available vehicle, (ii) waiting for 10 minutes, and (iii) terminating at the current position. This can be formally expressed as follows:

$$x_k = \begin{cases} \text{if } x_k \leq |O_k| & \text{Order } x_k \text{ is assigned to the available vehicle} \\ \text{if } x_k = |O_k| + 1 & \text{The available vehicle waits at current location for 10 minutes} \\ \text{if } x_k = |O_k| + 2 & \text{The available vehicle is terminated} \end{cases} \tag{5}$$

The decision space is equal to $|O_k|$ + 2. In contrast to the state- and outcome space, the decision space is discrete and relatively small. Remark that the decision space is dependent on the number of orders and therefore dependent on the problem instance.

Furthermore, several implications and constraints for the decision in Equation 5 have to be taken into consideration:

- Decisions are only made for the vehicle that is marked available $\hat{v}_k$ in stage $k$, not for other vehicles.

- $x_k = o$ is only feasible if the order is currently available; $A_{k,o} = 1$, this holds $\forall o \in \mathcal{O}$. This constraint prevents vehicles from performing the same transport request multiple times.
- In each stage the decision can be taken to terminate the vehicle. Therefore, the vehicle can terminate at each location, this is in line with the open VRP assumption.
- Once a vehicle terminates $q_{k,v} = 1$, it is rendered inactive, and incapable of taking further actions. Terminated vehicles are not marked as available $\hat{v}_k$ in subsequent stages post-termination.
- A decision can be made wherein the available vehicle $\hat{v}_k$ is required to wait for 10 minutes before becoming available again. If an order is released at the current location of $\hat{v}_k$ within this time window, the vehicle has the opportunity to fulfill this order, minimizing empty kilometers.
- In the current epoch the expected duration of performing a new order should not exceed the maximum time $t_{max}$.
- Order $o$ cannot be assigned to the currently available vehicle if the current time $t_k$ is smaller than the release time of the order $rl_o$.

### 4.4.4 Costs function

The total cost $C_k$ follows a weighted objective of the problem. The cost function is based on the following elements:

- The number of empty kilometers.
- The number of full kilometers.
- The duration of waiting or inactivity.
- The duration of performing the order.
- Penalty for pickup and delivery at locations outside their operational windows.
- Penalty for delivering orders after their respective deadline.
- Penalty for not satisfying orders.
- The makespan of the work day.

Remark that the several elements that contribute to the cost $C_k$ are only known after travel- and handling times are realized. Furthermore, some cost contributions can only be determined in the final decision epoch. To formalize the costs function, costs are divided into (i) cost contributions that are directly known after taking a decision, (ii) cost contributions that are known after the realization of travel- and handling times, and (iii) cost contributions that can only be calculated once the final epoch has been reached.

The deterministic cost contribution can be formalized as follows:

$$C_k^{det}(S_k, x_k) = \begin{cases} \xi_1 \ d_{B_{k,\hat{v}_k}, I_{k,\hat{v}_k}} + \xi_2 \ d_{I_{k,\hat{v}_k}, J_{k,\hat{v}_k}} & \text{if } x_k \leq |O_k| \\ \xi_3 \cdot 600 & \text{if } x_k = |O_k| + 1 \\ \xi_9 \cdot J_{K,\hat{v}_K} & \text{if } x_k = |O_k| + 2 \end{cases} \tag{6}$$

In case the decision to fulfill a transport request is taken, then the cost related to empty kilometers is defined as the distance between the begin node (the previous destination) and the origin. The cost related to loaded kilometers is defined as the distance between the origin and destination. Remark that $d_{i,j}$ represents the distance between node $i \in \mathcal{V}$ and $j \in \mathcal{V}$. In case the decision is taken to wait for 10 minutes, then $C_k^{det}$ is equal to the costs of waiting during this time frame. Lastly, in case the decision is taken to terminate the cost represents the distance away from Hoogtij.

The stochastic cost contribution can be formalized as follows:

$$\begin{aligned} C_k^{sto}(S_k^{post}, W_{k+1}) = \xi_3 \ &\max(0, St_{k,J_{k,W_{k+1}^v}} - W_{k+1}^t) \\ &+ \xi_4 \ (t_{k,W_{k+1}^v}^{dep} + W_{k+1}^t) \\ &+ \xi_5 \ \max(0, -E_{k,J_{k,W_{k+1}^v}}) \\ &+ \xi_6 \ \max(0, (t_k + W_{k+1}^t) - D_{k,W_{k+1}^v}) \end{aligned} \tag{7}$$

Firstly, the vehicle that becomes available next epoch is denoted as $W_{k+1}^v$ and the time between the two adjacent epochs is denoted as $W_{k+1}^t$. In Section 4.4.5 the exact calculation of these variables is discussed. The first element in the stochastic cost contribution represents the waiting cost in case an order is finished before the operational time of a location, can be seen as if the order arrived too early. The second element is related to the cost of the total order duration. The third element relates to the penalty cost of finishing an order after the operational window of the destination. The fourth element relates to the penalty costs of delivering orders after their respective deadline.

The cost contribution calculated once the final epoch is reached can be formalized as follows, remark that K represents the final decision epoch:

$$C_K^{fin} = \xi_7 \sum_{o \in \mathcal{O}} A_{K,o} \ + \ \xi_8 \cdot t_K + \xi_9 \cdot J_{K,\hat{v}_K} \tag{8}$$

The first element in this equation represents the penalty cost of the total number of non-satisfied orders in the final epoch. The second element represents the current time of the final epoch. Therefore, the second element represents the makespan. The last element represents the distance the current vehicle is away from Hoogtij.

By combining these three cost functions, the total cost $C_k$ can be calculated. The costs are incurred at different moments. The deterministic costs $C_k^{det}$ are incurred each time after the pre-decision state is modified with the decision taken by the agent. The stochastic cost $C_k^{sto}$ is incurred after the post-decision state is modified with the outcome (exogenous information). Importantly, the costs of the final epoch $C_K^{fin}$ are incurred once the terminal decision epoch has been reached. Additionally, it is worth noting that the value of the weight reflects the importance of each cost relative to the others. A visual representation of the cost composition for one Markov chain can be found in Figure 16.

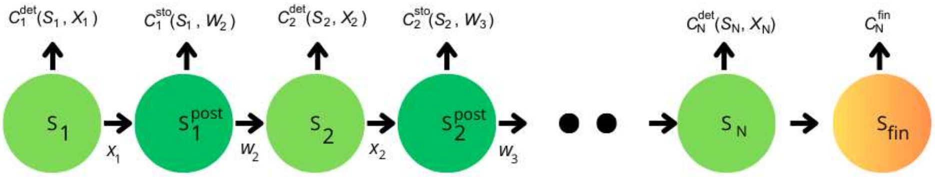


Figure 16: Schematic representation of the cost composition. The light-green circles represent the pre-decision states, the dark-green circles the post-decision states, and the orange circle represents the final state.

### 4.4.5 Exogenous information

The exogenous information $W_{k+1}$ captures all uncertainties that are realized in the post-decision state and become known at the beginning of the next state. Upon making a decision, the available vehicle either (i) starts executing the assigned order by heading to the pickup location or initiating loading, contingent on the current vehicle location and the pickup point of the order, (ii) undergoes a ten-minute wait until it becomes available again, or (iii) terminates and returns to the original Hoogtij. Hence, the realization of travel- and handling times influence the duration until the next vehicle is available. Similarly, in case vehicles are currently waiting, the duration until the waiting time elapsed also influences the duration until the next vehicle is available. Consequently, accurately computing the exogenous information in the post-decision state is crucial for transitioning to the next state. In this model, the exogenous information represents which vehicle will become available first, $W_{k+1}^{v}$, and the duration until the next vehicle becomes available, $W_{k+1}^{t}$.

**Realization travel- and handling times**

The realizations of travel - and handling times must consider the fact that the arrivals of the vehicles need to be in the future at epoch $k$ for each vehicle. This is formulated as $\mathbb{F}\left(T_{v,b,i,j,t} \mid T_{v,b,i,j,t} \geqslant t_{k,v}^{\mathrm{dep}}\right)$ with $\mathbb{F}$ being the inverse probability function for the travel time and handling time conditional to the time since departure. For this problem, the realization of the order duration must always be larger than the time since departure. Otherwise, infeasible solutions would be generated. This method of guaranteeing that the duration of the order is in the future is similar to the method used by Steenbergen et al. [43].

$T_{v,b,i,j,t}$ represents the travel time plus handling time for vehicle v $\in \mathcal{F}$ between begin node b $\in \mathcal{V}$ and origin i $\in \mathcal{V}$ and destination j $\in \mathcal{V}$ at time $t \in \mathbb{R}_{\geqslant 0}$. Remark that the begin node and origin can be the same node.

Lastly, $T_{v,b,i,j,t}$ is composed of multiple stochastic values. The handling time at both the origin node and the destination node and the travel time between origin and destination and possibly the travel time between begin node and origin, dependent whether the begin node and the origin are the same node. Furthermore, both the travel times and handling times are time-dependent. $T_{v,b,i,j,t}$ is formalized in Equation 9

$$T_{v,b,i,j,t} = t_{v,b,i,t}^{travel} + t_{v,i,t}^{handle} + t_{v,i,j,t}^{travel} + t_{v,j,t}^{handle} \tag{9}$$

**Formulation**

The exogenous information $W_{k+1}$ is characterized by the vehicle that first comes available $W_{k+1}^{v}$ and the time between this availability and the adjacent epoch. This time is denoted as $W_{k+1}^{t}$. Taking the minimum time until availability over all vehicles provides the first time until availability $W_{k+1}^{t}$ and the argument of this minimum gives the associated vehicle $W_{k+1}^{v}$. The exogenous information $W_{k+1}$ is formalized in Equation 10 and Equation 11

$$\begin{gathered} W_{k+1} = \left(W_{k+1}^{v}, W_{k+1}^{t}\right) = \\ \left(\operatorname*{argmin}_{v \in \mathcal{F}} \left(\max\left\{sub, St_{k,J_{k,v}}\right\}\right), \min_{v \in \mathcal{F}} \left(\max\left\{sub, St_{k,J_{k,v}}\right\}\right)\right) \end{gathered} \tag{10}$$

$$sub = \begin{cases} T_{v,B_{k,v},I_{k,v},J_{k,v},t_k} - t_{k,v}^{\mathrm{dep}} & \text{if } w_{k,v} = 0 \\ t_{k,v}^{\mathrm{fin}} & \text{if } w_{k,v} = 1 \end{cases} \tag{11}$$

**Demonstration**

For illustrative purposes, we provide a schematic representation of two scenarios of the same post-decision state in Figure 17. In realization 0, the exogenous information tuple $W_{k+1}$ of vehicle 1 was shorter than both instances of the time until each order arrived, represented by the dark green line. In realization 1, the exogenous information tuple $W_{k+1}$ did not require correction because the order performed by vehicle 0 was finished earlier compared to the elapsed waiting time.

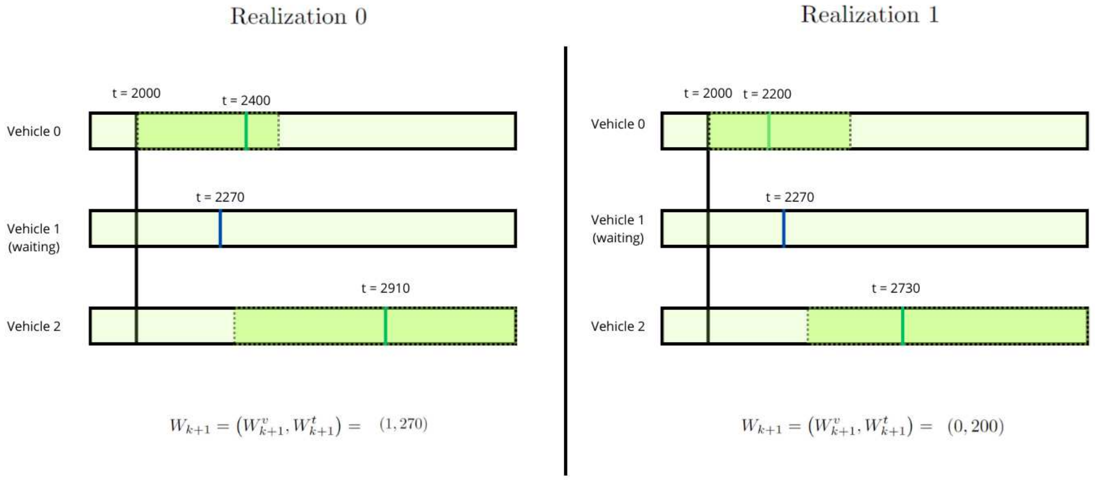


Figure 17: Schematic illustrating the impact of waiting vehicles in a random post-decision state. The current time is equal to 2000. The neon green blocks represent the ranges in which orders can be finished following the $\mathbb{F}$ inverse probability function. The dark green lines are the realized finishing times following the exogenous information. The blue line represents the finishing time of waiting. Remark that this line is equal between both realizations because it is not dependent on the stochastic factors.

.

### 4.4.6 Transition dynamics

The system can be transitioned from $S_k$ to $S_{k+1}$ according to some transition function $\mathcal{S}$ once a decision $x_k$ is made, costs are calculated based on the above mentioned cost functions and the exogenous information $W_{k+1}$ is calculated using the inverse probability function $\mathbb{F}$. This can be defined as follows:

$$S_{k+1} = \mathcal{S}(S_k, x_k, W_{k+1}) \tag{12}$$

The transition has a deterministic element from the pre-decision state to the post-decision state and a stochastic element from the post-decision state to the next state.

**Deterministic transition**

We first consider the deterministic case in which an order is assigned to the available vehicle $x_k \leq |O_k|$ and incorporate the decision with the pre-decision state:

$$w_{k,\hat{v}_k} = 0 \tag{13}$$

$$A_{k,x_k} = 0 \tag{14}$$

$$I_{k,\hat{v}_k} = ori_{x_k} \tag{15}$$

$$J_{k,\hat{v}_k} = dest_{x_k} \tag{16}$$

$$t^{\text{dep}}_{k,\hat{v}_k} = 0 \tag{17}$$

$$D_{k,\hat{v}_k} = dl_{x_k} \tag{18}$$

$$t^{\text{fin}}_{k,\hat{v}_k} = \mathbb{E}[T_{\hat{v}_k, B_{k,\hat{v}_k}, I_{k,\hat{v}_k}, J_{k,\hat{v}_k}, t_k}] \tag{19}$$

We also consider the deterministic case in which the decision is made to wait: $x_k = |O_k| + 1$.

$$w_{k,\hat{v}_k} = 1 \tag{20}$$

$$t^{\text{dep}}_{k,\hat{v}_k} = 0 \tag{21}$$

$$t^{\text{fin}}_{k,\hat{v}_k} = 600 \tag{22}$$

We also consider the deterministic case in which the decision is made to terminate: $x_k = |O_k| + 2$.

$$q_{k,\hat{v}_k} = 1 \tag{23}$$

$$t^{\text{fin}}_{k,v} = \infty \tag{24}$$

Equation 13 updates the status of the vehicle to active. Equation 14 updates the available order list and modifies assigned orders as unavailable. Equation 15 assigns the origin of the order as the new origin of the vehicle $\hat{v}_k$. Equation 16 assigns the destination of the order as the new destination of the vehicle $\hat{v}_k$. Equation 17 reset the time since departure back to zero. Equation 18 assigns the deadline of the order to the state variable carrying information regarding the deadline of the order it currently transports. Equation 19 updates the expected time until the order is finished. Equation 20 updates the status of the vehicle to waiting. Equation 21 updates the time since departure, in this context this variable is used as time since waiting, to zero. Equation 22 updates the expected time until the order is finished to ten minutes, in this context this variable is used as expected time until the waiting time is over. Equation 23 terminates the currently available vehicle by updating the termination status to one. Lastly, for modeling purposes the expected time until the next order arrives is set to infinity if the vehicle is terminated in Equation 24.

**Stochastic transition**
The transition from post-decision state to the next state can be formalized as follows:

$$\hat{v}_{k+1} = W^v_{k+1} \tag{25}$$

$$B_{k+1,W^v_{k+1}} = J_{k,W^v_{k+1}} \tag{26}$$

$$t^{\text{dep}}_{k+1,v} = t^{\text{dep},x}_{k,v} + W^t_{k+1} \forall v \in \mathcal{F} \tag{27}$$

$$St_{k+1,i} = St^x_{k,i} - W^t_{k+1} \ \forall i \in \mathcal{V} \tag{28}$$

$$E_{k+1,i} = E^x_{k,i} - W^t_{k+1} \ \forall i \in \mathcal{V} \tag{29}$$

$$w_{k,W^v_{k+1}} = \begin{cases} 0 & \text{if } St_{k+1,J_{k,W^v_{k+1}}} \leq 0 \\ 1 & \text{if } St_{k+1,J_{k,W^v_{k+1}}} > 0 \end{cases} \tag{30}$$

$$t^{\text{fin}}_{k+1,v} = \begin{cases} T_{v,B_{k,v},I_{k,v},J_{k,v},t_k} - t^{\text{dep}}_{k,v} & \text{if } w_{k,v} = 0 \\ t^{\text{fin}}_{k,v} - W^t_{k+1} & \text{if } w_{k,v} = 1 \wedge St_{k+1,J_{k,W^v_{k+1}}} \leq 0 \\ St_{k+1,J_{k,W^v_{k+1}}} & \text{if } w_{k,v} = 1 \wedge St_{k+1,J_{k,W^v_{k+1}}} > 0 \end{cases} \quad \forall v \in \mathcal{F} \tag{31}$$

$$t_{k+1} = t^x_k + W^t_{k+1} \tag{32}$$

Equation 25 updates the available vehicle to the first available vehicle following the exogenous information. Equation 26 sets the begin node of the available vehicle to its current destination. Equation

27 updates the time since the departure of all vehicles with the difference between the current and next decision epoch. Furthermore, equations 28 and 29 update the times until start and end of time window for each location. Equation 30 updates the waiting status of the available vehicle in case this vehicle arrives before the operational window of its destination has started. Equation 31 updates the expected finish time of each vehicle. The exact formula that is used is dependent on whether the vehicle is waiting or not and whether the vehicle is waiting for the opening of an operational window or chooses to wait. Furthermore, the underlying assumption is that variable $T_{v,B_{k,v},I_{k,v},J_{k,v},t_k}$ respects the condition that the expected order duration is larger than the time since departure, as discussed in 4.4.5. Lastly, in Equation 32 the current time is updated.

### 4.4.7 Objective function

This Section describes the method of finding the policy to solve the sequential decision problem formulated to optimality. The problem formulation itself draws parallels to the problem formulation proposed by Steenbergen et al. [43]. Consequently, the formulation of the probabilities of vehicle arrivals is similar to the formulation used in the paper of Steenbergen et al.

We aim to find a policy $\pi \in \Pi$ and make decisions based on that policy $X_k^\pi(S_k)$. This policy minimizes the expected total costs $C_k(S_k, x_k)$ given initial state $S_0$ summing over all decision epochs $k \in \mathcal{K}$. This can be formalized in the objective function as follows:

$$\min_{\pi \in \Pi} \mathbb{E}\left[\sum_{k=0}^{K} C_k\left(S_k, X_k^\pi\left(S_k\right)\right) \mid S_0\right] \tag{33}$$

The state trajectory follows the transition functions discussed in 4.4.6. State $S_0$ is initialized based on the problem instance and the final epoch is reached in case (i) all orders have been fulfilled or (ii) the current time $t_k$ is larger than $t_{max}$.

Conceptually, the optimal policy can be obtained by solving the Bellman equation 34.

$$V_k\left(S_k\right)=\min_{\pi \in \Pi}\left[C_k\left(S_k, X_k^\pi\left(S_k\right)\right)+\sum_{\omega_{k+1} \in \Omega_{k+1}} \mathbb{P}\left(W_{k+1}=\omega_{k+1}\right) V_{k+1}\left(S_{k+1} \mid S_k, X_k^\pi\left(S_k\right), \omega_{k+1}\right)\right] \forall S_k \tag{34}$$

However, as previously described the costs cannot be computed deterministically because we are dealing with stochastic travel- and handling times. Altough the calculation of the direct costs $C_k(S_k, X_k^\pi(S_k))$ is feasible, calculating the expected future costs considering all possible future outcomes becomes infeasible. Specifically, determining the second element of the bellman equation where we sum over $\Omega_{k+1}$ is intractable because the outcome space is continuous. Consequently, (deep) reinforcement learning can be utilized to approximate the value function and solve the sequential decision problem in an effective manner.

## 4.5 Algorithmic framework

As outlined in Chapter 1 the research goal of this thesis is to explore the effectivity of decision-making algorithms based on DRL in the context of route optimization. In this section, we discuss the theory and logic on which two decision-making algorithms are based. In Section 4.5.1, we discuss the consolidation algorithm. This rule-based algorithm is based on the planning and scheduling principles employed by the planners of Gam Bakker and functions as the benchmark against which the agent trained using deep reinforcement learning will be tested. We train neural networks based on a DRL algorithm called deep controlled learning, this algorithm is discussed in Section 4.5.2. The neural networks trained using the DCL algorithm and used as agents will be abbreviated to NNARLs.

### 4.5.1 Consolidation algorithm

The consolidation algorithm is a rule-based algorithm employed in the dynamic environment to select an action for the available vehicle in each decision epoch. The underlying logic of the rules is based on the planning and scheduling principles used by the planning team of Gam Bakker. It is important to remark that while the consolidation algorithm is based on the current planning and scheduling principles, it is not currently used in practice as discussed in Section 2.2. This is because our research methodology requires an online algorithm that makes decisions based on the current state of the formulated E2E transport network for comparison. The current route planning method operates offline.

