

A simulation based approach to synchromodal container transport

W.J. de Koning

Graduation thesis
MSc Applied Mathematics



A simulation based approach to synchromodal container transport

by

W.J. de Koning

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday October 1, 2019 at 01:30 PM.

Student number:	4741625	
Project duration:	November 1, 2018 – October 1, 2019	
Thesis committee:	Dr. D. C. Gijswijt	TU Delft (supervisor)
	Prof. dr. ir. K. I. Aardal,	TU Delft
	Prof. dr. F. H. J. Redig	TU Delft
	Dr. F. Phillipson,	TNO
	Ir. I. Chiscop,	TNO

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



ABSTRACT

Logistics service providers (LSPs) offering container transport to the hinterland of the Netherlands face the challenge of efficiently using the capacity of the barge in order to minimize cost, while part of the relevant information is still lacking at the moment decisions have to be made. The existing infrastructure and the transportation activities are studied and modeled as an online optimization problem with simultaneously vehicle routing and container-to-mode assignment. A characteristic of great importance in the problem is the uncertainty element that reflects in the requested appointment times that have to be confirmed by another agent in the network. An online optimization approach is proposed, where the input data come in sequentially and decisions have to be made in between, because new information becomes available only after the decision has been made. At each decision moment, the uncertainty element is converted to an offline optimization problem by disregarding the uncertainty or by simulating various potential future scenarios. Subsequently, the problem is modeled as a multi-commodity network design problem on a time-space graph. Four different solution methods are developed in order to solve the online optimization problem. Three confirmation based methods concern a model in which the uncertainty element is partially disregarded, by assuming that each requested appointment time will be scheduled at a specific time relative to the requested one. Alternatively, a much more complex method is developed in which various future scenarios are simulated for the requested appointment times given their probability vector. The simulation based model seeks robust solutions that are *resistant to change*, i.e., feasible and (sub)optimal for every potential future scenario that has been simulated. Using randomly generated (but realistic) instances, the computational results show that the proposed simulation model surpasses the simpler models both in terms of outcomes, robustness and reliability. However, the practical relevance is somewhat restricted in the sense that the model is built on several assumptions.

Keywords: *intermodal transport, synchronodal transport, online optimization, multi-period time window, simulation algorithm*

PREFACE

The thesis in front of you has been written to fulfill the graduation requirements of the Master of Science in Applied Mathematics at Delft University of Technology. The research has been carried out within the project entitled *Complexity Methods for Predictive Synchronodality* (CoMet-PS). One of the partners in the project is the Netherlands Organization for applied scientific research (TNO), where I got the opportunity to apply the knowledge I gained during my study and giving me the possibility to learn an incredible amount.

I would like to thank my supervisor at the TU Delft, Dion Gijswijt, for his valuable guidance in the thesis process, interest and the biweekly brainstorm sessions. Furthermore, I would like to thank my daily supervisors at TNO, Frank Phillipson and Irina Chiscop, for their ideas, feedback and guidance. And my co-supervisor at TNO, Alex Sangers, for getting me around at TNO in the beginning of the project. Last but not least, I would like to thank the colleagues and other interns within the TNO CSR department for the great working atmosphere, coffee breaks and the interesting conversations.

*Wouter de Koning
Delft, September 2019*

CONTENTS

1	INTRODUCTION	1
1.1	Problem description	2
1.2	Base instance	4
1.3	Research question	7
1.4	Report structure	7
2	ESTABLISHING THE CONCEPTS	9
2.1	Inter- and synchmodal freight transport	9
2.2	Integer linear programming	12
3	LITERATURE REVIEW	15
3.1	Framework for synchmodal problems	15
3.2	Container scheduling	17
3.3	Container scheduling and vehicle routing	20
3.4	Contribution made by this work	22
4	OFFLINE OPTIMIZATION	23
4.1	Creating in- and outland orders	23
4.2	Multi-Commodity Network Design Problem	24
4.2.1	General MCNDP	25
4.2.2	Adding practice-oriented constraints	26
4.2.3	Determining the cost structure	28
4.3	Approaching the lower bound	29
5	ONLINE OPTIMIZATION	31
5.1	Some remarks on online optimization	31
5.1.1	The source node for shipped orders	31
5.1.2	Varying the period between two decision moments	31
5.1.3	Unloading process	32
5.2	The benchmark approach	33
5.3	Confirmation based models	34
5.3.1	RC model	34
5.3.2	EC model	36
5.3.3	AC model	36
5.4	Simulation model	37
5.4.1	Definitions	37
5.4.2	Start of the algorithm	38
5.4.3	Decision space	38
5.4.4	Solving the ILP	41
6	COMPUTATIONAL RESULTS	45
6.1	Offline results	45
6.2	Online results	47
6.2.1	Sensitivity analysis on the number of simulations	50
6.2.2	Computational time	52
6.2.3	LP relaxation	53
6.2.4	Extending the amount of (uncertain) data	53
6.3	Summary	55
7	CONCLUSION	57
7.1	Conclusions	57
7.2	Recommendations	58
A	PSEUDO CODE	61