The overarching goal of the consolidation algorithm is to minimize empty kilometers through order consolidation and vehicle reuse. It operates on several key principles:

- Prioritization of released orders based on earliest deadlines.
- Prioritization of executing orders with pickup locations similar to the available vehicle's location (consolidation).
- If no orders are released at the current location of the available vehicle but are available elsewhere, the vehicle will travel to the closest pickup location, minimizing empty kilometers.
- Recommendation for the currently available vehicle to wait only if all non-satisfied orders are currently not released.
- Recommendation for terminating the currently available vehicle only if all orders are satisfied.

These rules can be formalized into a flowchart, guiding decision-making for the currently available vehicle in each decision epoch. It is important to note that the algorithm only makes a decision for the currently available vehicle in real-time. It does not make decision in this epoch for any other vehicle.

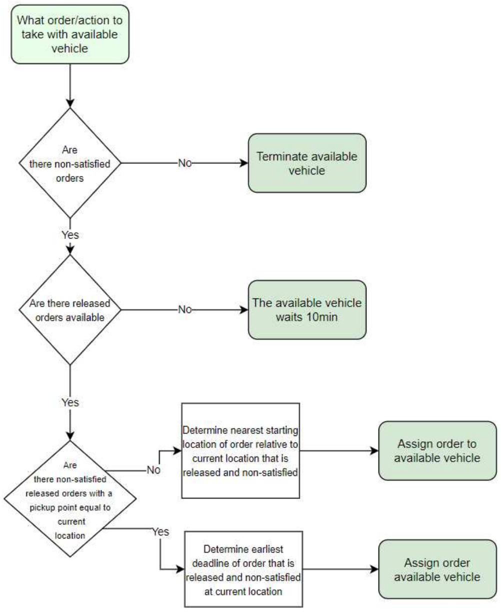

What order/action to take with available vehicle
Are there non-satisfied orders
No
Terminate available vehicle
Yes
Are there released orders available
No
The available vehicle waits 10min
Yes
Are there non-satisfied released orders with a pickup point equal to current location
No
Determine nearest starting location of order relative to current location that is released and non-satisfied
Assign order to available vehicle
Yes
Determine earliest deadline of order that is released and non-satisfied at current location
Assign order available vehicle

Figure 18: Flowchart representing the underlying logic of the consolidation algorithm and the decision flow.

### 4.5.2 Neural network agents

The DRL algorithm that is used for training the neural network is deep controlled learning (DCL). This algorithm is proposed in the paper of Temizoz et al. [49]. The algorithm is integrated into the DynaPlex toolbox and functions as the primary algorithm to train neural network agents. In this section, we outline how this method functions and why this method is used. It is important to distinguish that the DCL algorithm trains neural networks and that the decisions of these NNs are compared against the decisions of the consolidation algorithm. The DCL algorithm itself does not operate as the decision-maker.

The DCL algorithm is designed in the context of inventory control applications and offers an approach to solve MDPs with exogenous inputs using approximate policy iteration. Unlike traditional DRL methods, DCL tackles the computational challenges posed by large state spaces and uncertainty in external factors more efficiently. The approach is explained below.

First, the DCL algorithm employs parallelization techniques similar to AlphaZero to utilize the computational capabilities to sample states efficiently. By distributing simulation processes across multiple computational threads, DCL utilizes computing resources while achieving comprehensive coverage of the state space.
Second, the dcl-algorithm incorporates the Sequential Halving algorithm, which systematically allocates resources to promising actions and eliminates suboptimal ones. This approach ensures that computational resources are allocated effectively, maximizing efficiency in action selection.
The DCL algorithm also employs variance control mechanisms like Common Random Numbers to mitigate the variance in trajectory costs. By utilizing a fixed budget of exogenous scenarios, DCL ensures that computational resources are used efficiently.
Third, the DCL algorithm leverages neural networks for policy representation, enabling stable policy approximation across diverse scenarios. With sufficient samples and appropriate training, neural networks provide efficient classification-based solutions, contributing to stable training and good generalization under stochastic conditions.

By utilizing these strategies, the DCL algorithm efficiently handles large state spaces and stochastic environments, allowing for effective decision-making in real-world inventory control applications without the requirement of computational resources. Its ability to balance computational efficiency with performance optimization makes it a powerful tool for addressing complex decision-making problems in dynamic and stochastic environments. The formulation of the DCL algorithm is outlined in Algorithm 1 below. In short, the combination of state sampling, sequential halving, and neural network-based policy representation makes agents trained using DCL effective for addressing MDP problems that involve exogenous inputs.

**Algorithm 1** Deep Controlled Learning, Algorithm proposed by Temizoz et al. [49]

```
Input: MDP model: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{W}, \boldsymbol{\Xi}, f, C, \alpha, \mathbf{s}_0 \rangle$, initial policy: $\pi_0$, neural network structure: $N_\theta$, number of approximate policy iterations: $n$, number of states to be collected: $N$, number of exogenous scenarios per state-action pairs: $M$, depth of the rollouts (horizon length): $H$, length of the warmup period: $L$, number of workers: $w$
for $i = 0, 1, \ldots, n-1$ do
    Initialize $\mathcal{K}_i = \{\}$ , the dataset
    for each worker $= 1, \ldots, w$ in parallel do                          ▷ parallelization
        $\mathbf{s}_1 = \text{SampleStartState}(\mathcal{M}, \pi_i, L)$              ▷ sample starting state
        Generate exogenous scenario $\boldsymbol{\xi}$ by (8), $|\boldsymbol{\xi}| = \lceil N/w \rceil$
        for $k = 1, \ldots, \lceil N/w \rceil$ do
            Find $\hat{\pi}_i^+(\mathbf{s}_k) = \text{Simulator}(\mathcal{M}, \mathbf{s}_k, \pi_i, M, H)$       ▷ SH with CRN
            Add $(\mathbf{s}_k, \hat{\pi}_i^+(\mathbf{s}_k))$ to the data set $\mathcal{K}_i$
            $\mathbf{s}_{k+1} = f(\mathbf{s}_k, \hat{\pi}_i^+(\mathbf{s}_k), \xi_{k-1})$
        end for
    end for
    $\pi_{i+1} = \text{Classifier}(N_\theta, \mathcal{K}_i)$                      ▷ training neural networks
end for
Output: $\pi_1, \ldots, \pi_n$
```

To explicitly illustrate how neural networks, trained using the DCL algorithm, function as decision-makers in the context of the routing problem, we present an exemplary network in Figure 19. The input features represent individual state variables as described in Section 4.4.2. Therefore, the number of input features or the size of the input layer is equal to the size of the state. Each output node represents an action that the neural network could take. Thus, the size of the output layer is equal to the action space. Furthermore, in the context of neural networks, training involves adjusting the weights to train the network to output the desired output. The weights in a neural network are essentially parameters that control the strength of connections between neurons in different layers and implicitly capture relevant decision-making patterns. In summary, the neural network takes a state from the state space as input and outputs an action from the action space.

Similarly to the consolidation algorithm It is important to note that the neural network only decides what action to take for the currently available vehicle. It does not make decisions in this epoch for any other vehicle.

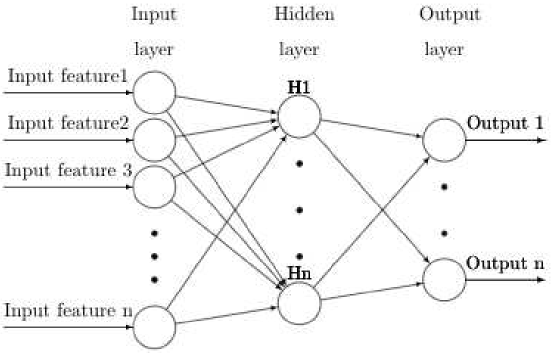


Figure 19: Schematic of feedforward neural network with an arbitrary number of input features, an arbitrary number of output features, and one hidden layer.

## 4.6 Conclusions

In this chapter, the research question "*How can we model the E2E transport network of Gam Bakker to evaluate the performance of deep reinforcement learning agents?*" is answered. We defined the mathematical formulation and notation of the E2E transport network as a vehicle routing problem with pickup and delivery (VRPPD). Subsequently, we formulated the VRPPD as an indefinite MDP following the proposed framework by Powell [2], thereby highlighting each element of the MDP individually. The complexity of the MDP lies in the fact that the specific time between adjacent stages is not known beforehand because the availability of vehicles is subjected to stochasticity. Lastly, we discussed the algorithmic framework in which we elaborated upon a rule-based algorithm functioning as a benchmark decision-maker and the deep controlled learning algorithm, which is used to train neural network agents.

# 5 Experimental design

In this chapter the aim is to elaborate upon the experimental design of this research. In Section 5.1, we shortly elaborate upon the programming software used for this research project. In Section 5.2, we discuss the methodology of testing the effectivity of neural network agents in the context of the E2E transport network. In Section 5.3, we elaborate upon the methodology of verification of the model. Most importantly, in Section 5.4, we elaborate upon the research methodology to study the effectivity of DRL in the context of the E2E transport network.

## 5.1 Programming Software

The MDP formulated in Chapter 4 has been implemented in the DynaPlex toolbox, primarily developed using the *C++* language. Furthermore, the DynaPlex environment can be built for deployment on Snellius as *Linux* code. Snellius is the Dutch supercomputer with capabilities for high-performance computing, data-processing, and machine learning. Data analysis of performance measurements and -metrics is mainly performed using *Python*.

## 5.2 Validity neural network agents

We need to validate whether neural networks can effectively operate as real-time decision agents within the simulated E2E transport network. Therefore, before training a neural network using RL, we first train multiple neural networks using supervised learning. The consolidation algorithm is used to label data during the data collection phase.

The primary objective of these experiments is to assess the accuracy of the neural network in making decisions comparable to the benchmark set by the consolidation algorithm. Achieving 100% accuracy would imply that, in all sampled state configurations, the neural network, trained through supervised learning, makes decisions identical to those made by the consolidation algorithm. Furthermore, other commonly used metrics to evaluate the performance of a neural network include the mean absolute error (MAE) and the mean squared error (MSE). However, since the absolute difference does not provide a clear indication of the decision quality we will be strictly looking at the accuracy statistic.

The second objective is to find the architecture that leads to the best classification performance of the neural networks since the architecture of a neural network contributes to the overall performance of the neural network. We focus on varying the number of hidden layers and the number of neurons per layer.

**Experimental approach**
Data is collected by sampling states using a technique similar to the technique used in the first phase of the deep controlled learning algorithm 1. Excerpts of the dataset are provided in Appendix D to offer additional context of the dataset. Determining the required number of samples for training a neural network is dependent on several factors, including the complexity of the problem, the architecture of the network, the data quality, and the desired level of performance. For this research project, we opt to train multiple networks employing a dataset comprising 25,000 diverse states representing the transport model. Initial results from this sample size exhibited promising outcomes, suggesting its adequacy for our research purposes.

The efficacy of various neural networks trained using supervised learning is evaluated by assessing their ability to predict desired outputs, as labeled by the consolidation algorithm. This evaluation determines predictive accuracy, where higher accuracy indicates the model's capability to make precise predictions and is implicitly capable of recognizing specific conditions in the state relevant to decision agents.

Additionally, we train and analyze various neural network architectures to compare their performance and assess the impact of different structures.

The following architectural factors and learning parameters hold for all supervised trained neural networks that will be tested:

- The ReLu function is used as the activation function in the hidden layers.
- The Softmax function is used as the activation function of the output layer.
- The sparse categorical cross-entropy function is used as the loss function. Sparse categorical cross-entropy is an extension of the categorical cross-entropy loss function, specifically designed for categorical classification problems where the labels are provided as integers instead of vectors.
- The number of input elements lies around the 85. The exact number of input elements is dependent on the problem instance and model parameters.
- The number of output element lies around the 25. The exact number of output elements is dependent on the problem instances.
- We employ adaptive moment estimation (Adam) [51] as an optimizer.

Additionally, we use a confusion matrix to further evaluate the neural network's performance, providing detailed insights for more precise error analysis. This matrix serves as a versatile tool, facilitating a better understanding and interpretation of neural network outcomes.

**Additional experimental approaches**
The initial results presented in Section 6.1 in Table 13 and Figure 20 suggest that there is potential to improve the overall accuracy of the trained neural networks. While our focus in Table 13 lies on adjusting model complexity to enhance performance, there are several other approaches for refining efficacy. In these additional experiments, we utilize the best-performing architecture.

We integrated (i) regularization techniques, specifically ridge regression [52], into our supervised learning approach to mitigate overfitting and improve generalization. Additionally, we focused on (ii) data augmentation by generating two new datasets. One dataset comprised 2500 samples, aimed at reducing the size of the training dataset to mitigate overfitting and enable the neural network to capture broader patterns rather than memorizing specific instances. The other dataset, encompassing 50000 samples, aimed to diversify the training set. Furthermore, we carried out (iii) hyperparameter tuning, where we varied learning rates and batch sizes to optimize model performance. This expanded approach allows for a more comprehensive exploration of methods to enhance the accuracy of our neural network trained using supervised learning.

## 5.3 Model verification

We have to verify the correctness of the MDP and ensure that the model is correctly implemented from the indefinite MDP formulation presented in Chapter 4. Verification is an important step in the design process and implicitly validates the results of the simulation study. The testing is performed in two phases; automatic and manual analysis.

DynaPlex provides automated MDP unit testing, enabling the simulation of large quantities of MDP trajectories. By analyzing individual MDP trajectories (akin to analyzing the transport operation during an individual day), we can thoroughly explore the state space, ensuring that the model handles all states appropriately, avoids transitioning into invalid states, and accurately incorporates actions recommended under random policies or the CA while not allowing invalid action to be taken.

Manual analysis complements this process by allowing for the examination of individual states and state transitions. To verify correctness, we will conduct a manual analysis on several states (transitions) by iteratively examining the following modeling aspects:

- The correctness of the costs that are incurred in the pre-decision states, post-decision states, and final state.

- The correctness of exogenous information, which involves checking whether the realized times align logically with expected order durations and the time since departure. Additionally, we must ensure that realized arrival times are conditional on the time since departure ($T_{v,b,i,j,t} \mid T_{v,b,i,j,t} \geqslant t^{\text{dep}}_{k,v}$ ).
- Adherence to the opening times of locations and the release times of orders.
- Adherence to availability of orders.

## 5.4 Simulation study

We simulate the E2E transport network within the DynaPlex environment. The main aim of the simulation study is to understand the performance of the decision-making agents within the transport network and specifically focus on the effectiveness of deep controlled learning.

In Section 5.4.1 we describe the approach to evaluate the simulated routing problem solutions. In Section 5.4.2 we describe the cost parameters utilized to train the neural network agents and express the quality of solutions for the VRPPD. In Section 5.4.3 we elaborate upon the hyperparameters that specify the configuration by which the DRL algorithm trains. In Section 5.4.4 we elaborate upon the problem instances utilized in the simulation study.

### 5.4.1 Evaluation approach

We compare the performance of the consolidation algorithm against that of the NNARLs based on the following key performance indicators: (i) the costs, (ii) the number of empty kilometers, (iii) the waiting time, and (iv) the total lateness in a single day. These performance indicators offer quantifiable metrics that help in assessing the performance of a routing solution and allow for performance comparison between varying routing solutions. Due to the stochastic nature of the transport network, conducting multiple replications of simulations is important. Therefore, we conduct 1000 replications for each combination of policy and problem instance to ensure robust evaluation. Based on the findings outlined in Section 6.3.6, it becomes evident that the number of vehicles in the problem configuration significantly influences the performance of the consolidation algorithm. Therefore, we proceed to assess the performance of both the NNARL and the consolidation algorithm in problem configurations involving three and five available vehicles.

The aforementioned metrics provide an effective evaluation of the overall performance of the two decision-making policies. However, they do not provide insight into the specific differences in actions taken by the consolidation algorithm and the NNARL. To gain deeper insights, we compare actions between the two policies in the same state. We intentionally select states from the state space that present easily understandable considerations. These states are iteratively varied in terms of current time to analyze how decision-making policies react to changes in this state variable.

Furthermore, we investigate the impact of varying the number of vehicles on overall performance by analyzing the same metrics mentioned above for both the CA and the NNARL. Unlike the initial simulation, where problem instances were varied, we maintain constant problem instances while varying the number of vehicles.

Lastly, we explore the effects of three different cost parameters on the performance of the NNARLs and the consolidation algorithm. We examine whether changing cost parameters in the problem configuration leads to significant total cost changes. We systematically vary (i) the cost per empty kilometer, (ii) the cost per waiting minute, and (iii) the cost per minute orders arrived past the deadline. We assign each cost parameter two different levels. By performing eight experiments, we can measure individual cost effects four times for each cost parameter. It is important to note that our focus is on examining individual cost effects. Our primary objective is to compare the performance of the NNARL against the consolidation algorithm.

### 5.4.2 Cost parameters

The cost parameters serve as weights for the cost function and implicate their relative importance. In practice, costs are not specifically assigned to performance characteristics. For example, there are no costs specifically related to empty kilometers or full kilometers, similarly, there is no specific cost associated with waiting. Therefore, the value of each cost parameter has to be estimated. However, based on an interview with Gam Bakker in which several propositions and considerations have been presented, we can deduct the relative importance of each cost-contributing factor. The propositions and answers of this interview can be found in Appendix G. It is important to recognize that in the current situation, the operational costs are not specifically coupled to system metrics such as empty kilometers or waiting time.

In Table 11, the exact values of the cost parameters are highlighted. There are several remarks. Firstly, the cost of driving empty is three times as high as the cost of driving full. Secondly, for each non-satisfied order, the penalty costs that are incurred are relatively large as the completion of each order is the top priority. Thirdly, there are costs for in which location a vehicle is terminated. This cost stems from the fact that vehicles have to return to Hoogtij at the end of the day. If a vehicle terminates at another location, costs have to be incurred for driving back to the original post.

Table 11: Cost parameters utilized in the simulation runs.

| **Cost** | **Price** | **Information** |
|---|---|---|
| Empty drive | 2.1 per km | Kilometers that vehicles are empty. |
| Full drive | 0.7 per km | Effective kilometers, three times as cheap. |
| Order duration | 0.828 per min | The duration of an order. |
| Past location deadline | 6.0 per min | Order is delivered past the closing time of a location. |
| Past order deadline | 3.0 per min | Order is delivered past the deadline of the order. |
| Waiting time | 2.5 per min | Vehicle is waiting. |
| Non-satisfied order | 2000 per order | Order is not satisfied during the day. |
| Ending location | 2.1 per km away from Hoogtij | Return to original post. |
| Makespan | 0.98 per hour | Incentive to end day as early as possible |

The second group of parameters can be categorized as the time- and distance parameters group. These parameters represent the expected handling times at the locations, the expected traveling times between locations, and the distance between locations. These parameters are used to sample times during the simulation. All the values of this parameter group are based on the data gathered in Chapter 2 and mostly based on historical board computer data. The matrices and vectors containing all the handling, traveling, and distance information can be found in the Appendix C.

### 5.4.3 Hyperparameters deep controlled learning

The hyperparameters used to configure the DCL algorithm stay consistent for all varying problem instances under varying modeling parameters and are presented in Table 12

Table 12: Hyperparameters used for the deep controlled algorithm in the simulation study.

| **Notation** | **Description** | **Value / Setting** |
|---|---|---|
| $\pi_0$ | Initial policy | CA |
| $N_\theta$ | Neural network structure | 258 x 128 x 128 x 64 x 64 |
| n | Number of policy iterations (generations) | 4 |
| N | Number of states to be collected | 25000 |
| M | Number of exogenous scenarios per state-action pairs | 5000 |
| H | Depth of the rollouts (horizon length) | - |
| L | Length of the warmup period | 0 |
|  | Number epochs early stopping patience | 10 |

The neural network architecture mirrors the best-performing structure identified in Section 6.1. Given the time constraint and the maximum number of orders per day, specifying a horizon length is unnecessary as these factors inherently limit the number of epochs within a single repetition of the Markov chain. Notably, the factors influencing the overall performance of NNARL the most are the number of states to be sampled ($N$) and the number of exogenous scenarios per state-action pair ($M$). Tuning these two hyperparameters ensures a balance between agent performance and computational efficiency.

### 5.4.4 Problem instances

For the simulation study, three different problem instances are utilized. Utilizing varying problem instances is common practice in simulation studies. Exposure to varying instances tests the robustness and generalizability of the DCL algorithm.

The problem instances are based on historical data and are representative of expected daily business in the proposed E2E transport network. The problem instances are formatted as a list of orders that need to be satisfied during that day. As previously described each order is characterized by a pickup location, delivery location, release time, and deadline. The problem instances are based on historical data of 14 March 2023, 12 June 2023 and 28 August 2023. The exact order information for each problem instance can be found in the Appendix B.

To clarify, each problem instance represents a unique set of orders that needs to be satisfied. We will be training neural networks on specific problem instances. Because neural network agents take the state as the input vector, and the size of the state (and therefore the size of the input vector) depends on factors such as the number of locations, number of orders, and vehicles. As a result, the neural network agent must be retrained for each specific problem instance and specific model parameter configuration. Consequently, neural network agents are only evaluated on the same problem instances it has been trained on.

## 5.5 Conclusions

In this chapter, the research question "*What experiments need to be performed to test the effectiveness of deep reinforcement learning techniques in the proposed E2E transport network model?*" is answered. We outlined the experimental setup required to validate whether neural networks could be used as decision-making agents for the E2E transport network problem. We shortly elaborated upon the testing procedure that verifies whether the MDP is correctly implemented.