1

INTRODUCTION

In recent years, a remarkable growth is noticeable [32] in the number of containers that have to be transported from one place to another by different kinds of resources, e.g., trucks, trains and barges. A set of these resources linked together with the purpose of transporting freight (or people) from one place to another, is called a *logistics network*. Logistics networks are present almost everywhere around us. The train you take to work each day, the handling of your baggage at an airport (Figure 1.1), or the packages you order at webshops (Figure 1.2) are all part of such complex networks.



Figure 1.1: Baggage handling



Figure 1.2: Package service

A logistics network is usually run by a logistics service provider (LSP), who faces the problem of delivering the right amount of freight in the right place at the right time. Due to the ever-growing complexity of these networks, an LSP needs efficient tools to support his decisions in order to strive for the optimal network performance at minimal cost [14]. The decision making process could be classified into three levels: strategic, tactical and operational. The levels are described by both Schmidt and Wilhelm [31], and Crainic and Laporte [9]. The *strategic level* includes the design of the physical network, the location and capacity of facilities and resource acquisition. The establishment of such designs requires a lot of money and effort, and these decisions have a long-lasting impact on the network performance. Therefore, the decisions at strategic level have long-term goals. The *tactical level* includes the allocation and use of the established resources, and those decisions are considered as the medium term decisions. The *operational level* is concerned with short term decisions that need to be made by local management. The most important operational decisions relate to scheduling the transport and maintenance services, routing and dispatching of vehicles and allocating freight to transport modes.

In order to make the decision making process at operation level more efficient, reliable and sustainable, the concept of *synchromodal container transport* is extensively covered in Section 2.1. Briefly, synchromodal container transport includes two important characteristics [33]:

1. Customers will only tell the LSP when and where their freight needs to arrive, therefore, entrusting the LSP with the planning.
2. Planners will use data that is real-time, and routes will become subject to change in real-time when beneficial (in terms of time, costs and CO₂ emissions).

In other words, LSPs have the opportunity to switch mode of transport of the containers based on the real-time data. Every day, every hour or every minute, plans are adjusted when disturbances occur, but also, robust plans are made taking potential future disturbances into account. The planning problems that arise due to this added flexibility are synchromodal planning problems. In this thesis the focus is mainly on synchromodal planning at operational level from the perspective of a logistics service provider.

1.1 PROBLEM DESCRIPTION

The challenge faced by a Dutch logistic service provider is the transportation of containers from the eastern part of the Netherlands to the port of Rotterdam and vice versa. Every day, multiple barges depart from the single inland terminal in the east to different deep-sea terminals within the port of Rotterdam. The travel time of such a long-haul trip is around twenty to twenty-four hours. Nowadays, most barges transport containers from the single inland terminal to different deep-sea terminals within the port and return to the inland terminal: a round trip. In such a round trip each barge delivers containers and, simultaneously, collects containers from the same, and possibly other, terminals. The collected containers are transported back to the inland terminal where the barge started its journey. Briefly, we might consider the transportation of freight from (i) the single inland terminal to multiple locations within a far away region, denoted by *outland orders*, and from (ii) locations within the region back to the single inland terminal, denoted by *inland orders*. Notice that no freight needs to be shipped between any two deep-sea terminals within the port of Rotterdam.

Besides the use of a limited number of barges, we assume that there is an unlimited number of trucks that can be used for urgent freight which cannot be transported by barge. At each decision moment, a barge planner has to decide which containers to allocate to which barges and/or trucks. This decision has to be made in such a way that the network performance of the Dutch LSP is optimized over time. For example, trucking a container to (or from) a given terminal right now, even if it is possible to transport it by barge, might reduce the number of trucks used in total. Indeed, trucking one container right now might prevent that multiple containers have to be trucked at a later stage.