Furthermore, we proposed an extensive simulation study in which a set of simulated state trajectories under both policies are compared for varying problem instances. Based on the simulated trajectories we can produce performance statistics. Furthermore, the experimental design elaborates on further evaluation techniques such as (i) in-depth analysis of state-action pairs, (ii) analysis of the impact of the number of vehicles on the overall performance, and (iii) analysis of the impact of several cost parameters on the performance of neural network agents. The overall goal of the experimental design is to facilitate extensive evaluation within the modeled E2E transport network.

# 6 Experimental results

This chapter discusses the experimental results. Firstly, we present the results regarding neural networks trained using supervised learning in Section 6.1. In Section 6.2, we shortly discuss the results regarding model verification. Lastly, we present the results regarding the performance of several neural networks trained using deep reinforcement in the context of the E2E transport network in Section 6.3.

## 6.1 Validity neural network agents

In Table 13 we present the results of the experiments designed to analyze to which degree neural network agents could replicate the decision-making behavior of the consolidation algorithm within the context of the VRPPD. We employed supervised learning to train various neural networks to replicate the action, given a specific state, labeled by the consolidation algorithm.

Table 13: The accuracy of the outcomes of neural network agents with varying neural network architectures. These results are based on 25000 samples of the problem instance of 14 March 2023, in which 80% was used as training data and 20% as testing data. 64 x 64 in the Architecture column indicates an architecture with two sequential hidden layers both containing 64 neurons.

| **Architecture** | **Hidden layers** | **Accuracy** |
|---|---|---|
| 64 | 1 | 0.731 |
| 64 x 64 | 2 | 0.742 |
| 128 x 128 | 2 | 0.753 |
| 256 x 128 x 64 | 3 | 0.755 |
| 256 x 128 x 128 x 64 x 64 | 5 | 0.768 |
| 256 x 256 x 128 x 128 x 128 x 64 x 64 | 7 | 0.751 |
| 256 x 256 x 256 x 128 x 128 x 128 x 128 x 64 x 64 | 9 | 0.702 |

Based on the presented accuracy values in Table 13 we make four observations:

- The overall performance of the neural network in replicating the decisions of the consolidation algorithm hovers around 75% across various architectures. This indicates that in 25% of the sampled states the supervised learning-trained neural network fails to mirror the desired behavior of the consolidation algorithm.
- Modifying the architectures does not yield changes in accuracy larger than 7% across various architectures.
- Contrary to expectations, increasing the depth of the neural network does not consistently lead to improved performance in this E2E transport model. It is interesting to observe that the accuracy steadily increases until 5 layers but starts to drop after that point.
- The relatively small neural network with just one hidden layer demonstrates comparable performance to larger architectures. This suggests that a compact neural network may already capture essential decision-making patterns within the E2E transport model.

Accuracy serves as an effective metric for evaluating the overall performance of the neural network, it may lack granularity in understanding the specific differences between targeted actions and predicted actions. To gain a more nuanced perspective, the results of the best-performing neural network are further examined through the confusion matrix presented in Figure 20.

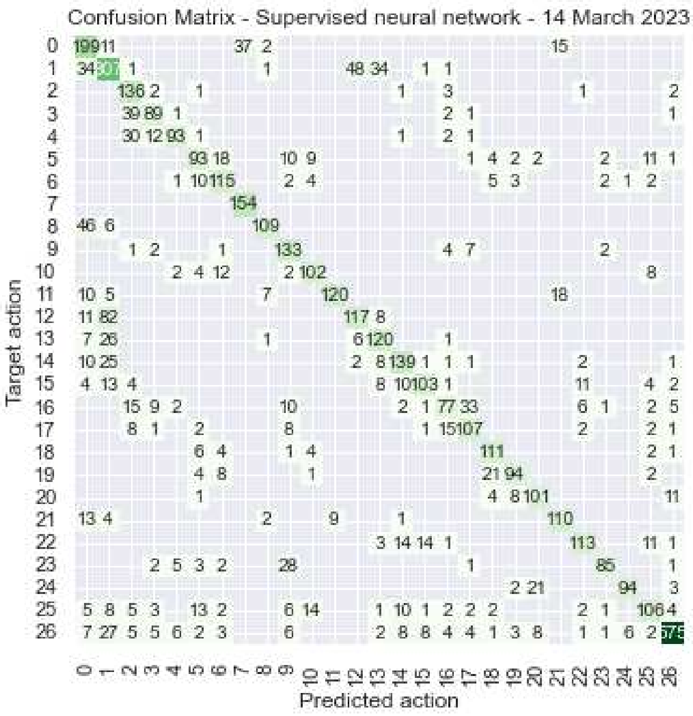


Figure 20: Confusion Matrix of the test results, 5000 samples of problem instance 14 March 2023. This matrix is used as an evaluation tool for the neural network.

| **Predicted** | **Target** | **Occurrences** |
|---|---|---|
| 7 | 0 | 37 |
| 0 | 1 | 34 |
| 2 | 3 | 39 |
| 2 | 4 | 30 |
| 0 | 8 | 46 |
| 12 | 1 | 48 |
| 1 | 12 | 82 |
| 13 | 1 | 34 |
| 17 | 16 | 33 |
| 14 | 15 | 101 |

Figure 21: Combinations with high inaccuracy occurrences between predicted actions and target actions.

In case the neural network was capable of predicting target actions with 100% accuracy, all values in the confusion matrix would align along the diagonal. The first observation from Figure 20 is that the neural network is more prone to incorrectly predict actions 25 and 26 compared to other actions. These actions represent the actions specifying that the available vehicle either has to wait or terminate. Capturing the conditions that lead to waiting or termination actions in the consolidation algorithm may be inherently more challenging for the neural network. Furthermore, other inaccuracies appear to be well-dispersed without a discernible pattern. However, certain instances of inaccuracies are recurrent. For instance, the supervised neural network agent predicted action 21 instead of targeted action 11 on 18 occasions. Consequently, in Figure 21 we present ten instances in which an incorrect prediction occurs 30 times or more. These combinations are subjected to further examination to evaluate why the neural network is consistently misclassifying these instances.

Based on the analysis of the combinations presented in Figure 21 there seem to be two categories that mainly lead to these inaccuracies.

- The first category involves pairs of similar orders with slightly differing delivery windows, such as orders 0 and 1, orders 14 and 15, or orders 16 and 17. These orders need to be transported between identical locations but differ in release times and deadlines by some hours. For instance, order 14 must be transported from the production facility to Hoogtij between 05:00 - 11:00, while order 15 has a delivery window of 08:00 - 14:00. The supervised neural network seems to struggle with distinguishing which order takes priority, unlike the consolidation algorithm.
- The second category encompasses orders with similar release times and deadlines but different routes, such as orders 7 and 0 or orders 1 and 12. For example, order 1 requires transport from Hoogtij to the production plant between 05:00 - 09:00, while order 12 needs transport from the production plant to Hoogtij, also between 05:00 - 09:00. The decision of which order to prioritize depends on the current location of the available vehicle. The underlying logic of the consolidation algorithm likely aims to minimize empty kilometers if an order is available at the current location.

**Additional results**
Data augmentation, regularization techniques, and hyperparameter tuning, aimed at improving the classification accuracy, did not yield substantial improvement. The results are presented in Table 14 and are compared against the results in Table 13. The highest accuracies, after the incorporation of the three strategies in the experimental approach, also hovered around 76%.

Table 14: The accuracy of the outcomes of neural network agents trained on data sets containing varying numbers of states of the problem instance 14 March 2023. Each neural network has an architecture of 258 x 128 x 128 x 64 x 64. Targets are generated using the consolidation algorithm. The results from neural network agents in which the training method included ridge regression are presented in accuracy regularization.

| **Number of samples** | **Accuracy** | **Accuracy regularization** |
|---|---|---|
| 2500 | 0.759 | 0.754 |
| 25000 | 0.768 | 0.769 |
| 50000 | 0.770 | 0.772 |

Changing the hyperparameter tuning did not produce any variance in the performance of neural networks. Therefore we only presented the accuracy results for neural networks trained on (i) varying sizes of samples and (ii) with or without regularization. These results are presented in Table 14. We made the following two observations based on Table 14.

- The data set containing 2500 state samples shows similar accuracies comparable to data sets with higher cardinality. This suggests that the broader decision-making patterns from the consolidation algorithm can be learned with a smaller number of samples than originally trained on.
- All accuracies are relatively close to each other. While there are differences, explicitly stating that the incorporation of one approach leads to better results seems unjustifiable.

**Interpretation**
The results suggest that various neural networks trained using supervised learning are reasonably effective in replicating the decision-making behavior of the consolidation algorithm within the E2E transport network, achieving accuracies around 75%. However, it is important to note that there is no universally accepted threshold in the literature to define what is considered “reasonable” accuracy. The adequacy of accuracy levels is contingent on the particular context and demands of the decision-making problem addressed by the neural network modeled on the heuristic.

Upon analyzing the confusion matrix and the nature of the inaccuracies, it becomes apparent that the neural network, trained through supervised learning, may not fully capture more nuanced considerations relevant to the decision-making process of the E2E transport network. Nevertheless, neural networks are inherently powerful tools for pattern recognition and can learn complex relationships within their input vector, and in this case, the state description. Based on the results in this chapter, it is evident that supervised learning may not be as effective in mimicking the consolidation algorithm and capturing all relevant decision-making conditions from the state description. Therefore, presuming that supervised learning may not suffice in fully leveraging the capabilities of neural networks, employing reinforcement learning offers an alternative.

## 6.2 Model verification

Automatic verification was performed for a hundred repetitions by utilizing the MDP unit test feature embedded in the DynaPlex toolbox. These automatic tests all ran without returning any error message. These results suggest a correct implementation of the indefinite MDP. Similarly, manual analysis of various modeling aspects confirmed satisfactory implementation, and no unexpected behavior was found.

In Appendix E we present two examples describing how we checked the correctness of four relevant modeling aspects.

## 6.3 Results Simulation

In this section we present the results of the simulation study of the E2E transportation model within the DynaPlex environment. In Section 6.3.1, we present the results regarding the average total costs of the simulated trajectories. In Section 6.3.2, we present the results regarding the average number of empty kilometers in the simulated trajectories. In Section 6.3.3, we present the results regarding the average total waiting time in the simulated trajectories. In Section 6.3.4, we present the results regarding the average total lateness in the simulated trajectories.

### 6.3.1 Cost analysis

The costs serve as the most effective metric for evaluating the performance of a policy within the transport network. We conducted multiple simulation runs on three diverse problem instances, employing both the neural network trained using the DCL algorithm as an agent and the consolidation algorithm. It is important to note that the DCL algorithm trained the neural network to minimize the costs. The cost parameters described in Section 5.4.2 and the hyperparameters described in Section 5.4.3 were utilized.

In Table 15, we present the cost analysis results of the configuration in which we use 3 vehicles to satisfy transport requests. The NNARL appears to outperform the consolidation algorithm. The cost reduction using NNARLs relative to using the CA are respectively 8.0%, 6.8%, and 13.0% for problem instances 12 June 2023, 14 March 2023, and 28 August 2023. The average performance gap is equal to 9.1%.

Table 15: Average costs and standard deviation for each real-sized problem instance using 3 vehicles utilizing two different decision-making policies. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** | **CA** | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 3320.1 ± 676.6 | 3610.5 ± 981.5 | 8.0% |
| 14 March 2023 | 5760.9 ± 2170.4 | 6184.3 ± 2260.2 | 6.8% |
| 28 August 2023 | 4224.7 ± 1131.4 | 4858.0 ± 1210.7 | 13.0% |

In Table 16, we present the cost analysis results of the configuration in which we use 5 vehicles to satisfy the transport requests. The NNARL appears to outperform the consolidation algorithm. The cost reduction using NNARLs relative to using the CA are respectively 44.3%, 20.9%, and 56.1% for problem instances 12 June 2023, 14 March 2023, and 28 August 2023. In the configuration in which we use 5 vehicles to satisfy customer demand, the average performance gap is equal to 40.4%.

Table 16: Average costs and standard deviation for each real-sized problem instance using 5 vehicles utilizing two different decision-making policies. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** | **CA** | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 3772.0 ± 995.1 | 6767.8 ± 2230.3 | 44.3% |
| 14 March 2023 | 4967.4 ± 1331.8 | 6277.8 ± 1518.5 | 20.9% |
| 28 August 2023 | 4336.8 ± 1176.2 | 9877.6 ± 3209.7 | 56.1% |

The histograms in Figure 22 corroborate the data presented in Table 15. Similarly, the histograms in Figure 23 corroborate the data presented in Table 16. The histograms provide context regarding the total costs for individual trajectories and show the distribution of the costs under both policies.

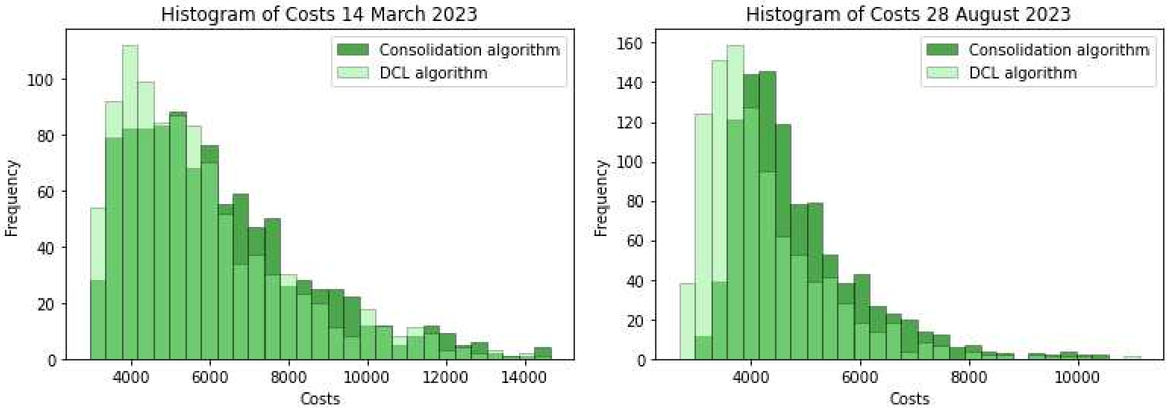


Figure 22: Histogram showing the costs made for 1000 repetitions for both decision-making algorithms in a specific problem instance. These results are generated using 3 vehicles and in accordance with the cost parameters presented in Table 11.

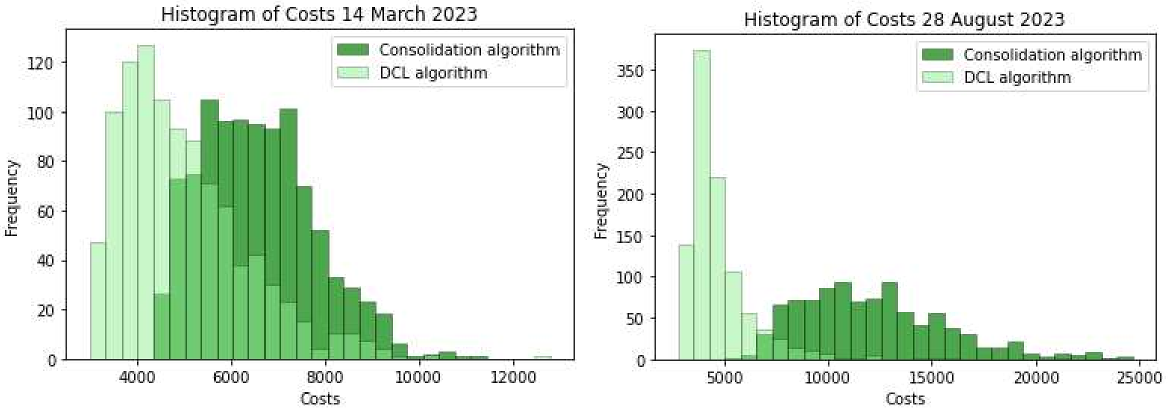


Figure 23: Histogram showing the costs made for 1000 repetitions for both decision-making algorithms in a specific problem instance. These results are generated using 5 vehicles and in accordance with the cost parameters presented in Table 11.

There are several observations made based on Table 15, Table 16, Figure 22, and Figure 23:

- In all instances and configurations, the average costs are larger for routing solutions generated by the consolidation algorithm compared to the routing solutions generated by the NNARLs. However, the performance gap is significantly smaller when the number of vehicles available in the problem configuration is three compared to five.
- In Figure 22 the variance of the costs under the consolidation algorithm and the variance of the costs under the NNARLs is almost identical. This indicates that both policies generate comparable robust results. In the case of five vehicles, we see, specifically in Figure 23, that this variance increases. This indicates that the consolidation algorithm generates less robust results in this configuration and is more dependent on stochasticity. The numerical results support this observation.
- In all four histograms the cost distribution under the NNARLs seems positively skewed. Indicating that the NNARL frequently produces routing solutions with costs below the average and less frequently have high outliers.

### 6.3.2 Empty kilometer analysis

In Table 17 and Table 18, we present the average number of empty kilometers for each policy and each real-sized problem instance based on 1000 repetitions. The results in Table 17 are based on problem configurations with three vehicles. The results in Table 18 are based on problem configuration with five vehicles.

Table 17: Average number of empty kilometers for each policy and each real-sized problem instance using 3 vehicles. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** (km) | **CA** (km) | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 377.6 ± 49.5 | 393.5 ± 65.2 | 4.0% |
| 14 March 2023 | 443.5 ± 18.1 | 446.5 ± 17.4 | 0.7% |
| 28 August 2023 | 427.7 ± 25.7 | 541.4 ± 45.7 | 21.0% |

Table 18: Average number of empty kilometers for each policy and each real-sized problem instance using 5 vehicles. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** (km) | **CA** (km) | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 412.6 ± 18.6 | 1770.5 ± 356.6 | 76.7% |
| 14 March 2023 | 523.4 ± 17.7 | 1480.9 ± 415.7 | 64.7% |
| 28 August 2023 | 445.3 ± 16.9 | 2775.1 ± 420.7 | 83.9 % |

The key observations are as follows:

- The routing solutions generated by the NNARL exhibit a reduced number of empty kilometers when contrasted with the routing solutions generated by the consolidation algorithm.
- The average performance gap between the routing solutions of the NNARL compared to those of the consolidation algorithm in the problem with three vehicles is 8.6%. In contrast, the average gap with five vehicles is equal to 75.1%. This suggests that the degree to which the NNARL leads to fewer empty kilometers compared to the consolidation algorithm is contingent on the number of vehicles.
- In Table 18, the variance of the number of empty kilometers is significantly smaller when NNARL is employed as the agent, relative to the consolidation algorithm. This indicates that the performance of the consolidation algorithm is more affected by stochasticity compared to NNARL. It is worth noting that this discrepancy is not observed in Table 17, suggesting that this affection is conditional on the number of vehicles used.
- The average number of empty kilometers in the routing solutions generated by the NNARL does not differ as much between the problem configurations with three or five available vehicles.

### 6.3.3 Total waiting time analysis

In Table 19 and Table 20, we present the average total waiting time in the routing solutions for each policy and each real-sized problem instance based on 1000 repetitions. The results in Table 19 are based on problem configurations with three vehicles. The results in Table 20 are based on problem configuration with five vehicles.

Table 19: Average total waiting time summed over all vehicles for each policy and each real-sized problem instance using 3 vehicles. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** (min) | **CA** (min) | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 152.1 ± 36.7 | 156.8 ± 42.0 | 3.0% |
| 14 March 2023 | 153.0 ± 29.6 | 153.1 ± 29.6 | 0.1% |
| 28 August 2023 | 147.4 ± 32.4 | 191.3 ± 89.0 | 23.0% |

Table 20: Average total waiting time in summed over all vehicles for each policy and each real-sized problem instance using 5 vehicles. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** (min) | **CA** (min) | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 142.4 ± 28.2 | 1019.8 ± 171.1 | 86.0% |
| 14 March 2023 | 189.1 ± 31.8 | 756.6 ± 193.3 | 75.0% |
| 28 August 2023 | 148.6 ± 30.1 | 1361.2 ± 195.2 | 89.1% |

The key observations are as follows:

- The routing solutions generated by the NNARL demonstrate a lower average waiting time compared to the routing solutions generated by the consolidation algorithm.
- The average performance gap between the routing solutions of the NNARL compared to those of the consolidation algorithm in the problem with three vehicles is 8.7%. In contrast, the average gap with five vehicles is equal to 83.4%. This suggests that the degree to which the NNARL, compared to the consolidation algorithm, leads to less total waiting time is contingent on the number of vehicles.
- The average total waiting time in the routing solutions generated by the NNARL does not differ as much between the problem configurations with three or five available vehicles.
- The results and performance gaps in total waiting time appear to align with the patterns observed in empty kilometers.

### 6.3.4 Total lateness analysis

In Table 21 and Table 22, we present the average total lateness in the routing solutions for each policy and each real-sized problem instance based on 1000 repetitions. The results in Table 21 are based on problem configurations with three vehicles. The results in Table 22 are based on problem configuration with five vehicles.

Table 21: Average total time orders were delivered past the deadline summed over all orders for each policy and each real-sized problem instance using 3 vehicles. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** (min) | **CA** (min) | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 60.1 ± 90.0 | 123.2 ± 194.5 | 51.2% |
| 14 March 2023 | 341.0 ± 352.6 | 498.0 ± 424.1 | 31.5% |
| 28 August 2023 | 85.9 ± 127.2 | 228.2 ± 250.9 | 62.4% |

Table 22: Average total time orders were delivered past the deadline summed over all orders for each policy and each real-sized problem instance using 5 vehicles. Results are based on 1000 repetitions.

| **Problem Instance** | **NNARL** (min) | **CA** (min) | **Performance gap** |
|---|---|---|---|
| 12 June 2023 | 45.2 ± 61.6 | 462.4 ± 669.3 | 90.2% |
| 14 March 2023 | 241.1 ± 249.9 | 368.8 ± 418.0 | 34.6% |
| 28 August 2023 | 79.9 ± 106.5 | 230.5 ± 267.7 | 61.1% |

The key observation is as follows:

- The routing solutions generated by the NNARL demonstrate a lower average total lateness compared to the routing solutions generated by the consolidation algorithm.
- The difference between the average performance gap in the problem configuration with three vehicles and the average performance gap in the problem configuration with five vehicles is relatively low compared to the differences seen in the empty kilometer - and total waiting time analysis.

- The routing solution generated by the NNARL in the problem configuration with five vehicles has a lower total lateness compared to the routing solution with three vehicles. This is the first metric that shows improvement with additional vehicles.
- The problem instance of 14 March contains the most transport requests, as shown in Appendix B. The routing solutions generated with five available vehicles show both the NNARL and the consolidation algorithm benefit from this additional vehicle availability. This is not the case for the other two problem instances

### 6.3.5 Action selection analysis

The performance indicators above suggest that the neural network agents trained using reinforcement learning outperform the consolidation algorithm across all problem instances to varying degrees. However, these results do not shed light on the specific differences in actions between the neural network and the consolidation algorithm. To gain deeper insights, we compare actions between the two policies by analyzing the actions in the context of several toy problems. We deliberately select states from the state space that present easily understandable considerations. Presenting the entire state of the E2E transport network in one figure is counterproductive as too many variables have to be visualized in one figure.

**Employing effective waiting actions**
The first focus point of the analysis is whether the agent is able to show "intelligent" behavior by waiting on specific release times and thereby reducing empty kilometers. The first toy problem is visualized in Figure 24. The image shows a simplified version of a state where implicitly the decision must be made for the currently available vehicle whether waiting for the release of order 5 is expected to be more beneficial than driving empty from Hoogtij to the production plant thereby instantly beginning executing order 9.

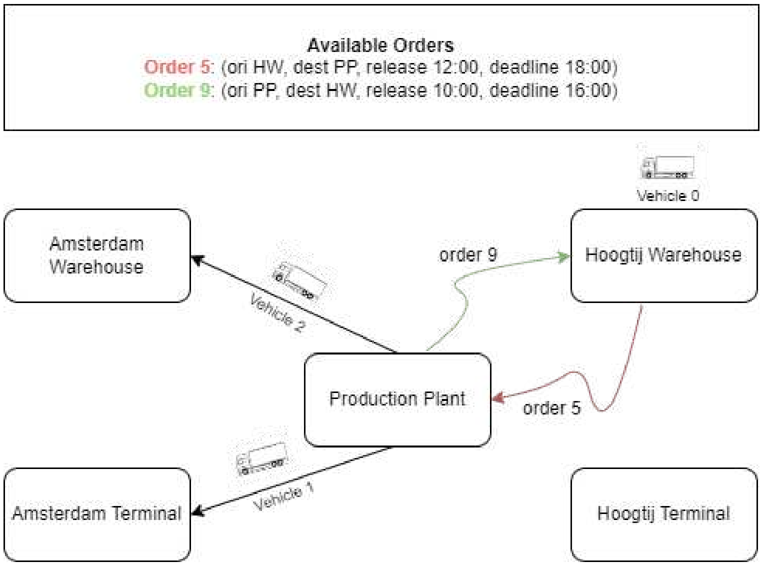


Figure 24: Toy problem 1 variant 1. In this instance, a vehicle becomes available at Hoogtij Warehouse while two other vehicles are driving towards Amsterdam terminal and Amsterdam warehouse. There are two orders left that have to be performed and a decision has to be made for the available vehicle.

The state represented by the toy problem presented in Figure 24 is iteratively modified by varying the state variable representing the current time. The prescribed actions recommended by the two policies are presented in Table 23. Remark that the prescribed decisions at 11:30 and 11:45 diverge between the neural network agent trained using RL and the consolidation algorithm. The underlying logic of the consolidation algorithm prioritizes executing released orders in all situations while the neural network seems to be able to recognize the benefits of waiting.

Table 23: Actions taken by the two policies under varying state variable CurrentTime values for toy problem 1 variant 1.

| **Current time** | 10:30 | 10:45 | 11:00 | 11:15 | 11:30 | 11:45 | 12:00 |
|---|---|---|---|---|---|---|---|
| **Action NNARL** | 9 | 9 | 9 | 9 | 26 (wait) | 26 | 5 |
| **Action CA** | 9 | 9 | 9 | 9 | 9 | 9 | 5 |

In the specific toy problem instance of Figure 24, the threshold for beneficial waiting lies between 30 minutes and 45 minutes. Intuitively it seems logical to add some form of conditional statement to the consolidation algorithm prescribing the waiting action if an order becomes available within 30 minutes at the current location.

However, it is important to remark that the effectiveness of waiting is dependent on many other state variables. For example, the location of another vehicle could affect whether waiting is effective. To illustrate this, the same toy problem presented in Figure 24 is presented in Figure 25. A slight modification is made and one vehicle that drove towards Amsterdam terminal is now driving towards the production plant.

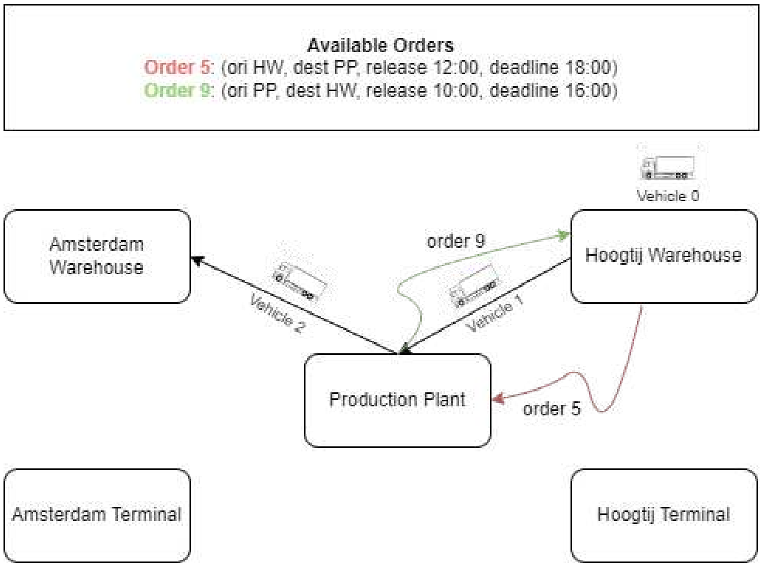


Figure 25: Toy problem 1, variant 2. In this instance, a vehicle becomes available at Hoogtij Warehouse while two other vehicles are driving towards Amsterdam terminal and the production plant. There are two orders left that have to be performed and a decision has to be made for the available vehicle.

Table 24: Actions taken by the two policies under varying state variable CurrentTime values for toy problem 1 variant 2.

| **Current time** | 10:30 | 10:45 | 11:00 | 11:15 | 11:30 | 11:45 | 12:00 |
|---|---|---|---|---|---|---|---|
| **Action NNARL** | 27 (terminate) | 27 | 27 | 26 (wait) | 26 | 26 | 5 |
| **Action CA** | 9 | 9 | 9 | 9 | 9 | 9 | 5 |

The results presented in Table 24 show that the neural network agent takes different decisions for the toy problem compared to the decisions in Table 23. The neural network agent is recommending waiting actions from 11:15 onwards. Furthermore, if the vehicle becomes available before 11:15 the decision is to terminate the available vehicle. Based on logical reasoning it seems the agent calculated that it is efficient to not assign order 9 to the available vehicle because the vehicle that is currently driving toward the production plant could handle order 9 after arrival. Moreover, after the completion of order 9, the neural network seems to expect that there is sufficient time to let the same vehicle complete order 5. Therefore, the expected penalty costs do not weigh up against the costs of waiting. Therefore, the most cost-efficient decision seems to terminate the current vehicle and let vehicle 1 satisfy the orders.

In summary, based on the analysis we found no clear threshold for efficient waiting that can be applied in the consolidation algorithm. This suggests that the cost-efficiency of waiting does not solely depend on the current time and is also conditional on other state variables. Nevertheless, in these specifically selected variants of the same toy problem, it becomes evident that the neural network trained using reinforcement learning can utilize the waiting action effectively.

**Recognizing long term effects**

The second focus point of the analysis is to assess whether the agent exhibits "intelligent" behavior by recognizing decisions that result in lower costs in the long term, even if the decision appears worse in the short term. This predictive and forward-thinking insight is a common ability of neural networks trained using reinforcement learning and is also seen in the results presented in Table 25.

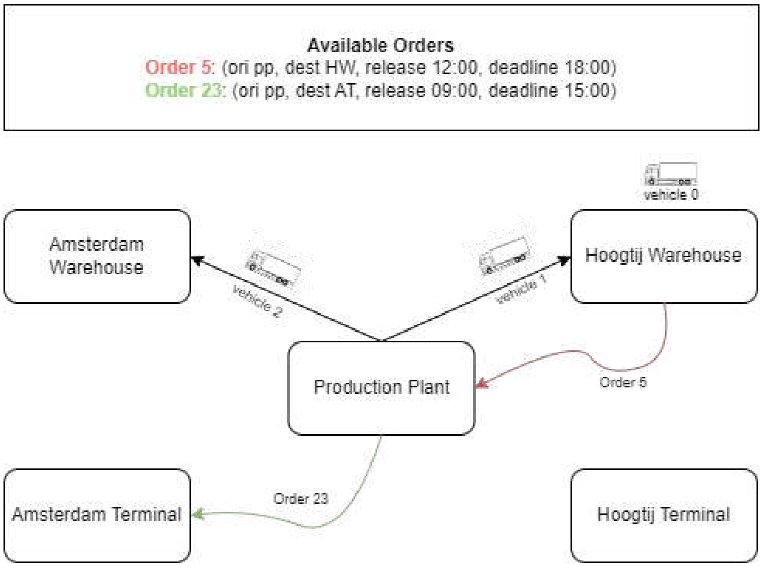


Figure 26: Toy problem 1, variant 2. In this instance, a vehicle becomes available at Hoogtij Warehouse while two other vehicles are driving towards the Amsterdam Warehouse and the Hoogtij Warehouse. There are two orders left that have to be performed and a decision has to be made for the available vehicle.

Table 25: Actions taken by the two policies under varying state variable "CurrentTime" values for toy problem 2.

| **Current Time** | 12:30 | 12:45 | 13:00 | 13:15 | 13:30 | 13:45 | 14:00 |
|---|---|---|---|---|---|---|---|
| **Action Neural Network** | 5 | 5 | 23 | 23 | 23 | 23 | 23 |
| **Action CA** | 5 | 5 | 5 | 5 | 5 | 5 | 5 |

Based on the current situation depicted in Figure 26, initially, it appears logical to execute order 5 followed by order 23, minimizing empty kilometers and making the best short-term decision. However, it is worth noting that the deadline for order 23 is set at 15:00. If order 5 is executed too close to the deadline of order 23, there is a risk of missing the deadline. It seems that the neural network anticipated this scenario and opts to execute order 23 if the current time is later than 13:00. Furthermore, another vehicle will be arriving at the Hoogtij warehouse that will be able to execute order 5 once the currently available vehicle is executing order 23.

### 6.3.6 Number of vehicles analysis

The influence of the number of available vehicles in the transport network is examined through various key performance indicators. Specifically, we analyze the average costs, average number of empty kilometers, average minutes orders were delivered past their deadline, and average waiting time summed over all vehicles.

Table 26: Average costs in E2E transport network solved with varying numbers of vehicles. Results based on 100 repetitions and problem instance 12 June 2023.

| **Vehicles** | **NNARL** | **CA** |
|---|---|---|
| 2 | 9563.8 | 11551.4 |
| 3 | 3320.1 | 3610.5 |
| 4 | 3565.5 | 4210.1 |
| 5 | 3662.1 | 6767.9 |
| 6 | 4056.9 | 10781.5 |

The analysis of Table 26 reveals a discernible impact of the number of vehicles on average costs. In the problem instance of 12 June 2023, the data suggests that employing three vehicles results in the lowest costs, while the use of two vehicles leads to a relatively large increase in costs. Table 28 indicates the number of orders delivered past their deadline (total lateness), contributing to higher penalty costs. Additionally, a noticeable trend is observed where costs under the consolidation algorithm policy increase more rapidly with an increase in the number of vehicles. This is attributed to the underlying logic, specifying that vehicles can only be terminated once all orders have been fulfilled. This suggests that the NNARL has learned in which situation terminating a vehicle has positive effects.

Table 27: Average empty kilometers in E2E transport network solved with varying numbers of vehicles. Results based on 100 repetitions and problem instance 12 June 2023.

| **Vehicles** | **NNARL** (km) | **CA** (km) |
|---|---|---|
| 2 | 276.1 | 282.6 |
| 3 | 377.6 | 393.5 |
| 4 | 404.2 | 967.8 |
| 5 | 408.5 | 1825.3 |
| 6 | 403.2 | 2480.8 |

Generally, a positive trend is observed for both policies concerning the number of empty kilometers driven and the availability of vehicles to varying degrees between the policies. The RL Neural Network policy appears to yield a relatively stable number of empty kilometers with an increasing number of vehicles. In contrast, the consolidation algorithm exhibits a tendency toward a rapid increase in the number of empty kilometers. Notably, the RL Neural Network and consolidation algorithm provide similar results for two and three vehicles. However, the difference becomes more remarkable when four or more vehicles are employed.

Table 28: Average minutes past deadline in E2E transport network solved with varying numbers of vehicles. Results based on 100 repetitions and problem instance 12 June 2023

| **Vehicles** | **NNARL** (min) | **CA** (min) |
|---|---|---|
| 2 | 1100.6 | 1578.6 |
| 3 | 60.1 | 123.2 |
| 4 | 157.8 | 87.4 |
| 5 | 87.8 | 457.8 |
| 6 | 134.4 | 1305.8 |

The notable observation is that the number of orders delivered past their respective deadlines is quite substantial when utilizing two vehicles, this observation holds for both policies. This suggests that two vehicles may not be sufficient for the transport network. Interestingly, for the NNARL, it is challenging to discern a specific trend in the results. Contrary to intuitive expectation, the average number of minutes orders are delivered past their deadlines does not exhibit a consistent decrease for the DRL policy while there are more vehicles available. Secondly, the consolidation algorithm produces less efficient results. This can be attributed to the underlying logic of the consolidation algorithm, where the location of other vehicles is not taken into consideration. As a result, sub-optimal orders are assigned to vehicles that are currently available, while the expectation is that another vehicle becomes available within the next five minutes, that is capable of delivering the specific order on time. In essence, vehicles impede each other's progress when utilizing the consolidation algorithm, especially evident with six vehicles.

Table 29: Average waiting time in E2E transport network solved with varying numbers of vehicles. Results based on 100 repetitions and problem instance 12 June 2023

| **Vehicles** | **NNARL** (min) | **CA** (min) |
|---|---|---|
| 2 | 109.3 | 106.1 |
| 3 | 152.1 | 156.8 |
| 4 | 140.0 | 438.8 |
| 5 | 145.3 | 1051.3 |
| 6 | 142.5 | 1547.6 |

The waiting time, indicative of time spent ineffectively, remains relatively stable for all vehicle configurations when utilizing the RL Neural Network agent. In contrast, the consolidation algorithm demonstrates ineffectiveness in managing a larger number of vehicles.

In summary, the number of vehicles has varying effects on the overall performance of the transport network, and increasing the number of vehicles available does not necessarily lead to better or desired behavior. Utilizing three vehicles for the problem instance of 12 June 2023 yields the best results. The results also highlight that the NNARL demonstrates better generalization for varying numbers of vehicles compared to the consolidation algorithm.

### 6.3.7 Cost parameter influence

We compared the routing solutions of eight problem configurations for the same problem instance of 12 June 2023. We iteratively examined the individual effects of three different cost parameters on key performance indicators. In Tables 30 and 31, we present the results of varying cost parameters on the average costs of the routing solutions.

Table 30: Comparison between average total costs for varying cost parameters. These results are generated with problem instance 12 June 2023, with 5 vehicles and based on 100 repetitions.

| **Cost empty km** | **Cost waiting time** | **Cost deadline miss** | **NNARL** average cost | **CA** average cost |
|---|---|---|---|---|
| 1.4 per km (-) | 1.25 per min (-) | 1.5 per min (-) | 3172.7 | 3489.8 |
| | | 3.0 per min (+) | 3316.2 | 4064.7 |
| | 2.5 per min (+) | 1.5 per min (-) | 3187.4 | 4803.5 |
| | | 3.0 per min (+) | 3584.6 | 5462.4 |
| 2.1 per km (+) | 1.25 per min (-) | 1.5 per min (-) | 3552.5 | 4779.6 |
| | | 3.0 per min (+) | 3675.0 | 5678.9 |
| | 2.5 per min (+) | 1.5 per min (-) | 3325.0 | 6107.8 |
| | | 3.0 per min (+) | 3662.1 | 6767.9 |

Table 31: Individual effect of costs parameters on the average total costs. The effect represents the average change in performance indicator due to moving a cost parameter from level "-" to level "+".

| **Cost parameter** | **NNARL** effect on costs | **CA** effect on costs |
|---|---|---|
| Empty kilometer | 238.4 | 1378.5 |
| Waiting time | 10.7 | 1282.2 |
| Too late deadline | 250.1 | 698.7 |

Based on Tables 30 and 31, we show that there is an increase in total cost when increasing a cost parameter, this holds for all three cost parameters. For example, increasing the cost parameter of driving empty from the low level of 1.4 km to the high level of 2.1 km is expected to increase the average costs by 238.4 under the NNARL. Based on the results in Table 31, the consolidation algorithm seems to be heavily influenced by any increase in cost parameter in comparison to the NNARLs. This behavior suggests that the neural networks are trained to be cost-efficient and possibly adjust decision-making patterns. In contrast, the consolidation algorithm is static and logically will not change decision-making patterns based on adjusted cost parameters.

## 6.4 Conclusions

In this chapter, the research question "*What is the performance of the agents trained using deep reinforcement learning?*" is answered. We evaluated the performance of both the neural network agents trained using deep controlled learning and the consolidation algorithm based on several key performance indicators and subsequently showed that the neural network agents trained using deep controlled learning provided better results. The numerical results show that the NNARLs are on average 9.1% more cost-efficient in routing problems with three available vehicles, and 40.4% more cost-efficient in routing problems with five available vehicles. We showed that the effectiveness of NNARL is less contingent on the problem configuration relative to the consolidation algorithm. This implies that the deep controlled algorithm is capable of producing policies that function in a wider range of problem configurations. Furthermore, we evaluated several state-action pairs in which the most cost-efficient decision was not easily determined and compared the decision of the NNARL against the decision of the consolidation algorithm. The NNARLs seemed to exhibit intelligent behavior in this analysis and seemed capable of cost-efficient considerations.

# 7 Conclusions and recommendations

This chapter answers the main research question and concludes the research in Section 7.1. In Section 7.2, we elaborate upon the limitations of this research and suggest further research directions. In Section 7.3, we make recommendations to Gam Bakker regarding their transport operations and the development of a potential planning tool. Lastly, in Section 7.4, we elaborate upon the practical- and theoretical contribution of this thesis.

## 7.1 Conclusions

To gain insight into how DRL can contribute to efficient route planning and truck scheduling in Gam Bakker's proposed transport network, we answer the main research question:

**How can deep reinforcement learning contribute to efficient route planning and truck scheduling in Gam Bakker's proposed transport network?**

Before exploring potential solutions, we contextualized the underlying research question. We began with an analysis of both the current transportation network and the proposed E2E transport network. In this analysis, we (i) outlined the existing transport network, (ii) provided insights into relevant performance metrics and modeling parameters, and (iii) shed light on the current planning methodology employed by Gam Bakker's planning team.

Additionally, we conducted a literature research. This research introduced the vehicle routing problem with pickup and delivery (VRPPD), given that the E2E transport network falls within this classification. We traced the historical development of VRPPD and its solution methodologies, contextualizing it within its historical framework to discern current trends and solution approaches. Lastly, we conducted a literature review on the application of deep reinforcement learning in route planning, aiming to understand current solution methodologies and identify any existing gaps in the literature.

To answer the research question, we developed a solution methodology to effectively train a neural network that takes the most cost-efficient decisions in real-time within a modeled E2E transport network in varying problem configurations and compared the policy against a benchmark policy. The development and testing of this methodology consisted of three phases.

**1. Modelling the proposed E2E transport network as an indefinite MDP.**
We began by formulating Gam Bakker's E2E transport network as an MDP. The underlying methodology for formulating a VRP variant with stochastic elements as an MDP is based on the formulation proposed by Steenbergen et al.[43]. Key modeling characteristics are (i) Simulation over a time horizon representative of one day in which a set number of orders needs to be satisfied. (ii) In each post-decision state we sample the arrival times of all vehicles based on the expected arrival times, the time since departure, and the waiting times. The time until the first vehicle arrives is used as the time interval to the next state. Therefore, time intervals between consecutive states are heterogeneous and dependent on stochastic factors. (iii) The modeling design allows for dynamic decision-making. This methodology of planning varies from the current methodology which utilizes a one-day-ahead approach for route planning.