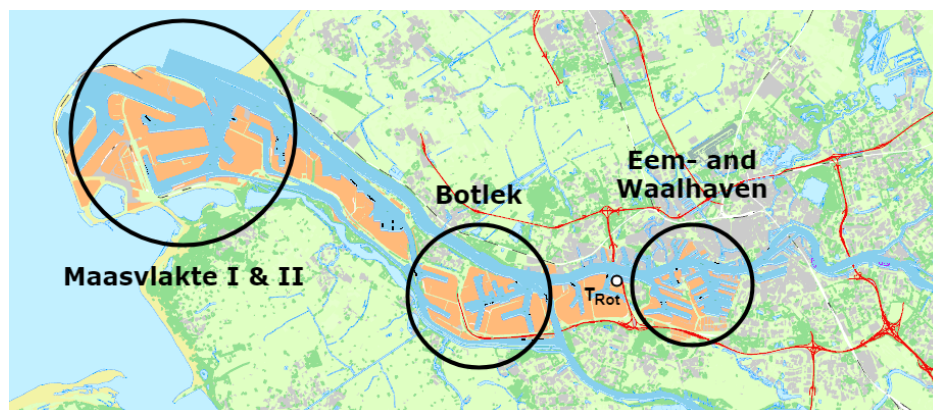


Figure 1.3: Location of barge terminals in the port of Rotterdam, including the container terminal (T_{Rot}) operated by the Dutch LSP

The port of Rotterdam has about 33 container terminals, including empty depots. These terminals are spread over a rather large port area. Spatial clusters of terminals are found in the area of Eem- and Waalhaven, Botlek and Maasvlakte I and II, as illustrated in Figure 1.3. The distance between Eem- and Waalhaven and Maasvlakte

is about 40 kilometers on water, which is about two to three hours by barge [22]. In addition to the single inland terminal in the Eastern part of the country, a container terminal within the port of Rotterdam belongs to the Dutch LSP, denoted by T_{Rot} . At this container terminal, freight could be temporarily stored or switch vehicles.

The region (the port of Rotterdam) is far away from the origin (the single inland terminal), but locations within the region are close to each other. Thus the long-haul trip from the origin to the region (and back) is the most challenging one. Furthermore, the terminals within the region are controlled by other agents, who confirm the requested calls from the LSP with a delay of approximately half a day. Most of the time, these confirmed calls deviate from the requested ones, which means that the barge planners have to deal with some uncertainty in their available data.

To further clarify the uncertainty element, we will have a look into the process of the orders. The LSP receives orders from clients on a daily basis. These orders consist of one or more empty (or full) containers that have to be picked up at a certain terminal in the port of Rotterdam, and then have to be transported to the client's warehouse located near the single inland terminal, where the freight is loaded (or unloaded) at an agreed time. Thereafter, the full (or empty) containers have to be transported back to a certain terminal in the port of Rotterdam. The pick-up and delivery terminals might be different. When the LSP receives an incoming order, the pick-up and delivery location are prespecified. The pick-up and delivery time, however, are not known yet. The barge planner has to make a call towards the particular terminal(s) in order to request for an appointment, consisting of a specific date and time. Moreover, the number of containers that have to be handled has to be passed on. After some delay, the specific terminal responds to the request in the form of a confirmed appointment time. As mentioned before, the confirmed appointment time often deviates from the requested one, implying that decisions have to be made based on stochastic information. Even if an order is known a few days in advance, the barge planners still make a call only 36 hours in advance relative to the requested appointment time. Since the other agents in the region confirm the calls only 24 hours in advance, regardless of the circumstances, the barge planners have experienced that it does not benefit to call at an earlier stage.

Using the agreed time at the client's warehouse for (un)loading and the requested/confirmed pick-up and delivery times at the corresponding terminals, the orders K of the clients could be split into *inland orders* K^{in} and *outland orders* K^{out} . Each inland order $k \in K^{in}$ consists of a

- pick-up location
- requested/confirmed pick-up time
- delivery location
- delivery time (i.e., due time¹)
- size of the order

Notice that we could omit the delivery location, since every inland order has to be transported to the single inland terminal in the eastern part of the Netherlands. The trip from the inland terminal to the client, and vice versa, is always done by truck, so we may disregard that part of the trip. Further, each outland order $k \in K^{out}$ consists of a

- pick-up location
- pick-up time (i.e., release time)
- delivery location
- requested/confirmed delivery time
- size of the order

¹ The order has to be delivered before or at this time, so it is less strict than the confirmed delivery time corresponding to an outland order.

By the same reasoning, we could omit the pick-up location, since every outland order has to be transported from the single inland terminal to a terminal in the port of Rotterdam.