We conclude that the MDP formulation appropriately represents the scoped E2E transport network under the made model assumptions. The most complex modeling step is to accurately capture the traveling- and arrival dynamics of vehicles dependent on stochastic factors. This complexity is captured within the formulation of the transition dynamics. Utilizing the principles of unit testing, various MDPs were analyzed. Pairs of consecutive states were analyzed to see if the event realization in combination with the post-decision state transitioned to logical next states. All transitions exhibited logical behavior mirroring the proposed situation.

Furthermore, we conclude that costs are correctly incurred within the model. It is important to verify whether the model correctly incurs costs similar to the conceptual formulation presented in the model design. Several MDP unit tests were performed, during each transition from state to post-decision state, each transition from post-decision state to the next state, and in the final state, associated costs are correctly incurred.

**2. Testing the capabilities of neural network agents within the context of the E2E transport network.**
We conclude that the neural networks, trained with supervised learning techniques, are not able to mirror the consolidation algorithm sufficiently well enough to achieve high accuracy values in the upper nineties. Based on further analysis of the confusion matrix and recurrent mispredictions, we observed that the neural network can recognize patterns relatively well. However, the neural network trained with supervised learning has difficulty capturing more subtle conditions required for complex considerations.

**3. Performing a simulation study in which the performance of the NNARL is compared against the performance of CA.**
We conclude that the NNARL outperforms the consolidation algorithm for various problem instances within the model of the proposed E2E transport network. The average costs made by the NNARL are significantly lower than those made under the consolidation policy. However, the performance gap between the NNARL and the consolidation algorithm varies considerably between problem configurations. We showed that the rule-based consolidation algorithm performs almost as well as the NNARL in a problem configuration with three vehicles. This is in contrast with the problem configuration of five vehicles, in which the average performance gap is significantly larger. Therefore, we conclude that the effectivity of the NNARL compared to the consolidation algorithm is contingent on the respective problem instance and problem configuration. Furthermore, we conclude that the NNARL produces more robust results compared to the CA. This is indicated by the variance between the resulting costs which is smaller under the NNARL compared to the variance under the consolidation algorithm.

To provide further insight into the composition of the resulting total costs we analyzed three performance indicators that function as cost contributors. These cost contributors are (i) the number of kilometers driven under both policies, (ii) the total waiting time, and (iii) the total lateness.

**Conclusion**
Conceptually and experimentally DRL has demonstrated its efficacy in training agents that contribute to efficient route planning and truck scheduling within Gam Bakker's proposed transport network model. The numerical results show that the NNARLs are on average 9.1% more cost-efficient in routing problems with three available vehicles, and 40.4% more cost-efficient in routing problems with five available vehicles. However, inherent limitations stemming from the model design, model properties, and research scope constrain the overall capabilities of the model and the DRL-trained agents. These limitations will be further addressed in Section 7.2. Consequently, future research could explore more complex model designs to overcome these limitations. Recommendations for further research will also be discussed in Section 7.2.

While DRL has shown promise in the simulated environment, concrete implementation steps are still required to fulfill the original research objective set by BBI of developing a planning tool. Recommendations for facilitating this development process will be outlined in Section 7.3. Additionally, concrete decision-making insights distilled from the simulation study will also be discussed in Section 7.3.

## 7.2 Limitations and further research

The theoretical results of this research are promising. However, the research has several limitations originating from causes like the scope of the research, the available research time, the lack of data representative of the proposed situation, and the overall simplifications made to model the E2E model. We discuss the limitations of the research and provide suggestions for further research below:

1. Neural networks excel at exploiting complex patterns and handling multiple objectives to make effective decisions in decision-making systems, in contrast to simple rule-based heuristics. However, this research was deliberately scoped and framed to only consider orders transported between the five predefined locations, initially aiming to provide a simple foundation for the model. Fortunately, the

formulation is extensible to more complex networks. Furthermore, the results obtained in this research align with findings in the literature, suggesting that the complexity of the scoped transport network is sufficient. This is also demonstrated in Section 6.3.5 where simple considerations were effectively dealt with by the NNARL.

In reality, Gam Bakker's clients place orders with specifications that vary more greatly compared to the modeled transport network. This means that the number of possible pickup and delivery locations is more diverse than modeled, significantly increasing the complexity of the model. For future research, it is recommended to include additional transport locations. It is expected that this expansion will widen the performance gap between NNARL and CA, further emphasizing the effectiveness of DRL.

2. The current methodology for training neural network agents involves training them on specific problem instances, resulting in agents that are tailored to a fixed set of orders and a fixed problem configuration. Consequently, if another set of orders needs to be satisfied, the same agent will not function appropriately. This issue stems from the formulation of the MDP and the input requirements of the neural network. The neural network agent takes the state as the input vector, and the size of the state (and therefore the size of the input vector) depends on factors such as the number of locations, number of orders, and vehicles. As a result, the neural network agent must be retrained for each specific problem instance and specific model configuration. To clarify, another approach to expressing the state space could enable increased generalizability.
In summary, while the model design and methodology for creating neural network agents are generalizable, the agents themselves only operate effectively on specific problem instances with specific model parameters. This limitation restricts the efficacy of individual agents, requiring users to allocate sufficient computational resources to train the agent for each new problem instance. Further research could explore the possibility of expressing transport networks, with varying modeling configurations and varying sets of orders, in a consistent state size, thereby improving the generalizability of individual trained agents.

3. The training of neural networks using RL relies on the assignment of costs to actions and the realization of travel and handling times as feedback to guide the learning process. These costs implicitly define the objective towards which the agent must strive. Accordingly, we assigned costs to various performance indicators, such as the cost per empty kilometer and the cost incurred for each minute an order exceeds its deadline. However, while this methodology is highly effective for training the neural network agent, it does not precisely mirror real-world scenarios. In practice, costs primarily encompass fuel, personnel expenses, and penalties. Nonetheless, in reinforcement learning, it is common to influence decision behavior by associating specific actions with artificially designated costs. In short, while our findings suggest the agent's cost efficiency, its performance may diverge from real-world cost-effectiveness thereby limiting the validity of the model.

4. In consideration of climate change, Gam Bakker is transitioning its vehicle fleet to incorporate more electric vehicles. Electric vehicles are characterized by their relatively limited range and prolonged charging periods compared to fossil fuel counterparts. Consequently, minimizing traveling distances and reducing charging durations during the day becomes increasingly important. To augment the realism of the transport network model, it is recommended to incorporate a model feature that stores the distances traveled and the battery life level for each vehicle. As discussed earlier, this addition is anticipated to elevate the complexity of the decision-making process. Consequently, it is expected that the performance gap between a simple rule-based heuristic, such as the CA, and the NNARL will widen, as the neural network should adeptly discern conditions where recharging is cost-effective. Furthermore, exploring the integration of this feature is marked as having priority. Following the presentation of our research findings to the Gam Bakker board, Gam Bakker highlighted that this extension should be prioritized for implementation.

5. In response to Gam Bakker's initiative to transition its fleet to electric vehicles, the board, in collaboration with the planning team, is discussing the addition of an alternative route option over the river—a ferry. The ferry offers the advantage of reducing travel distance and required battery charge, while also bypassing the frequently congested Coentunnel. However, it is worth noting that the ferry

route may entail slightly longer average travel times, although it is less susceptible to traffic jams. For future research, it is recommended to explore the addition of an extra dimension to the action space. The first dimension would represent which order will be executed similar to the current action space, and the second dimension would represent which route will be taken: the Coentunnel or the ferry.

6. The results demonstrate that NNARL outperforms the consolidation algorithm in the modeled E2E transport network. As detailed in Section 6.3.6, the cost discrepancy outlined in Table 26 varies based on the number of vehicles deployed. This suggests that the consolidation algorithm struggles to manage a fleet of five or six vehicles effectively. Conversely, when three vehicles are employed in the problem instance of June 12, 2023, the cost gap between NNARL and the consolidation algorithm is relatively small. This suggests that the consolidation algorithm may benefit from additional rules, particularly concerning efficient vehicle termination. Introducing extra rules to identify conditions where termination could be advantageous might enhance the consolidation algorithm's versatility and ability to handle diverse situations. Similarly, exploring improvements to optimize waiting actions could further improve the performance of the consolidation algorithm. In summary, enhancing the rule-based consolidation algorithm could elevate its effectiveness as a benchmark, raising the standard for performance in the environment where neural networks trained using reinforcement learning operate.

7. The DynaPlex toolbox employs a DRL algorithm known as deep controlled learning (DCL) for training neural network agents. The paper [49] introducing the DCL algorithm outlines its underlying principles, emphasizing its effectiveness in training agents that are capable of functioning in a stochastic environment. Future research could investigate the extent to which this deep controlled learning is effective. Neural networks could be trained using other commonly used DRL algorithms, such as proximal policy optimization (PPO). Ideally, comparing agents trained with different DRL frameworks would allow us to determine whether DCL is the ideal choice for addressing the stochastic exogenous factors modeled within the E2E transport network.

8. A key limitation of this research project is the absence of planning and order data representative of the E2E transport network. While order data is available from Gam Bakker between Hoogtij and the PP, and from the competitor responsible for transport between the Amsterdam locations and PP, there is a lack of data under the conditions studied in this thesis. Consequently, despite the well-reasoned setup of the model and input data, this research is constrained by the absence of real E2E transport network data, including performance data. This absence prevents a direct comparison of the dynamic approach using both the NNARL and the consolidation algorithm against a static planning approach. Therefore, while the results are well-reasoned, the research is still limited by the shortage of real data for comparison. Additionally, apart from the lack of benchmarking data, other unforeseen dynamics or oversights may not be adequately addressed. Overall, the absence of data on the proposed transport network limits the overall validity of the model and by extension the research itself.

## 7.3 Recommendations BBI and Gam Bakker

In this section, we discuss the recommendations for BBI and Gam Bakker to refine the current planning approach. In Section 7.3.1, we make recommendations regarding the development of a planning tool. In Section 7.3.2, we make general planning recommendations directly applicable to the current planning approach.

### 7.3.1 Recommendations development planning tool

The original aim set by BBI was to create a planning tool for Gam Bakker utilizing a DRL algorithm. However, during the research process, it became evident that exploring the potential contribution of DRL to efficient route planning is a considerable undertaking in itself. Additionally, developing a practical tool ready for implementation at Gam Bakker represents another substantial project. Therefore, the decision was made to primarily focus on modeling the E2E transport network and analyzing and validating the efficacy of the DRL algorithm, with less emphasis on the implementation aspect. To provide BBI and Gam Bakker with sufficient instructions for future development the following recommendations are outlined:

1. The first recommendation is focused on model improvement. We recommend three model extensions that have already been discussed in Section 7.2. While the simplified model suffices for research purposes and testing the effectiveness of DRL, these three additional problem features are required for the successful deployment of the model as an underlying planning agent.
The first extension involves adding more pickup and delivery locations in the Amsterdam area. In reality, vehicles do not solely perform orders between the locations modeled in the E2E system. To meet the actual demand of clients, the model needs to include more pickup and delivery locations.
The second extension entails incorporating electric vehicle constraints. Given Gam Bakker's focus on the energy transition, this addition is essential. Since the model currently does not consider any form of action radius, including these constraints is relevant. Recharging requires significant waiting time and could potentially lead to delays and overall increase of ineffective time usage.
The third extension involves adding extra routing options between the north and south side of Amsterdam, specifically in the form of a ferry. Aligned with the introduction of electric vehicles, the planning team is seriously considering incorporating the ferry as a route option. This route would result in fewer kilometers traveled and circumvent the Coentunnel, potentially avoiding traffic jams.

2. The second recommendation is to establish a digital environment where the neural network agent can operate and simultaneously access real-time data through APIs. In this research project, the transition dynamics within the MDP were simulated, and data were sampled from various distributions using common random numbers based on historical data, to assess the efficacy of the neural network agents. However, in reality, real-time location and activity status data from individual vehicles are available. Leveraging this real-time data is highly advantageous as it enables our agent to make dynamic decisions based on current information.
However, the challenge lies in correctly importing and formatting this real-time data. It must be formatted consistently with the input vector used to train the neural network for the specific problem instance. This ensures compatibility and effectiveness in decision-making based on real-time information.

3. Currently, the planning team adopts a one-day-ahead approach to schedule all orders, all orders are divided over all vehicles and the sequence of execution is predetermined. For example, vehicle 0 will execute orders $A$, $B$, and $C$ in order while vehicle 1 will execute orders $D$, $E$, $F$, and $G$ in order. The original concept envisioned by BBI was to integrate the neural network agent as a planning support tool, allowing it to suggest better actions when a vehicle becomes available after finishing an order, potentially deviating from the planned sequence. While this modification could reduce overall costs, it also introduces a cascading effect: changing the sequence for one vehicle also affects the sequence of another vehicle, potentially leading to less efficient transport by other vehicles. Therefore, it is important to define the exact role of the agent and consider its decisions in the context of their impact on other order sequences.

There are several methodologies to integrate the planning tool into the planning process. The ideal approach involves developing the model further to decide upon complete new sequences for all vehicles each time a vehicle becomes available, effectively re-planning the entire day each time a vehicle becomes available. However, this model redesign is complex and may not be the immediate next step in development. Instead, a more logical progression would be to introduce a threshold for improvement when recommending an action that deviates from the current sequence. Only actions expected to significantly reduce direct costs would be recommended, with planners responsible for re-planning other sequences as needed. This approach creates a planning process dependent on both the tool and the planners themselves. Lastly, Gam Bakker could choose to forego the one-day-ahead approach entirely and rely solely on the dynamic decision-making capabilities of the neural network and not plan anything.

In summary, while the theoretical results are promising, developing a planning tool requires careful consideration of its usage and the relationship between the tool and planners. We recommend establishing a clear vision for the interaction between the planning tool and planners team for effective implementation and further development of the tool. We deem the planning approach in which the model functions as a corrector most realistic. In which the current schedule for each vehicle is considered as input data and dynamic decisions are only recommended if sufficient direct costs are saved. Meanwhile, the planner is responsible for replanning the sequences if necessary. With this relationship the tools effectiveness in recognizing when direct costs savings could be made is utilized, while the responsibility for replanning the current day lays with the planners.

4. As previously mentioned in Section 7.2, training neural networks for effective decision-making demands sufficient computational resources. Due to the fluctuating sizes of the input layer, because of the varying model parameters for each day, retraining the neural network daily is necessary. Therefore, the availability of sufficient computational resources is a important condition for successful implementation. Currently, all training was performed on a HPC only accessible for academic purposes.

### 7.3.2 Recommendations planning general

General recommendations regarding the route planning of vehicles within the E2E transport network can be deduced by analyzing the decision-making behavior of the NNARL and comparing the results between the NNARL and the CA. The recommendations below will assume that there is no form of planning tool and are specific recommendations if the current method of planning is used in the E2E transport network.

1. We recommend exploring opportunities to incorporate orders from other clients traveling between Amsterdam and Hoogtij, aiming to minimize kilometers traveled. By assuming responsibility for transport routes from the production plant to the Amsterdam Terminal and Amsterdam Warehouse in the proposed E2E transport network, Gam Bakker's vehicles will handle a significant portion of Cargill's orders south of the river. However, as there are no return orders, trailers cannot be repurposed, resulting in a consistent percentage of empty kilometers despite transitioning to a dynamic approach and utilizing an effective neural network agent. To address this issue, we suggest standardizing the inclusion of orders from south of the river to Hoogtij in the planning process. Although this may not technically repurpose vehicles within the E2E network, it will decrease the number of empty kilometers and reduce the frequency of passages through the Coentunnel. In line with this recommendation, Gam Bakker could focus on proactively searching for clients that require transport movements from the south of the river in the direction of Hoogtij.

2. We recommend that planners assign hypothetical costs to performance metrics such as empty kilometers and waiting time. Currently, planners, in consultation with the respective available vehicle drivers, determine the effectiveness of waiting based on a rule of thumb: if the waiting time for an order exceeds the travel time from the current location to the pickup location where an order is available, the decision is made to start driving towards the pickup location of the order. While this rule is straightforward, we have observed that effective waiting depends on various factors. By assigning hypothetical costs to each waiting minute and cost for each kilometer driven empty, planners can make more informed decisions based on these costs.

3. The results discussed in Section 6.3.6 demonstrate the substantial influence of the number of vehicles on the average total costs. In case a set of vehicles will drive exclusively within the E2E transport, it is advisable to determine the optimal number of vehicles. Utilizing simulation can aid in calculating the most cost-effective vehicle count for that specific day.

## 7.4 Contributions

This section discusses the contribution of this research. In Section 7.4.1, we elaborate upon the theoretical contributions. In Section 7.4.2, we elaborate upon the practical contributions.

### 7.4.1 Theoretical contribution

Although our research was conducted as a case study for BBI and Cargill, its contribution mainly lies in advancing DRL methodologies within the context of route optimization and VRPs. In short, we employed a novel DRL framework called deep controlled learning to dynamically solve a homogeneous capacitated multi vehicle routing problem with pickup and delivery with time constraints and stochastic travel- and handling times. The performance of the developed agent was analyzed through a simulation study, wherein key performance indicators were evaluated over multiple repetitions.

Currently, the body of literature related to applying DRL for (dynamically) solving VRPs is fairly limited. However, several papers explore the first implementation of DRL frameworks within route optimization context. Examples of papers previously discussed that are closely related are Nazari et al. [7] and Zhang et al. [8] and Zhang et al. 2 [53]. Specifically, Zhang et al. 2 propose a DRL framework that solves a pickup and delivery problem (PDP). PDPs are characterized by a precedence relationship similar to VRPPDs. Based on experimental results they concluded that their DRL method achieves the best overall performance compared to conventional rule-based heuristics.
In the discussion of Zhang et al. regarding future research, they recommend exploring problem extensions such as multiple vehicles with capacity constraints, time window constraints, and dynamic traffic. This research project aligns well with their suggestions, as we incorporated time window constraints and stochastic traffic conditions.
Similarly, in the paper by Zhang et al. [8], the discussion about future research mentions exploring the possibilities of generalizing their DRL framework into online problems, thereby addressing real-time decision-making and stochastic travel conditions.
Furthermore, most DRL frameworks utilized in the route optimization literature are based on some variant of the DQN algorithm or an actor-critic model to train the neural network agents. This research is the first to employ deep controlled learning. Temizoz et al. [49] emphasized the need for testing the efficacy of DRL frameworks in other problem contexts, specifically focusing on inventory control.

Overall, this research contributes to the current body of literature and aligns well with the recommendations made by previous studies for future research.

### 7.4.2 Practical contribution

The original solution direction envisioned by BBI for this research was twofold. (i) Develop a planning tool capable of supporting Gam Bakker's planning team in the proposed E2E transport network as a supporting tool for their current approach to route planning. (ii) Gaining insight into the behavior of the E2E network and the effectiveness of DRL in a well-scoped transport network, while also familiarizing themselves with the DynaPlex toolbox.

**Planning tool**
We were not successful in the development of a planning tool, or even a basic version, capable of efficiently supporting Gam Bakker's planning team by recommending cost-effective actions during the transport operations planning process. This tool was intended to draw real-time data input through APIs and feature an intuitive user interface for making decisions based on real-time information. However, the formulation and integration of the MDP model into DynaPlex proved more complex and time-consuming than initially anticipated. Consequently, the practical contribution of this research is more centered around the results of the simulation study. In summary, in the context of the development of a planning tool, the practical contribution lies in the preparatory phase and recommendations on how such a planning tool could be designed.

The formulated model is generalizable to any transport network where a vendor operates based on a pull system. It can be utilized for simulation studies to analyze system behavior under various policies. However, there are several limitations, presented in Section 7.2.

Overall, while the initial objectives related to the planning tool were not fully achieved, the research provides practical insights and directions for future development in this area.

**Practical insights**
This research project does provide sufficient evidence that neural network agents trained using DRL are effective in decision-making within Gam Bakker's E2E transport network. Overall, this research demonstrates to BBI and its parent company, Bolk Transport, the potential of AI advancements in route planning. This allows the companies to assess the current usability of this technology and familiarize themselves with the underlying techniques. Assuming further advancements in AI usability in the future, BBI will have better insight into the underlying technology and principles if some form of commercial planning tool becomes available.

Lastly, based on the simulation study, we observed specific behavior and performance patterns within Gam Bakker's E2E transport network. These insights currently inform the planning process and support decision-making in the proposed transport network, representing the most direct form of practical contributions.

## Aknowlegdements

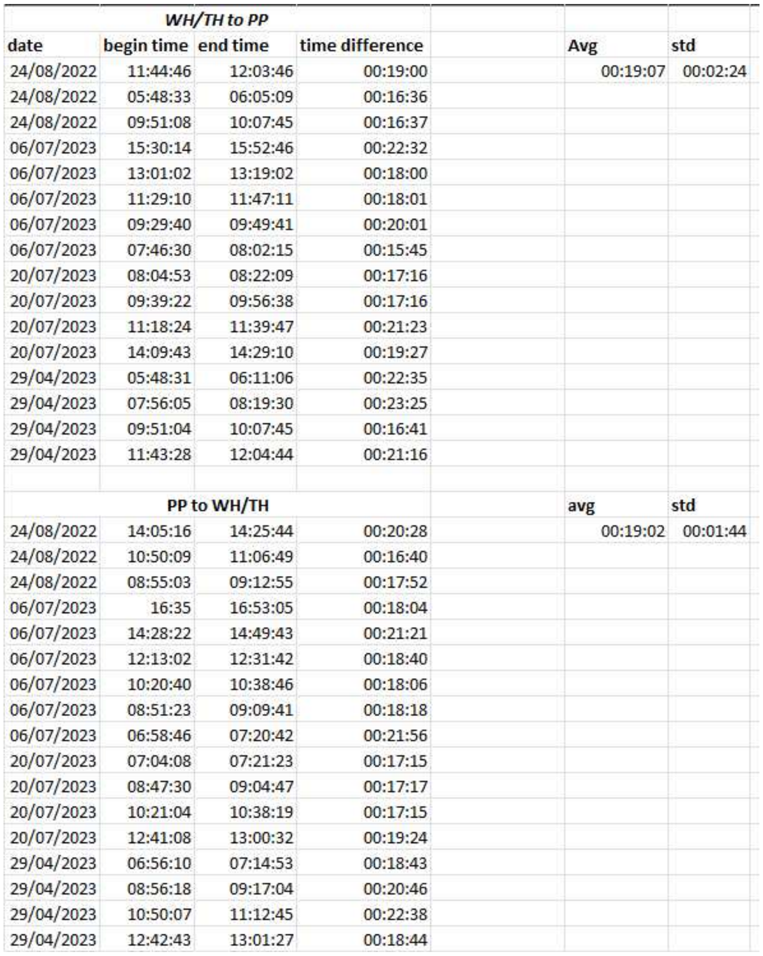
While writing this research thesis, we used ChatGPT to correct and improve academic phrasing grammatically. After using this tool, we reviewed and edited the content as needed and took full responsibility for the content of the work.