As mentioned before, the travel time of the long-haul trip between the single inland terminal and the port of Rotterdam is around twenty to twenty-four hours. Since the other agents confirm the requested calls only 24 hours in advance, a planning may become subject to change when beneficial. For example, at some point in time, the LSP assigns outland order $k \in K^{out}$ to barge \mathcal{B} , located at the origin, while the order is not confirmed yet. At that time, based on the requested pick-up and delivery times, the LSP benefits the most when the barge first picks up inland order $k' \in K^{in}$, then delivers the outland order and finally picks up inland order $k'' \in K^{in}$. However, when time passes by, the requested times are confirmed and might deviate. Based on the real-time data it might be more beneficial to unload the outland order at the container terminal T_{Rot} and deliver the order at the delivery appointment by truck, such that the barge is able to visit some other confirmed appointments. Since the LSP has the ability to change the plan when beneficial, using multiple modes of transport, the problem described coincides with a synchromodal planning problem.

Our goal is to make use of all the practical information that is available in order to formulate an optimization problem that suits the preferences of the barge planners.

1.2 BASE INSTANCE

To be able to deal with the big, real-life problem and to explore solution methods, we start our work with a small base instance. We consider the following simplified instance obtained by reducing the size of the real-life problem and introducing some assumptions. The network under consideration, as shown in Figure 1.4, consists of:

- 1 single inland terminal operated by the barge planner, denoted by T_{Origin} ,
- 1 container terminal in the Port of Rotterdam operated by the barge planner, denoted by T_{Rot} ,
- 3 deep-sea terminals (region D), denoted by T_1, T_2 and T_3 .

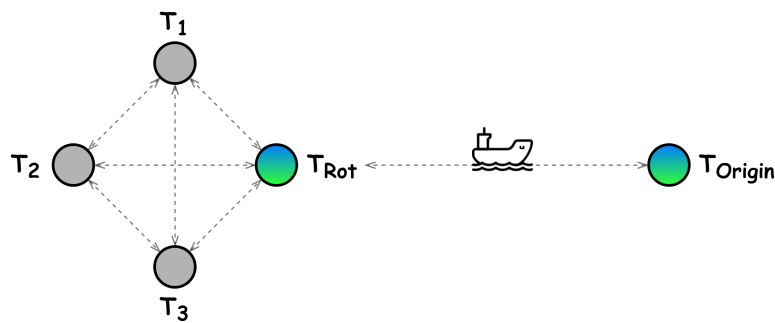


Figure 1.4: The network under consideration in the base instance

Observe that there is a main difference between the terminals operated by the barge planner and the deep-sea terminals. At the terminals operated by the barge planner, the barge planners have *complete freedom*. That is to say, barges can always moor at these terminals and decisions can be made (or changed) last minute. At the region D , however, a barge planner has to make a request for an appointment at a specific deep-sea terminal. After some delay, half a day in our base instance, a confirmed appointment time is returned, which may deviate from the requested one. Only in case of a confirmed appointment a barge may visit a deep-sea terminal.

To be able to send the containers through the network under consideration, the barge planners have access to the following resources:

- 2 barges $\mathcal{B}_1, \mathcal{B}_2$ with capacity of 10 containers,
- 1 barge \mathcal{B}_3 with capacity of 20 containers,
- unlimited number of trucks with capacity of 1 container; there is a fixed cost per time step (or kilometer) traveled by truck.

A characteristic of great importance in the problem is the uncertainty element that reflects in both the requested appointment times that have to be confirmed and the orders not announced yet. In order to address the presence of uncertainty, a *multi-period time window (MPTW)* approach is introduced. The approach is a reactive method that solves iteratively the planning problem by moving forward on the MPTW after each decision made, assuming that the status of the system is updated as soon as the stochastic and unknown elements (i.e., orders) become deterministic and known, respectively.

In terminology we distinguish controlled time windows and single-period time windows, that are both subintervals of the MPTW. A *controlled time window (CTW)* is defined as the period between two consecutive decision moments, i.e., the part of the planning that is actually performed. The time window containing all the relevant information known at an arbitrary decision moment is defined as the *single-period time window (SPTW)*.

In Figure 1.5 the multi-period time window approach is visualized. At the start of each decision moment t , a decision has to be made for the upcoming CTW, based on the information available in the concerned SPTW, i.e., based on the state of the system at the start of decision moment t . Observe that the information available could be both deterministic and stochastic.

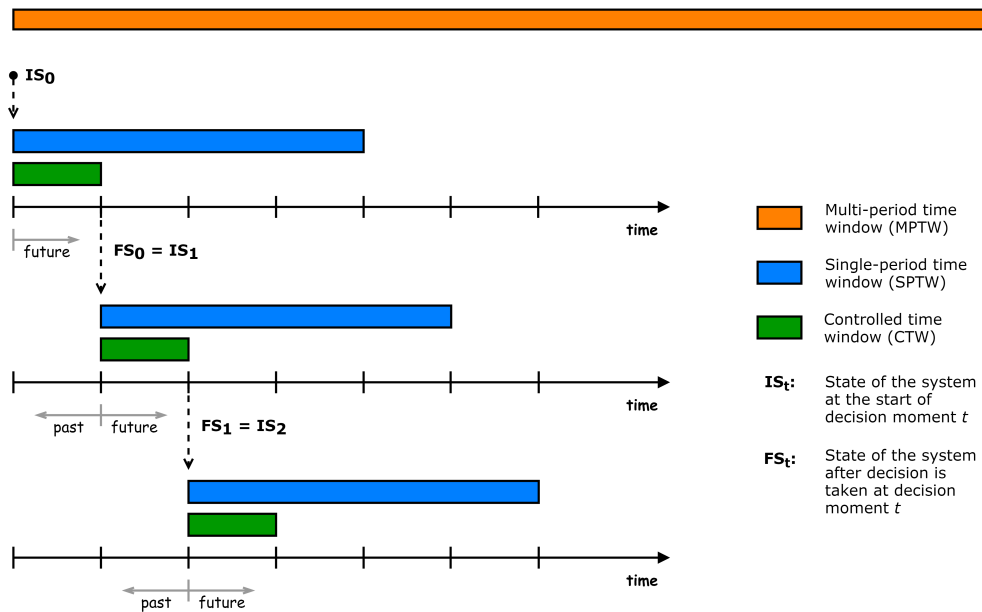


Figure 1.5: Solving the planning problem using a multi-period time window approach

In addition, to deal with the uncertainty, we need to simplify our mathematical model by making several assumptions.

- The MPTW is divided into a finite number of time steps (i.e., discrete approach), where each time step corresponds to three hours in real-life. The multi-period time window starts at time step 0 and covers nine days, until time step 72.

- Inland orders become known when the barge planners request an appointment, that is 12 time steps in advance relative to the requested appointment time. Furthermore, the requested appointment is confirmed 4 time steps later, i.e., 8 time steps in advance.
- Outland orders become known 12 time steps in advance relative to the release time². The corresponding requested appointment is confirmed in the same way, i.e., 8 time steps in advance relative to the requested appointment time.
- The confirmed appointment time could be scheduled at the requested appointment time, earlier (at most one time step) or later (one until five time steps), with probability distribution $\mathbf{p} = (p_{-1}, p_0, p_{+1}, p_{+2}, p_{+3}, p_{+4}, p_{+5})$.
- At each decision moment, a single-period time window of 32 time steps is considered, which can be divided into three parts relative to the requested appointment times, as shown in Figure 1.6.

PART 1: Orders having a requested appointment time in the interval $[0, 8]$ that is confirmed already.

PART 2: Orders having a requested appointment time in the interval $(8, 12]$ that is not confirmed yet.

PART 3: Outland orders³ having a pick-up time in the interval $[0, 12]$ and a requested appointment time strictly greater than 12.

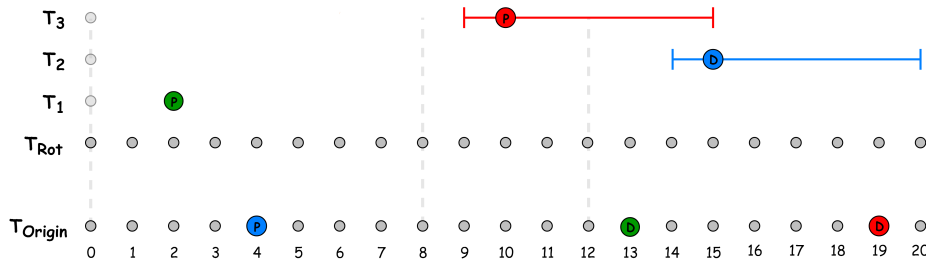


Figure 1.6: Example of the three parts in a SPTW, where the green, red and blue dots (and intervals) correspond to the first, second and third part, respectively.