The DynaPlex tool provided a valuable framework for developing the solution methodology. The toolbox facilitated programming the Markov decision process, training the neural networks, and conducting simulation runs for routing solutions. The comprehensive documentation provided additional support.

The data provided by Gam Bakker and Cargill provided validity to this research. We leveraged historical board computer data and planning data to derive the modeling parameters and generate a realistic model.

# A Travel times analysis onboard data

We present the results of the manual analysis in Figure 27 of the board computer data, to determine the average travel time between Hoogtij and the production plant locations. We identified several transport requests over the route segment from warehouse Hoogtij (WH) to the production plant (PP), and vice versa, in the board computer data. Thereafter, we determined the departing times and arrival times and calculated the time difference. This time difference represents the realized travel time for that transport request. We determined that the average travel time lies around 19 minutes for this specific route segment.



| | WH/TH to PP | | | | | |
|---|---|---|---|---|---|---|
| date | begin time | end time | time difference | | Avg | std |
| 24/08/2022 | 11:44:46 | 12:03:46 | 00:19:00 | | 00:19:07 | 00:02:24 |
| 24/08/2022 | 05:48:33 | 06:05:09 | 00:16:36 | | | |
| 24/08/2022 | 09:51:08 | 10:07:45 | 00:16:37 | | | |
| 06/07/2023 | 15:30:14 | 15:52:46 | 00:22:32 | | | |
| 06/07/2023 | 13:01:02 | 13:19:02 | 00:18:00 | | | |
| 06/07/2023 | 11:29:10 | 11:47:11 | 00:18:01 | | | |
| 06/07/2023 | 09:29:40 | 09:49:41 | 00:20:01 | | | |
| 06/07/2023 | 07:46:30 | 08:02:15 | 00:15:45 | | | |
| 20/07/2023 | 08:04:53 | 08:22:09 | 00:17:16 | | | |
| 20/07/2023 | 09:39:22 | 09:56:38 | 00:17:16 | | | |
| 20/07/2023 | 11:18:24 | 11:39:47 | 00:21:23 | | | |
| 20/07/2023 | 14:09:43 | 14:29:10 | 00:19:27 | | | |
| 29/04/2023 | 05:48:31 | 06:11:06 | 00:22:35 | | | |
| 29/04/2023 | 07:56:05 | 08:19:30 | 00:23:25 | | | |
| 29/04/2023 | 09:51:04 | 10:07:45 | 00:16:41 | | | |
| 29/04/2023 | 11:43:28 | 12:04:44 | 00:21:16 | | | |
| | | | | | | |
| | PP to WH/TH | | | | avg | std |
| 24/08/2022 | 14:05:16 | 14:25:44 | 00:20:28 | | 00:19:02 | 00:01:44 |
| 24/08/2022 | 10:50:09 | 11:06:49 | 00:16:40 | | | |
| 24/08/2022 | 08:55:03 | 09:12:55 | 00:17:52 | | | |
| 06/07/2023 | 16:35 | 16:53:05 | 00:18:04 | | | |
| 06/07/2023 | 14:28:22 | 14:49:43 | 00:21:21 | | | |
| 06/07/2023 | 12:13:02 | 12:31:42 | 00:18:40 | | | |
| 06/07/2023 | 10:20:40 | 10:38:46 | 00:18:06 | | | |
| 06/07/2023 | 08:51:23 | 09:09:41 | 00:18:18 | | | |
| 06/07/2023 | 06:58:46 | 07:20:42 | 00:21:56 | | | |
| 20/07/2023 | 07:04:08 | 07:21:23 | 00:17:15 | | | |
| 20/07/2023 | 08:47:30 | 09:04:47 | 00:17:17 | | | |
| 20/07/2023 | 10:21:04 | 10:38:19 | 00:17:15 | | | |
| 20/07/2023 | 12:41:08 | 13:00:32 | 00:19:24 | | | |
| 29/04/2023 | 06:56:10 | 07:14:53 | 00:18:43 | | | |
| 29/04/2023 | 08:56:18 | 09:17:04 | 00:20:46 | | | |
| 29/04/2023 | 10:50:07 | 11:12:45 | 00:22:38 | | | |
| 29/04/2023 | 12:42:43 | 13:01:27 | 00:18:44 | | | |

Figure 27: Data analysis of travel times based on board computer data.

# B Problem Instances - Transport requests specifics

In this appendix, we present the three problem instances. Each problem instance contains a unique set of transport requests that need to be satisfied. These problem instances are based on planning data received from Gam Bakker and its competitor, who is currently responsible for transport between the production plant and the locations in Amsterdam.

Table 32: Transport requests in problem instance 12 June 2023. The origin represents the pickup point of the order. The destination represents the delivery location of the order. The locations are represented as follows: 0: production plant, 1: Hoogtij Warehouse, 2: Hoogtij Terminal, 3: Amsterdam Warehouse, 4: Amsterdam Terminal.

| **Order** | **Origin** | **Destination** | **Release** | **Deadline** |
|---|---|---|---|---|
| 0 | 1 | 0 | 05:00 | 07:00 |
| 1 | 1 | 0 | 05:00 | 09:00 |
| 2 | 1 | 0 | 08:00 | 12:00 |
| 3 | 1 | 0 | 08:00 | 13:00 |
| 4 | 1 | 0 | 08:00 | 13:00 |
| 5 | 1 | 0 | 12:00 | 17:00 |
| 6 | 1 | 0 | 12:00 | 17:00 |
| 7 | 0 | 1 | 05:00 | 08:00 |
| 8 | 0 | 1 | 06:00 | 10:00 |
| 9 | 0 | 1 | 06:00 | 10:00 |
| 10 | 0 | 1 | 09:00 | 14:00 |
| 11 | 0 | 1 | 09:00 | 15:00 |
| 12 | 0 | 1 | 14:00 | 20:00 |
| 13 | 0 | 3 | 05:00 | 08:00 |
| 14 | 0 | 3 | 06:00 | 10:00 |
| 15 | 0 | 3 | 06:00 | 10:00 |
| 16 | 0 | 3 | 07:00 | 11:00 |
| 17 | 0 | 3 | 09:00 | 13:00 |
| 18 | 0 | 3 | 13:00 | 16:00 |
| 19 | 0 | 3 | 13:00 | 18:00 |
| 20 | 0 | 4 | 10:00 | 15:00 |
| 21 | 0 | 4 | 10:00 | 16:00 |
| 22 | 0 | 4 | 14:00 | 19:00 |

Table 33: Orders problem instance 14 March 2023. Location numbers are similar to the abovementioned table.

| Order | Origin | Destination | Release | Deadline |
|---|---|---|---|---|
| 0 | 1 | 0 | 05:00 | 07:00 |
| 1 | 1 | 0 | 05:00 | 09:00 |
| 2 | 1 | 0 | 09:00 | 13:00 |
| 3 | 1 | 0 | 09:00 | 13:00 |
| 4 | 1 | 0 | 09:00 | 13:00 |
| 5 | 1 | 0 | 12:00 | 18:00 |
| 6 | 1 | 0 | 12:00 | 18:00 |
| 7 | 0 | 1 | 05:00 | 07:00 |
| 8 | 0 | 1 | 05:00 | 08:00 |
| 9 | 0 | 1 | 10:00 | 16:00 |
| 10 | 0 | 1 | 13:00 | 16:00 |
| 11 | 0 | 3 | 05:00 | 08:00 |
| 12 | 0 | 3 | 05:00 | 09:00 |
| 13 | 0 | 3 | 05:00 | 10:00 |
| 14 | 0 | 3 | 05:00 | 11:00 |
| 15 | 0 | 3 | 05:00 | 11:00 |
| 16 | 0 | 3 | 08:00 | 14:00 |
| 17 | 0 | 3 | 08:00 | 14:00 |
| 18 | 0 | 3 | 13:00 | 16:00 |
| 19 | 0 | 3 | 13:00 | 16:00 |
| 20 | 0 | 3 | 13:00 | 19:00 |
| 21 | 0 | 4 | 05:00 | 08:00 |
| 22 | 0 | 4 | 08:00 | 11:00 |
| 23 | 0 | 4 | 09:00 | 15:00 |
| 24 | 0 | 4 | 14:00 | 20:00 |

Table 34: Orders problem instance 28 August 2023. Location numbers are similar to the abovementioned table.

| Order | Origin | Destination | Release | Deadline |
|---|---|---|---|---|
| 0 | 0 | 1 | 05:00 | 07:00 |
| 1 | 0 | 1 | 05:00 | 09:00 |
| 2 | 0 | 1 | 08:00 | 13:00 |
| 3 | 0 | 1 | 09:00 | 13:00 |
| 4 | 0 | 1 | 10:00 | 14:00 |
| 5 | 0 | 1 | 11:00 | 16:00 |
| 6 | 0 | 1 | 16:00 | 20:00 |
| 7 | 1 | 0 | 05:00 | 07:00 |
| 8 | 1 | 0 | 08:00 | 12:00 |
| 9 | 1 | 0 | 07:00 | 12:00 |
| 10 | 1 | 0 | 12:00 | 16:00 |
| 11 | 1 | 0 | 15:00 | 19:00 |
| 12 | 0 | 3 | 05:00 | 07:00 |
| 13 | 0 | 3 | 05:00 | 08:00 |
| 14 | 0 | 3 | 05:00 | 09:00 |
| 15 | 0 | 3 | 09:00 | 13:00 |
| 16 | 0 | 3 | 09:00 | 14:00 |
| 17 | 0 | 3 | 09:00 | 15:00 |
| 18 | 0 | 3 | 13:00 | 19:00 |
| 19 | 0 | 4 | 05:00 | 08:00 |
| 20 | 0 | 4 | 05:00 | 11:00 |
| 21 | 0 | 4 | 10:00 | 16:00 |
| 22 | 0 | 4 | 12:00 | 18:00 |

# C Modeling parameters

In Table 35, the distance between all location is presented in *kilometers*. The matrix is not perfectly symmetrical because not all routes facilitate two-way traffic. Furthermore, there is no effective difference between location Hoogtij Warehouse (1) and location Hoogtij Terminal (2) regarding distance relative to the other locations. The decision is made to explicitly state them individually based on the current transport flows.

Table 35: Distance matrix between locations within the E2E transport network in *kilometers*.

| | **Production Plant (0)** | **Hoogtij Warehouse (1)** | **Hoogtij Terminal (2)** | **Amsterdam Warehouse (3)** | **Amsterdam Terminal (4)** |
|---|---|---|---|---|---|
| **Production Plant (0)** | 0 | 14.2 | 14.2 | 28.5 | 26.8 |
| **Hoogtij Warehouse (1)** | 14.2 | 0 | 0 | 22.4 | 22.2 |
| **Hoogtij Terminal (2)** | 14.2 | 0 | 0 | 22.4 | 22.2 |
| **Amsterdam Warehouse (3)** | 28.5 | 23.6 | 23,6 | 0 | 3.4 |
| **Amsterdam Terminal (4)** | 26.8 | 22.2 | 22,2 | 3.4 | 0 |

In Table 36, the expected travel times between locations within the E2E transport network are presented. These values are based on the board computer data from Gam Bakker also presented in C. The realized travel times in the simulation are based on the expected travel times presented in 36 and the seasonality factor presented in Table 37.

Table 36: Expected travel time between locations within the E2E transport network in *minutes*. The matrix is not perfectly symmetrical because some roads are one-way traffic.

| | **Production Plant (0)** | **Hoogtij Warehouse (1)** | **Hoogtij Terminal (2)** | **Amsterdam Warehouse (3)** | **Amsterdam Terminal (4)** |
|---|---|---|---|---|---|
| **Production Plant (0)** | 0 | 19 | 19 | 31 | 29 |
| **Hoogtij Warehouse (1)** | 19 | 0 | 0 | 28 | 28 |
| **Hoogtij Terminal (2)** | 19 | 0 | 0 | 28 | 28 |
| **Amsterdam Warehouse (3)** | 31 | 30 | 30 | 0 | 7 |
| **Amsterdam Terminal (4)** | 29 | 28 | 28 | 7 | 0 |

In Table 37, we present the seasonality factors per hour. These factors are used to express the relative travel times per time frame within the problem formulation.

Table 37: Travel time daily seasonality multiplication factors. These factors are multiplied by the expected travel time between locations to receive the expected travel time between locations at the specific time. The realized travel times in the simulation are dependent on the expected travel times.

| **Time Window** | **Seasonality factor** | **Time Window** | **Seasonality factor** |
|---|---|---|---|
| 05:00 - 05:59 | 0.733 | 14:00 - 14:59 | 1.05 |
| 06:00 - 06:59 | 0.767 | 15:00 - 15:59 | 1.083 |
| 07:00 - 07:59 | 0.867 | 16:00 - 16:59 | 1.117 |
| 08:00 - 08:59 | 1.133 | 17:00 - 17:59 | 1.267 |
| 09:00 - 09:59 | 1.117 | 18:00 - 18:59 | 1.183 |
| 10:00 - 10:59 | 1.0 | 19:00 - 19:59 | 1.033 |
| 11:00 - 11:59 | 1.033 | 20:00 - 20:59 | 1.0 |
| 12:00 - 12:59 | 1.033 | 21:00 - 21:59 | 0.967 |
| 13:00 - 13:59 | 1.033 | 22:00 - 22:59 | 0.867 |

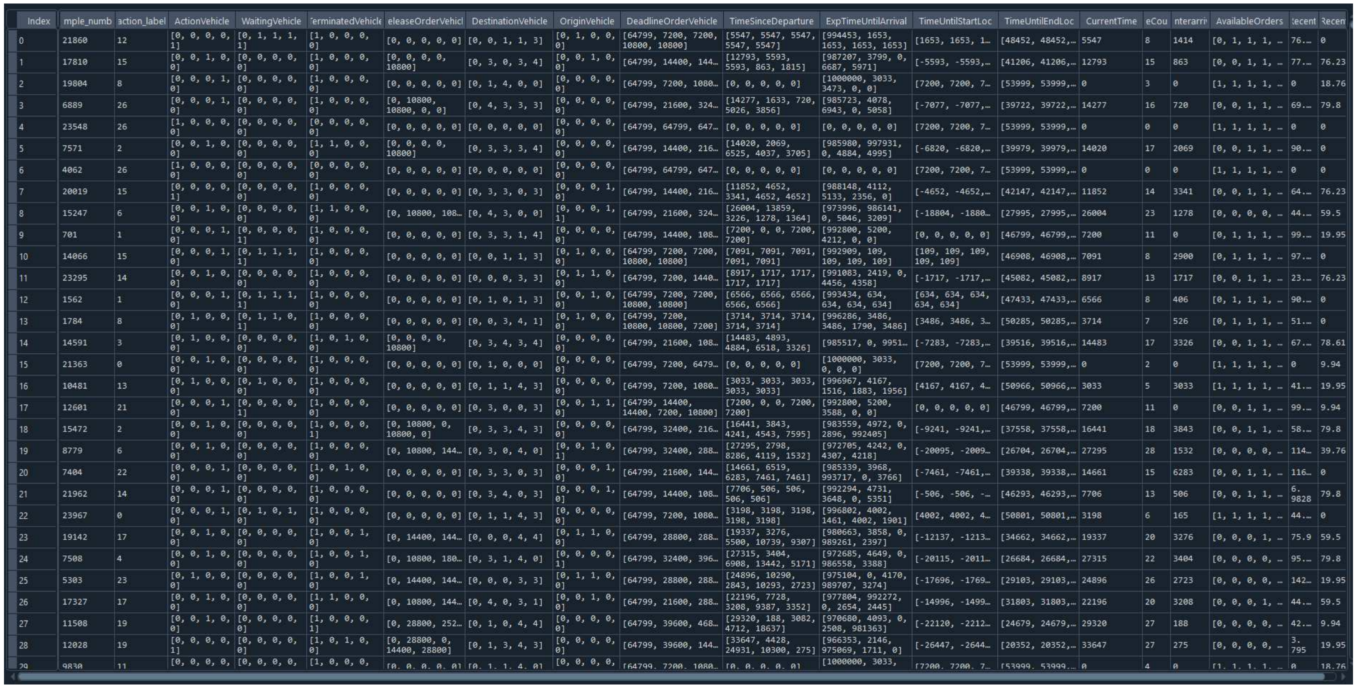
In Table 38, we present the expected handling times per location within the E2E transport network. These values are based on the board computer data from Gam Bakker.

Table 38: Handling time per location within the E2E transport network in minutes. This data is based on the board computer data.

| | **Production Plant (0)** | **Hoogtij Warehouse (1)** | **Hoogtij Terminal (2)** | **Amsterdam Warehouse (3)** | **Amsterdam Terminal (4)** |
|---|---|---|---|---|---|
| Handling Time | 20 | 30 | 30 | 30 | 30 |

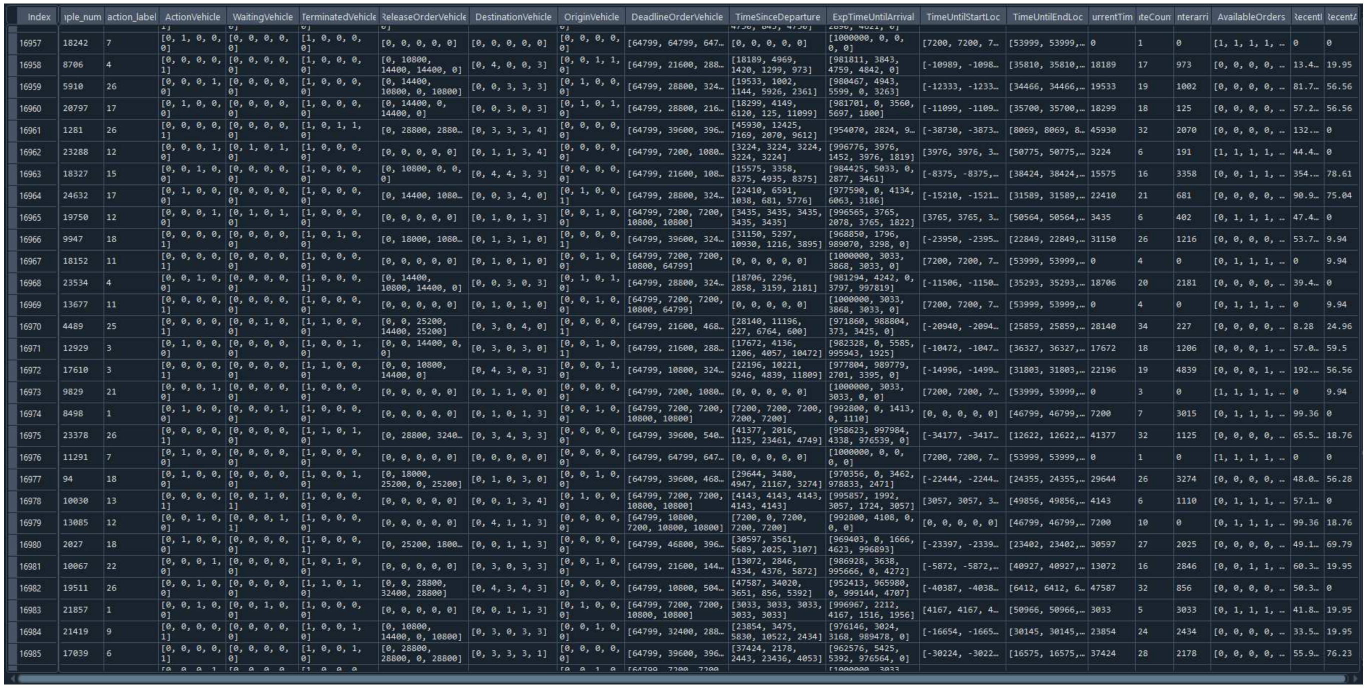
# D Example of dataset

In Figures 28 and 29, we present portions of a dataset containing 25000 samples of varying state descriptions of problem instance 14 March 2023 in a problem configuration with five available vehicles. We present these dataset portions to visualize the composition of the varying states.



| Index | mple_numb | action_label | ActionVehicle | WaitingVehicle | erminatedVehicle | eleaseOrderVehicl | DestinationVehicle | OriginVehicle | DeadlineOrderVehicle | TimeSinceDeparture | ExpTimeUntilArrival | TimeUntilStartLoc | TimeUntilEndLoc | CurrentTime | eCou | nterarri | AvailableOrders | ecent | Recen |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 21860 | 12 | [0, 0, 0, 0, 1] | [0, 1, 1, 1, 1] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 1, 1, 3] | [0, 1, 0, 0, 0] | [64799, 7200, 7200, 10800, 10800] | [5547, 5547, 5547, 5547, 5547] | [994453, 1653, 1653, 1653, 1653] | [1653, 1653, 1… | [48452, 48452,… | 5547 | 8 | 1414 | [0, 1, 1, 1, … | 76.… | 0 |
| 1 | 17810 | 15 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 10800] | [0, 3, 0, 3, 4] | [0, 0, 1, 0, 0] | [64799, 14400, 144… | [12793, 5593, 5593, 863, 1815] | [987207, 3799, 0, 6687, 5971] | [-5593, -5593,… | [41206, 41206,… | 12793 | 15 | 863 | [0, 0, 1, 1, … | 77.… | 76.23 |
| 2 | 19804 | 8 | [0, 0, 0, 1, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 4, 0, 0] | [0, 0, 0, 0, 0] | [64799, 7200, 1080… | [0, 0, 0, 0, 0] | [1000000, 3033, 3473, 0, 0] | [7200, 7200, 7… | [53999, 53999,… | 0 | 3 | 0 | [1, 1, 1, 1, … | 0 | 18.76 |
| 3 | 6889 | 26 | [0, 0, 0, 1, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 10800, 10800, 0, 0] | [0, 4, 3, 3, 3] | [0, 0, 0, 0, 0] | [64799, 21600, 324… | [14277, 1633, 720, 5026, 3856] | [985723, 4078, 6943, 0, 5058] | [-7077, -7077,… | [39722, 39722,… | 14277 | 16 | 720 | [0, 0, 1, 1, … | 69.… | 79.8 |
| 4 | 23548 | 26 | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [64799, 64799, 647… | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [7200, 7200, 7… | [53999, 53999,… | 0 | 0 | 0 | [1, 1, 1, 1, … | 0 | 0 |
| 5 | 7571 | 2 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 1, 0, 0, 0] | [0, 0, 0, 0, 10800] | [0, 3, 3, 3, 4] | [0, 0, 0, 0, 0] | [64799, 14400, 216… | [14020, 2069, 6525, 4037, 3705] | [985980, 997931, 0, 4884, 4995] | [-6820, -6820,… | [39979, 39979,… | 14020 | 17 | 2069 | [0, 0, 1, 1, … | 90.… | 0 |
| 6 | 4062 | 26 | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [64799, 64799, 647… | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [7200, 7200, 7… | [53999, 53999,… | 0 | 0 | 0 | [1, 1, 1, 1, … | 0 | 0 |
| 7 | 20019 | 15 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 3, 3, 0, 3] | [0, 0, 0, 1, 0] | [64799, 14400, 216… | [11852, 4652, 3341, 4652, 4652] | [988148, 4112, 5133, 2356, 0] | [-4652, -4652,… | [42147, 42147,… | 11852 | 14 | 3341 | [0, 0, 1, 1, … | 64.… | 76.23 |
| 8 | 15247 | 6 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 1, 0, 0, 0] | [0, 10800, 108… | [0, 4, 3, 0, 0] | [0, 0, 0, 1, 1] | [64799, 21600, 324… | [26004, 13859, 3226, 1278, 1364] | [973996, 986141, 0, 5046, 3209] | [-18804, -1880… | [27995, 27995,… | 26004 | 23 | 1278 | [0, 0, 0, 0, … | 44.… | 59.5 |
| 9 | 701 | 1 | [0, 0, 0, 1, 0] | [0, 0, 0, 0, 1] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 3, 3, 1, 4] | [0, 0, 0, 0, 0] | [64799, 14400, 108… | [7200, 0, 0, 7200, 7200] | [992800, 5200, 4212, 0, 0] | [0, 0, 0, 0, 0] | [46799, 46799,… | 7200 | 11 | 0 | [0, 1, 1, 1, … | 99.… | 19.95 |
| 10 | 14066 | 15 | [0, 0, 0, 1, 0] | [0, 1, 1, 1, 1] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 1, 1, 3] | [0, 1, 0, 0, 0] | [64799, 7200, 7200, 10800, 10800] | [7091, 7091, 7091, 7091, 7091] | [992909, 109, 109, 109, 109] | [109, 109, 109, 109, 109] | [46908, 46908,… | 7091 | 8 | 2900 | [0, 1, 1, 1, … | 97.… | 0 |
| 11 | 23295 | 14 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 3, 3] | [0, 1, 1, 0, 0] | [64799, 7200, 1440… | [8917, 1717, 1717, 1717, 1717] | [991083, 2419, 0, 4456, 4358] | [-1717, -1717,… | [45082, 45082,… | 8917 | 13 | 1717 | [0, 0, 1, 1, … | 23.… | 76.23 |
| 12 | 1562 | 1 | [0, 0, 0, 1, 0] | [0, 1, 1, 1, 1] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 0, 1, 3] | [0, 0, 1, 0, 0] | [64799, 7200, 7200, 10800, 10800] | [6566, 6566, 6566, 6566, 6566] | [993434, 634, 634, 634, 634] | [634, 634, 634, 634, 634] | [47433, 47433,… | 6566 | 8 | 406 | [0, 1, 1, 1, … | 90.… | 0 |
| 13 | 1784 | 8 | [0, 1, 0, 0, 0] | [0, 1, 1, 0, 1] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 3, 4, 1] | [0, 1, 0, 0, 0] | [64799, 7200, 10800, 10800, 7200] | [3714, 3714, 3714, 3714, 3714] | [996286, 3486, 3486, 1790, 3486] | [3486, 3486, 3… | [50285, 50285,… | 3714 | 7 | 526 | [0, 1, 1, 1, … | 51.… | 0 |
| 14 | 14591 | 3 | [0, 1, 0, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 1, 0, 0] | [0, 0, 0, 0, 10800] | [0, 3, 4, 3, 4] | [0, 0, 0, 0, 0] | [64799, 21600, 108… | [14483, 4893, 4884, 6518, 3326] | [985517, 0, 9951… | [-7283, -7283,… | [39516, 39516,… | 14483 | 17 | 3326 | [0, 0, 1, 1, … | 67.… | 78.61 |
| 15 | 21363 | 0 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 0, 0, 0] | [0, 0, 0, 0, 0] | [64799, 7200, 6479… | [0, 0, 0, 0, 0] | [1000000, 3033, 0, 0, 0] | [7200, 7200, 7… | [53999, 53999,… | 0 | 2 | 0 | [1, 1, 1, 1, … | 0 | 9.94 |
| 16 | 10481 | 13 | [0, 1, 0, 0, 0] | [0, 1, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 1, 4, 3] | [0, 0, 0, 0, 0] | [64799, 7200, 1080… | [3033, 3033, 3033, 3033, 3033] | [996967, 4167, 1516, 1883, 1956] | [4167, 4167, 4… | [50966, 50966,… | 3033 | 5 | 3033 | [1, 1, 1, 1, … | 41.… | 19.95 |
| 17 | 12601 | 21 | [0, 0, 0, 1, 0] | [0, 0, 0, 0, 1] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 3, 0, 0, 3] | [0, 0, 1, 1, 0] | [64799, 14400, 14400, 7200, 10800] | [7200, 0, 0, 7200, 7200] | [992800, 5200, 3588, 0, 0] | [0, 0, 0, 0, 0] | [46799, 46799,… | 7200 | 11 | 0 | [0, 0, 1, 1, … | 99.… | 9.94 |
| 18 | 15472 | 2 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 1] | [0, 10800, 0, 10800, 0] | [0, 3, 3, 4, 3] | [0, 0, 0, 0, 0] | [64799, 32400, 216… | [16441, 3843, 4241, 4543, 7595] | [983559, 4972, 0, 2896, 992405] | [-9241, -9241,… | [37558, 37558,… | 16441 | 18 | 3843 | [0, 0, 1, 1, … | 58.… | 79.8 |
| 19 | 8779 | 6 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 10800, 144… | [0, 3, 0, 4, 0] | [0, 0, 1, 0, 1] | [64799, 32400, 288… | [27295, 2798, 8286, 4119, 1532] | [972705, 4242, 0, 4307, 4218] | [-20095, -2009… | [26704, 26704,… | 27295 | 28 | 1532 | [0, 0, 0, 0, … | 114… | 39.76 |
| 20 | 7404 | 22 | [0, 0, 0, 1, 0] | [0, 0, 0, 0, 0] | [1, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [0, 3, 3, 0, 3] | [0, 0, 0, 1, 0] | [64799, 21600, 144… | [14661, 6519, 6283, 7461, 7461] | [985339, 3968, 993717, 0, 3766] | [-7461, -7461,… | [39338, 39338,… | 14661 | 15 | 6283 | [0, 0, 1, 1, … | 116… | 0 |
| 21 | 21962 | 14 | [0, 0, 0, 1, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 3, 4, 0, 3] | [0, 0, 0, 1, 0] | [64799, 14400, 108… | [7706, 506, 506, 506, 506] | [992294, 4731, 3648, 0, 5351] | [-506, -506, -… | [46293, 46293,… | 7706 | 13 | 506 | [0, 0, 1, 1, … | 6. 9828 | 79.8 |
| 22 | 23967 | 0 | [0, 0, 0, 1, 0] | [0, 1, 0, 1, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 1, 4, 3] | [0, 0, 0, 0, 0] | [64799, 7200, 1080… | [3198, 3198, 3198, 3198, 3198] | [996802, 4002, 1461, 4002, 1901] | [4002, 4002, 4… | [50801, 50801,… | 3198 | 6 | 165 | [1, 1, 1, 1, … | 44.… | 0 |
| 23 | 19142 | 17 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 1, 0] | [0, 14400, 144… | [0, 0, 0, 4, 4] | [0, 1, 1, 0, 0] | [64799, 28800, 288… | [19337, 3276, 5500, 10739, 9307] | [980663, 3858, 0, 989261, 2397] | [-12137, -1213… | [34662, 34662,… | 19337 | 20 | 3276 | [0, 0, 0, 1, … | 75.9 | 59.5 |
| 24 | 7508 | 4 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 1, 0] | [0, 10800, 180… | [0, 3, 1, 4, 0] | [0, 0, 0, 0, 1] | [64799, 32400, 396… | [27315, 3404, 6908, 13442, 5171] | [972685, 4649, 0, 986558, 3388] | [-20115, -2011… | [26684, 26684,… | 27315 | 22 | 3404 | [0, 0, 0, 0, … | 95.… | 79.8 |
| 25 | 5303 | 23 | [0, 1, 0, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 1, 0] | [0, 14400, 144… | [0, 0, 0, 3, 3] | [0, 1, 1, 0, 0] | [64799, 28800, 288… | [24896, 10290, 2843, 10293, 2723] | [975104, 0, 4170, 989707, 3274] | [-17696, -1769… | [29103, 29103,… | 24896 | 26 | 2723 | [0, 0, 0, 0, … | 142… | 19.95 |
| 26 | 17327 | 17 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 1, 0, 0, 0] | [0, 10800, 144… | [0, 4, 0, 3, 1] | [0, 0, 1, 0, 0] | [64799, 21600, 288… | [22196, 7728, 3208, 9387, 3352] | [977804, 992272, 0, 2654, 2445] | [-14996, -1499… | [31803, 31803,… | 22196 | 20 | 3208 | [0, 0, 0, 1, … | 44.… | 59.5 |
| 27 | 11508 | 19 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 1] | [0, 28800, 252… | [0, 1, 0, 4, 4] | [0, 0, 0, 0, 0] | [64799, 39600, 468… | [29320, 188, 3082, 4712, 18637] | [970680, 4093, 0, 2508, 981363] | [-22120, -2212… | [24679, 24679,… | 29320 | 27 | 188 | [0, 0, 0, 0, … | 42.… | 9.94 |
| 28 | 12028 | 19 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 0, 1, 0, 0] | [0, 28800, 0, 14400, 28800] | [0, 1, 3, 4, 3] | [0, 0, 0, 0, 0] | [64799, 39600, 144… | [33647, 4428, 24931, 10300, 275] | [966353, 2146, 975069, 1711, 0] | [-26447, -2644… | [20352, 20352,… | 33647 | 27 | 275 | [0, 0, 0, 0, … | 3. 795 | 19.95 |
| 29 | 9830 | 11 | [0, 0, 0, 0, | [0, 0, 0, 0, | [1, 0, 0, 0, | [0, 0, 0, 0, 0] | [0, 1, 1, 4, 0] | [0, 0, 0, 0, | [64799, 7200, 1080… | [0, 0, 0, 0, 0] | [1000000, 3033, | [7200, 7200, 7… | [53999, 53999,… | 0 | 4 | 0 | [1, 1, 1, 1, … | 0 | 18.76 |

Figure 28: Snippet of a DataFrame comprising 25000 samples from a problem instance recorded on March 14, 2023. Each row corresponds to a distinct sample, while each column represents a state variable, expressed either as an integer value or a vector.



| Index | ple_num | action_label | ActionVehicle | WaitingVehicle | TerminatedVehicle | ReleaseOrderVehicle | DestinationVehicle | OriginVehicle | DeadlineOrderVehicle | TimeSinceDeparture | ExpTimeUntilArrival | TimeUntilStartLoc | TimeUntilEndLoc | urrentTim | iteCoun | nterarri | AvailableOrders | Recentl | RecentA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16957 | 18242 | 7 | [0, 1, 0, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [64799, 64799, 647… | [0, 0, 0, 0, 0] | [1000000, 0, 0, 0, 0] | [7200, 7200, 7… | [53999, 53999,… | 0 | 1 | 0 | [1, 1, 1, 1, … | 0 | 0 |
| 16958 | 8706 | 4 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 10800, 14400, 14400, 0] | [0, 4, 0, 0, 3] | [0, 0, 1, 1, 0] | [64799, 21600, 288… | [18189, 4969, 1420, 1299, 973] | [981811, 3843, 4759, 4842, 0] | [-10989, -1098… | [35810, 35810,… | 18189 | 17 | 973 | [0, 0, 0, 0, … | 13.4… | 19.95 |
| 16959 | 5910 | 26 | [0, 0, 0, 1, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 14400, 10800, 0, 10800] | [0, 0, 3, 3, 3] | [0, 1, 0, 0, 0] | [64799, 28800, 324… | [19533, 1002, 1144, 5926, 2361] | [980467, 4943, 5599, 0, 3263] | [-12333, -1233… | [34466, 34466,… | 19533 | 19 | 1002 | [0, 0, 0, 0, … | 81.7… | 56.56 |
| 16960 | 20797 | 17 | [0, 1, 0, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 14400, 0, 14400, 0] | [0, 0, 3, 0, 3] | [0, 1, 0, 1, 0] | [64799, 28800, 216… | [18299, 4149, 6120, 125, 11099] | [981701, 0, 3560, 5697, 1800] | [-11099, -1109… | [35700, 35700,… | 18299 | 18 | 125 | [0, 0, 0, 0, … | 57.2… | 56.56 |
| 16961 | 1281 | 26 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 0, 1, 1, 0] | [0, 28800, 2880… | [0, 3, 3, 3, 4] | [0, 0, 0, 0, 0] | [64799, 39600, 396… | [45930, 12425, 7169, 2070, 9612] | [954070, 2824, 9… | [-38730, -3873… | [8069, 8069, 8… | 45930 | 32 | 2070 | [0, 0, 0, 0, … | 132.… | 0 |
| 16962 | 23288 | 12 | [0, 0, 0, 1, 0] | [0, 1, 0, 1, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 1, 3, 4] | [0, 0, 0, 0, 0] | [64799, 7200, 1080… | [3224, 3224, 3224, 3224, 3224] | [996776, 3976, 1452, 3976, 1819] | [3976, 3976, 3… | [50775, 50775,… | 3224 | 6 | 191 | [1, 1, 1, 1, … | 44.4… | 0 |
| 16963 | 18327 | 15 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 10800, 0, 0, 0] | [0, 4, 4, 3, 3] | [0, 0, 0, 0, 0] | [64799, 21600, 108… | [15575, 3358, 8375, 4935, 8375] | [984425, 5033, 0, 2877, 3461] | [-8375, -8375,… | [38424, 38424,… | 15575 | 16 | 3358 | [0, 0, 1, 1, … | 354.… | 78.61 |
| 16964 | 24632 | 17 | [0, 1, 0, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 14400, 1080… | [0, 0, 3, 4, 0] | [0, 1, 0, 0, 1] | [64799, 28800, 324… | [22410, 6591, 1038, 681, 5776] | [977590, 0, 4134, 6063, 3186] | [-15210, -1521… | [31589, 31589,… | 22410 | 21 | 681 | [0, 0, 0, 0, … | 90.9… | 75.04 |
| 16965 | 19750 | 12 | [0, 0, 0, 1, 0] | [0, 1, 0, 1, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 0, 1, 3] | [0, 0, 1, 0, 0] | [64799, 7200, 7200, 10800, 10800] | [3435, 3435, 3435, 3435, 3435] | [996565, 3765, 2078, 3765, 1822] | [3765, 3765, 3… | [50564, 50564,… | 3435 | 6 | 402 | [0, 1, 1, 1, … | 47.4… | 0 |
| 16966 | 9947 | 18 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 0, 1, 0, 0] | [0, 18000, 1080… | [0, 1, 3, 1, 0] | [0, 0, 0, 0, 1] | [64799, 39600, 324… | [31150, 5297, 10930, 1216, 3895] | [968850, 1796, 989070, 3298, 0] | [-23950, -2395… | [22849, 22849,… | 31150 | 26 | 1216 | [0, 0, 0, 0, … | 53.7… | 9.94 |
| 16967 | 18152 | 11 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 0, 1, 0] | [0, 0, 1, 0, 0] | [64799, 7200, 7200, 10800, 64799] | [0, 0, 0, 0, 0] | [1000000, 3033, 3868, 3033, 0] | [7200, 7200, 7… | [53999, 53999,… | 0 | 4 | 0 | [0, 1, 1, 1, … | 0 | 9.94 |
| 16968 | 23534 | 4 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 1] | [0, 14400, 10800, 14400, 0] | [0, 0, 3, 0, 3] | [0, 1, 0, 1, 0] | [64799, 28800, 324… | [18706, 2296, 2858, 3159, 2181] | [981294, 4242, 0, 3797, 997819] | [-11506, -1150… | [35293, 35293,… | 18706 | 20 | 2181 | [0, 0, 0, 0, … | 39.4… | 0 |
| 16969 | 13677 | 11 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 0, 1, 0] | [0, 0, 1, 0, 0] | [64799, 7200, 7200, 10800, 64799] | [0, 0, 0, 0, 0] | [1000000, 3033, 3868, 3033, 0] | [7200, 7200, 7… | [53999, 53999,… | 0 | 4 | 0 | [0, 1, 1, 1, … | 0 | 9.94 |
| 16970 | 4489 | 25 | [0, 0, 0, 0, 1] | [0, 0, 1, 0, 0] | [1, 1, 0, 0, 0] | [0, 0, 25200, 14400, 25200] | [0, 3, 0, 4, 0] | [0, 0, 0, 0, 1] | [64799, 21600, 468… | [28140, 11196, 227, 6764, 600] | [971860, 988804, 373, 3425, 0] | [-20940, -2094… | [25859, 25859,… | 28140 | 34 | 227 | [0, 0, 0, 0, … | 8.28 | 24.96 |
| 16971 | 12929 | 3 | [0, 1, 0, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 1, 0] | [0, 0, 14400, 0, 0] | [0, 3, 0, 3, 0] | [0, 0, 1, 0, 1] | [64799, 21600, 288… | [17672, 4136, 1206, 4057, 10472] | [982328, 0, 5585, 995943, 1925] | [-10472, -1047… | [36327, 36327,… | 17672 | 18 | 1206 | [0, 0, 0, 1, … | 57.0… | 59.5 |
| 16972 | 17610 | 3 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 1, 0, 0, 0] | [0, 0, 10800, 14400, 0] | [0, 4, 3, 0, 3] | [0, 0, 0, 1, 0] | [64799, 10800, 324… | [22196, 10221, 9246, 4839, 11809] | [977804, 989779, 2701, 3395, 0] | [-14996, -1499… | [31803, 31803,… | 22196 | 19 | 4839 | [0, 0, 0, 1, … | 192.… | 56.56 |
| 16973 | 9829 | 21 | [0, 0, 0, 1, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 1, 0, 0] | [0, 0, 0, 0, 0] | [64799, 7200, 1080… | [0, 0, 0, 0, 0] | [1000000, 3033, 3033, 0, 0] | [7200, 7200, 7… | [53999, 53999,… | 0 | 3 | 0 | [1, 1, 1, 1, … | 0 | 9.94 |
| 16974 | 8498 | 1 | [0, 1, 0, 0, 0] | [0, 0, 0, 1, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 1, 0, 1, 3] | [0, 0, 1, 0, 0] | [64799, 7200, 7200, 10800, 10800] | [7200, 7200, 7200, 7200, 7200] | [992800, 0, 1413, 0, 1110] | [0, 0, 0, 0, 0] | [46799, 46799,… | 7200 | 7 | 3015 | [0, 1, 1, 1, … | 99.36 | 0 |
| 16975 | 23378 | 26 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 1, 0, 1, 0] | [0, 28800, 3240… | [0, 3, 4, 3, 3] | [0, 0, 0, 0, 0] | [64799, 39600, 540… | [41377, 2016, 1125, 23461, 4749] | [958623, 997984, 4338, 976539, 0] | [-34177, -3417… | [12622, 12622,… | 41377 | 32 | 1125 | [0, 0, 0, 0, … | 65.5… | 18.76 |
| 16976 | 11291 | 7 | [0, 1, 0, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [64799, 64799, 647… | [0, 0, 0, 0, 0] | [1000000, 0, 0, 0, 0] | [7200, 7200, 7… | [53999, 53999,… | 0 | 1 | 0 | [1, 1, 1, 1, … | 0 | 0 |
| 16977 | 94 | 18 | [0, 1, 0, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 1, 0] | [0, 18000, 25200, 0, 25200] | [0, 1, 0, 3, 0] | [0, 0, 1, 0, 0] | [64799, 39600, 468… | [29644, 3480, 4947, 21167, 3274] | [970356, 0, 3462, 978833, 2471] | [-22444, -2244… | [24355, 24355,… | 29644 | 26 | 3274 | [0, 0, 0, 0, … | 48.0… | 56.28 |
| 16978 | 10030 | 13 | [0, 0, 0, 0, 1] | [0, 0, 1, 0, 1] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 1, 3, 4] | [0, 1, 0, 0, 0] | [64799, 7200, 7200, 10800, 10800] | [4143, 4143, 4143, 4143, 4143] | [995857, 1992, 3057, 1724, 3057] | [3057, 3057, 3… | [49856, 49856,… | 4143 | 6 | 1110 | [0, 1, 1, 1, … | 57.1… | 0 |
| 16979 | 13085 | 12 | [0, 0, 1, 0, 0] | [0, 0, 0, 1, 1] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 4, 1, 1, 3] | [0, 0, 0, 0, 0] | [64799, 10800, 7200, 10800, 10800] | [7200, 0, 7200, 7200, 7200] | [992800, 4108, 0, 0, 0] | [0, 0, 0, 0, 0] | [46799, 46799,… | 7200 | 10 | 0 | [0, 1, 1, 1, … | 99.36 | 18.76 |
| 16980 | 2027 | 18 | [0, 1, 0, 0, 0] | [0, 0, 0, 0, 0] | [1, 0, 0, 0, 1] | [0, 25200, 1800… | [0, 0, 1, 1, 3] | [0, 0, 0, 0, 0] | [64799, 46800, 396… | [30597, 3561, 5689, 2025, 3107] | [969403, 0, 1666, 4623, 996893] | [-23397, -2339… | [23402, 23402,… | 30597 | 27 | 2025 | [0, 0, 0, 0, … | 49.1… | 69.79 |
| 16981 | 10067 | 22 | [0, 0, 0, 1, 0] | [0, 0, 0, 0, 0] | [1, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [0, 3, 0, 3, 3] | [0, 0, 1, 0, 0] | [64799, 21600, 144… | [13072, 2846, 4334, 4376, 5872] | [986928, 3638, 995666, 0, 4272] | [-5872, -5872,… | [40927, 40927,… | 13072 | 16 | 2846 | [0, 0, 1, 1, … | 60.3… | 19.95 |
| 16982 | 19511 | 26 | [0, 0, 1, 0, 0] | [0, 0, 0, 0, 0] | [1, 1, 0, 1, 0] | [0, 0, 28800, 32400, 28800] | [0, 4, 3, 4, 3] | [0, 0, 0, 0, 0] | [64799, 10800, 504… | [47587, 34020, 3651, 856, 5392] | [952413, 965980, 0, 999144, 4707] | [-40387, -4038… | [6412, 6412, 6… | 47587 | 32 | 856 | [0, 0, 0, 0, … | 50.3… | 0 |
| 16983 | 21857 | 1 | [0, 0, 1, 0, 0] | [0, 0, 1, 0, 0] | [1, 0, 0, 0, 0] | [0, 0, 0, 0, 0] | [0, 0, 1, 1, 3] | [0, 1, 0, 0, 0] | [64799, 7200, 7200, 10800, 10800] | [3033, 3033, 3033, 3033, 3033] | [996967, 2212, 4167, 1516, 1956] | [4167, 4167, 4… | [50966, 50966,… | 3033 | 5 | 3033 | [0, 1, 1, 1, … | 41.8… | 19.95 |
| 16984 | 21419 | 9 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 0, 0, 1, 0] | [0, 10800, 14400, 0, 10800] | [0, 3, 0, 3, 3] | [0, 0, 1, 0, 0] | [64799, 32400, 288… | [23854, 3475, 5830, 10522, 2434] | [976146, 3024, 3168, 989478, 0] | [-16654, -1665… | [30145, 30145,… | 23854 | 24 | 2434 | [0, 0, 0, 0, … | 33.5… | 19.95 |
| 16985 | 17039 | 6 | [0, 0, 0, 0, 1] | [0, 0, 0, 0, 0] | [1, 0, 0, 1, 0] | [0, 28800, 28800, 0, 28800] | [0, 3, 3, 3, 1] | [0, 0, 0, 0, 0] | [64799, 39600, 396… | [37424, 2178, 2443, 23436, 4053] | [962576, 5425, 5392, 976564, 0] | [-30224, -3022… | [16575, 16575,… | 37424 | 28 | 2178 | [0, 0, 0, 0, … | 55.9… | 76.23 |

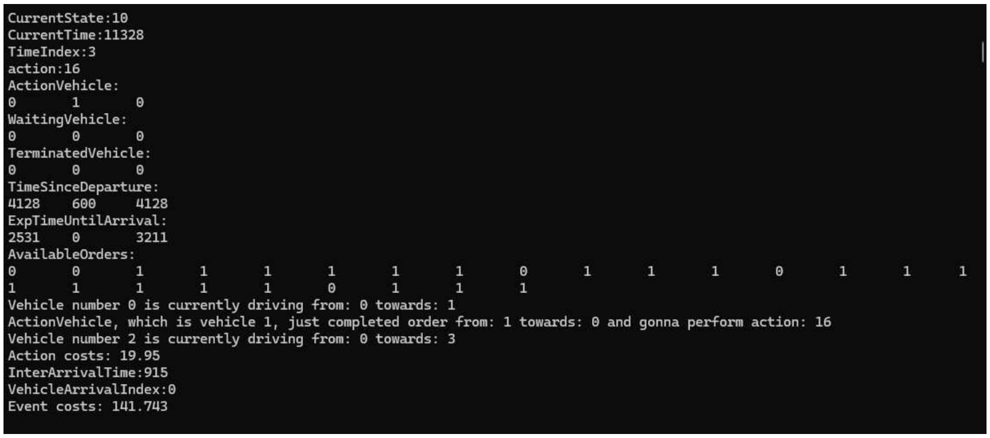
Figure 29: Snippet of a DataFrame comprising 25000 samples from a problem instance recorded on March 14, 2023. Each row corresponds to a distinct sample, while each column represents a state variable, expressed either as an integer value or a vector.

# E Model verification examples

In this appendix, we present two examples in which we iteratively explain the manual verification process. Figures 30 and 31 are descriptive statements of random states. In total, we manually tested ten states and their respective state transitions on logical behavior. Each state description is tested on four elements, described in 5.3: (i) The correctness of the cost incurrence, (ii) the correctness of the exogenous information, (iii) the adherence to opening- and release times and (iv) the adherence to the availability of transport requests.

**Example one**

The first example we will be verifying is the random state desribed in Figure 30:



```
CurrentState:10
CurrentTime:11328
TimeIndex:3
action:16
ActionVehicle:
0       1       0
WaitingVehicle:
0       0       0
TerminatedVehicle:
0       0       0
TimeSinceDeparture:
4128    600     4128
ExpTimeUntilArrival:
2531    0       3211
AvailableOrders:
0       0       1       1       1       1       1       1       0       1       1       1       0       1       1       1
1       1       1       1       1       0       1       1       1
Vehicle number 0 is currently driving from: 0 towards: 1
ActionVehicle, which is vehicle 1, just completed order from: 1 towards: 0 and gonna perform action: 16
Vehicle number 2 is currently driving from: 0 towards: 3
Action costs: 19.95
InterArrivalTime:915
VehicleArrivalIndex:0
Event costs: 141.743
```

Figure 30: Textual description of a random state and its exogenous information. In this state, vehicle 1 just finished executing an order driving from warehouse Hoogtij to the production plant and will be executing order 16. The next arriving vehicle will be vehicle 0 in 915 seconds.

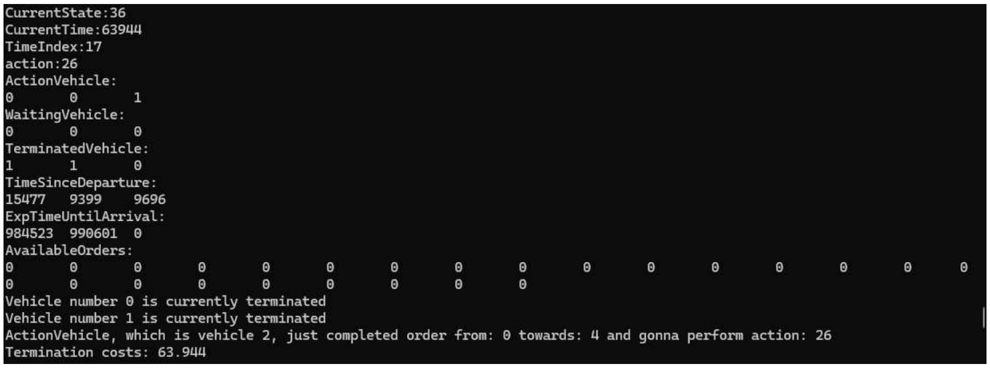
(i) The action is to execute order 16, which stands for driving from the production plant towards the Amsterdam terminal. The distance between these two locations equals 28,5 *km*. Because the current vehicle is already at the pickup location there are no empty kilometers. The cost parameter for driving full equals 0,7. Consequently, the deterministic cost equals $0{,}7 \cdot 28{,}5 = 19{,}95$. Therefore, the simulated costs are in line with the cost function of the paper model.
Similarly, we see in the realization of the post-decision state to the next state that the next arriving vehicle is vehicle 0. Vehicle 0 was performing order 8 with the deadline of 08:00. The realized finishing time of vehicle 0 with order 8 equals 08:24. This means that order 8 arrived 24 minutes past its deadline. Furthermore, the total duration of executing order 8 was 84 minutes. Consequently, the cost composition is as follows: $24 \cdot 3{,}0 + 84 \cdot 0{,}828 = 141{,}5$ . This is roughly similar to the cost the model incurred. Other cost factors do not play a role

(ii) Based on the textual description we see that the expected time until arrival for vehicle 0 is 2511 and for vehicle 2 is 3211. The realized time until the first arrival equals 915. While this is lower than the expected time the value is still within logical bounds suggesting logical behavior. Similarly, it is also more likely that vehicle 0 arrived before vehicle 2.

(iii) All locations are currently open. Therefore order 8 should not have to wait before they finish. This is not the case so the model exhibits logical behavior. Similarly, order 16 is released at time 08:00, since we have passed this time the order can be executed.

(iv) The state vector representing the availability of orders shows that order 16 was available in this random state. Therefore, no invalid actions was made.

**Example two**
The second example we will be verifying is a terminating state in Figure 31:



```
CurrentState:36
CurrentTime:63944
TimeIndex:17
action:26
ActionVehicle:
0       0       1
WaitingVehicle:
0       0       0
TerminatedVehicle:
1       1       0
TimeSinceDeparture:
15477   9399    9696
ExpTimeUntilArrival:
984523  990601  0
AvailableOrders:
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0
Vehicle number 0 is currently terminated
Vehicle number 1 is currently terminated
ActionVehicle, which is vehicle 2, just completed order from: 0 towards: 4 and gonna perform action: 26
Termination costs: 63.944
```

Figure 31: Textual description of a random state and its exogenous information. In this state, vehicle 2 arrived at location 4 and will be terminated because no orders are available. Remark that the time since departure and expected time until arrival for terminated vehicles are arbitrarily high for modeling purposes.

(i) In this final state the total costs incurred are 63,944. All orders are satisfied so there are no penalty costs, since all element values in the AvailableOrders vector equal 0. The makespan equals 22:46, following the multiplication with the cost parameters this equals 17,32. Furthermore, the distance between the production plant and the Amsterdam warehouse equals 26.8 *km*. Therefore the cost of driving back to Hoogtij equals $2{,}1 \cdot 26{,}8 = 56{,}28$. Therefore the compounded costs made in the final state equal $0 + 17.32 + 46.62 = 63.94$. This is similar to the cost the model incurred.

(ii) The exogenous information check holds less relevance in this scenario due to the state being a terminating one.

(iii) Regarding the time since departure for vehicle 2, it indicates that the order was initiated at 20:04, with the current time being 22:46. At this time, both the opening times of all locations and the release times of the orders have elapsed, satisfying this temporal check that orders cannot be assigned before opening- and release times.

(iv) Given that no orders are available, no order can be assigned to the available vehicle. Consequently, the only feasible actions are either the waiting action or the terminating action. The action taken in this instance is the termination action, thereby fulfilling this particular criterion.

# F Private

# G Private

# References

[1] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play*, vol. 362. 2018.

[2] W. Powell, *Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions.* Wiley, 2022.

[3] B. H. Ojeda Rios, E. C. Xavier, F. K. Miyazawa, P. Amorim, E. Curcio, and M. J. Santos, "Recent dynamic vehicle routing problems: A survey," *Computers  Industrial Engineering*, vol. 160, p. 107604, 2021.

[4] J. Montantes, "What you need to know about deep reinforcement learning," Month the Article was Published 2020. Towards Data Science, Medium.

[5] "Artificial intelligence to thrive in logistics according to dhl and ibm." https://www.dhl.com/global-en/home/press/press-archive/2018/artificial-intelligence-to-thrive-in-logistics-according-to-dhl-and-ibm.html. Accessed: 2024-01-12.

[6] N. P. Farazi, T. Ahamed, L. Barua, and B. Zou, "Deep reinforcement learning and transportation research: A comprehensive review," *CoRR*, vol. abs/2010.06187, 2020.

[7] M. Nazari, A. Oroojlooy, M. Takáč, and L. V. Snyder, "Reinforcement learning for solving the vehicle routing problem," *Advances in Neural Information Processing Systems*, vol. 2018-December, pp. 9839–9849, 2 2018.

[8] K. Zhang, F. He, Z. Zhang, X. Lin, and M. Li, "Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 121, 2 2020.

[9] I.-M. Chen, C. Zhao, and C.-Y. Chan, "A deep reinforcement learning-based approach to intelligent powertrain control for automated vehicles," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2620–2625, IEEE, 2019.

[10] T. Oda and C. Joe-Wong, "Movi: A model-free approach to dynamic fleet management," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 2708–2716, IEEE, 2018.

[11] A. Singh, A. Al-Abbasi, and V. Aggarwal, "A reinforcement learning based algorithm for multi-hop ride-sharing: Model-free approach," in *Neural Information Processing Systems (NeurIPS) Workshop*, 2019.

[12] DynaPlex, "Deep reinforcement learning for data-driven logistics," 2023. Retrieved from DynaPlex: `https://dynaplex.github.io/dynaplex/`.

[13] Bolk, "About bolk," 2023. Retrieved from Bolk: `https://bolk.com/en/about-bolk/`.

[14] B. B. Improvement, "About bbi," 2023. Retrieved from Bolk Business Improvements: `https://www.bolkbusinessimprovement.com/over-ons`.

[15] G. Bakker, "About gam bakker," 2023. Retrieved from Gam Bakker: `https://gambakker.com/nl/diensten/transport`.

[16] Cargill, "About cargill," 2023. Retrieved from Cargill: `https://www.cargill.com/about`.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* A Bradford Book, second ed., 2018.

[18] H. Heerkens and A. van Winden, *Solving Managerial Problems Systematically.* Noordhoff Uitgevers, 2017. Translated into English by Jan-Willem Tjooitink.

[19] "Data register of the dutch government — data overheid."

[20] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[21] 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*, vol. 22, no. 3, pp. 2033–2062, 2022.

[22] B. G. Luisa, *Algorithms and Complexity*, vol. 260. Elsevier Science, 2014.

[23] G. Desaulniers, J. Desrosiers, A. Erdmann, M. M. Solomon, and F. Soumis, *9. VRP with Pickup and Delivery*, pp. 225–242.

[24] H. Luoma-Halkola and O. Jolanki, "Aging well in the community: Understanding the complexities of older people's dial-a-ride bus journeys," *Journal of Aging Studies*, vol. 59, p. 100957, 2021.

[25] N. Wilson, J. Sussman, H. Wong, and B. Higonnet, "Scheduling algorithms for a dial-a-ride system," 1971.

[26] H. Min, "The multiple vehicle routing problem with simultaneous delivery and pick-up points," *Transportation Research Part A: General*, vol. 23, pp. 377–386, 9 1989.

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

[28] A. Attanasio, J. F. Cordeau, G. Ghiani, and G. Laporte, "Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem," *Parallel Computing*, vol. 30, pp. 377–387, 3 2004.

[29] T. Flatberg, G. Hasle, O. Kloster, E. J. Nilssen, and A. Riise, *Dynamic And Stochastic Vehicle Routing In Practice*, pp. 41–63. Boston, MA: Springer US, 2007.

[30] A. Beaudry, G. Laporte, and T. Melo, "Dynamic transportation of patients in hospitals," *OR Spectrum*, vol. 32, pp. 77–107, 2010.

[31] T. Zhang, W. A. Chaovalitwongse, and Y. Zhang, "Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries," *Computers  Operations Research*, vol. 39, pp. 2277–2290, 10 2012.

[32] R. P. Hornstra, A. Silva, K. J. Roodbergen, and L. C. Coelho, "The vehicle routing problem with simultaneous pickup and delivery and handling costs," *Computers  Operations Research*, vol. 115, p. 104858, 3 2020.

[33] M. Koulaeian, H. Seidgar, M. Kiani, and H. Fazlollahtabar, "A multi depot simultaneous pickup and delivery problem with balanced allocation of routes to drivers," *International Journal of Industrial Engineering : Theory Applications and Practice*, vol. 22, no. 2, p. 223 – 242, 2015. Cited by: 17.

[34] B. Olgun,  Koç, and F. Altiparmak, "A hyper heuristic for the green vehicle routing problem with simultaneous pickup and delivery," *Computers  Industrial Engineering*, vol. 153, p. 107010, 12 2020.

[35] Y. Rist and M. A. Forbes, "A new formulation for the dial-a-ride problem," *Transportation Science*, vol. 55, pp. 1113–1135, 2021.

[36] C. Ackermann and J. Rieck, "New optimization guidance for dynamic dial-a-ride problems," in *Operations Research Proceedings 2021* (N. Trautmann and M. Gnägi, eds.), (Cham), pp. 283–288, Springer International Publishing, 2022.

[37] S. Dong, "New formulations and solution methods for the dial-a-ride problem," 2022.

[38] X. Liang, G. H. de Almeida Correia, K. An, and B. van Arem, "Automated taxis' dial-a-ride problem with ride-sharing considering congestion-based dynamic travel times," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 260–281, 3 2020.

[39] S. M. Raza, M. Sajid, and J. Singh, "Vehicle routing problem using reinforcement learning: Recent advancements," in *Advanced Machine Intelligence and Signal Processing* (D. Gupta, K. Sambyo, M. Prasad, and S. Agarwal, eds.), (Singapore), pp. 269–280, Springer Nature Singapore, 2022.

[40] R. W. Y. L. Y. Geng, E. Liu, "Deep reinforcement learning based dynamic route planning for minimizing travel time," 2020.

[41] N. D. Kullman, J. E. Mendoza, M. Cousineau, and J. C. Goodson, "Atari-fying the vehicle routing problem with stochastic service requests," *CoRR*, vol. abs/1911.05922, 2019.

[42] J. Zhao, M. Mao, X. Zhao, and J. Zou, "A hybrid of deep reinforcement learning and local search for the vehicle routing problems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7208–7218, 2021.

[43] W. H. R. van Steenbergen, M.R.K. Mes, "Reinforcement learning for humanitarian relief distribution with trucks and uavs under travel time uncertainty," *University of Twente,*, 2021.

[44] B. Balaji, J. Bell-Masterson, E. Bilgin, A. Damianou, P. M. Garcia, A. Jain, R. Luo, A. Maggiar, B. Narayanaswamy, and C. Ye, "Orl: Reinforcement learning benchmarks for online stochastic optimization problems," 2019.

[45] Z. Iklassov, I. Sobirov, R. Solozabal, and M. Takac, "Reinforcement learning for solving stochastic vehicle routing problem," 2023.

[46] J. Li, L. Xin, Z. Cao, A. Lim, W. Song, and J. Zhang, "Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning," *CoRR*, vol. abs/2110.02634, 2021.

[47] A. G. Soroka, A. V. Meshcheryakov, and S. V. Gerasimov, "Deep reinforcement learning for the capacitated pickup and delivery problem with time windows," *Pattern Recognit. Image Anal.*, vol. 33, p. 169–178, jun 2023.

[48] A. K. Kalakanti, S. Verma, T. Paul, and T. Yoshida, "Rl solver pro: Reinforcement learning for solving vehicle routing problem," in *2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, pp. 94–99, 2019.

[49] T. Temizöz, C. Imdahl, R. Dijkman, D. Lamghari-Idrissi, and W. van Jaarsveld, "Deep controlled learning for inventory control," 11 2020.

[50] M. L. Puterman, "Markov decision processes: Discrete stochastic dynamic programming," *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, pp. 1–649, 1 2008.

[51] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 12 2014.

[52] G. C. McDonald, "Ridge regression," *WIREs Computational Statistics*, vol. 1, no. 1, pp. 93–100, 2009.

[53] J. Li, L. Xin, Z. Cao, A. Lim, W. Song, and J. Zhang, "Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2306–2315, 2022.