- The three deep-sea terminals in the region D may only be visited in case of a confirmed appointment.
- The number of barges that may visit an appointment is restricted to 1. Observe that the restriction does only apply to barges, visiting an appointment with one barge and multiple trucks is allowed.
- The travel times of the barges are known and fixed. In particular, we assume that the travel time between any two terminals in $D \cup T_{Rot}$ equals 1 time step (i.e., three hours), and the travel time between the origin T_{Origin} and the container terminal T_{Rot} equals 7 time steps (i.e., twenty-one hours).
- The travel times of the trucks are known and fixed. In particular, we assume that the travel time between any two terminals in $D \cup T_{Rot}$ equals 1 time step (i.e., three hours), and the travel time between the origin T_{Origin} and the container terminal T_{Rot} equals 2 time steps (i.e., six hours).

² In case an outland order would become known only 12 time steps in advance relative to the requested appointment time, the release time would become known only a few hours in advance. There would be an extra (unrealistic) stochastic element in our model that is not preferred.

³ By assumption, only outland orders could have a requested appointment time in the interval $(12, 32]$. Therefore, inland order are disregarded in the third part.

- At any point in time, an unlimited number of trucks is available at every terminal, which can (i) transport containers from the pick-up location directly to their destination or to the container terminal or (ii) transport containers from the container terminal to their destination. We charge costs for the use of trucks.
- Containers could be temporarily stored or switch vehicles at the container terminal T_{Rot} . However, handling time is taken into account for the unloading and loading process, i.e., one time step for each processing.
- Handling time is taken into account for the unloading process at the origin T_{Origin} , i.e., one time step.

1.3 RESEARCH QUESTION

In the base instance, a MPTW of nine days (i.e., 72 time steps) is considered. At each decision moment t , a decision has to be made for the upcoming CTW, based on the state of the system at the start of decision moment t . Such a decision is established by *solving the SPTW*. When time passes by, new information becomes available and again a SPTW needs to be solved. To be precise, we need to solve multiple SPTWs of length 32 time steps to gain suitable solutions (or decisions) for each CTW. At the end of the multi-period time window, these decisions can be merged and combined in order to gain a final solution for the entire MPTW. Therefore, we formulate the following research question:

“What kind of solution methods can be used for the single-period time windows, including uncertainty, to find an appropriate schedule and container assignment for the entire multi-period time window (within an acceptable amount of time)?”

In order to answer the research question, the following sub-questions have to be answered.

1. How can the base instance described in Section 1.2 be modeled in order to meet all the assumptions made?
2. What solution methods can be used to obtain both a schedule and a container assignment for every transportation mode in the network?
3. What can be said about the quality and practical relevance of the results obtained by the solution methods?
4. What can be said about results obtained by the solution methods if the assumption on the announcement time of orders is relaxed, i.e., when orders are announced at an earlier stage?
5. Does the chosen approach successfully incorporate elements from synchronomodality?

1.4 REPORT STRUCTURE

This thesis has been organized in the following way. Chapter 1 describes the general problem under study, and presents a simplified base instance which will be studied to explore solution methods. Chapter 2 gives a more detailed description of the freight transportation concepts and sketches some methods for solving (integer) linear programming problems. In Chapter 3 we take a look at the existing literature on intermodal and synchronomodal problems. Chapter 4 discusses the field of offline optimization, where all relevant information is known when solving the problem.

Only *ex post*, when all information has become available, the truly optimal solution can be computed offline, which can be used as a lower bound for the online optimization. In Chapter 5 solution methods for the base instance will be explored by entering the field of online optimization. In Chapter 6 results are presented, interpreted and discussed. Finally, Chapter 7 is dedicated to the conclusions and some recommendations for follow-up study.

2

ESTABLISHING THE CONCEPTS

In this chapter a more in-depth study is presented on different concepts relating to freight transport. In particular, the most widely used methods to transport freight are intermodal and synchromodal transport. Furthermore, an introduction to integer linear programming is given, because it is the main mathematical tool that will be used to solve the planning problem.

2.1 INTER- AND SYNCHROMODAL FREIGHT TRANSPORT

Over the years at least five concepts relating to freight transportation have been introduced: multimodal, intermodal, combined, co-modal and synchromodal transport. Recently, the *Physical Internet* (PI, π) and *digital twin* have generated a lot of attention among both practitioners and academics. The paper by Reis [27] pinpoints the main properties of the first five concepts. The original concept is multimodal transport, and the most recent one is synchromodal transport. Every new concept utilizes certain elements from the previous concepts and introduces new ones. In Figure 2.1 the sequential relations between these five freight transport related concepts are shown.

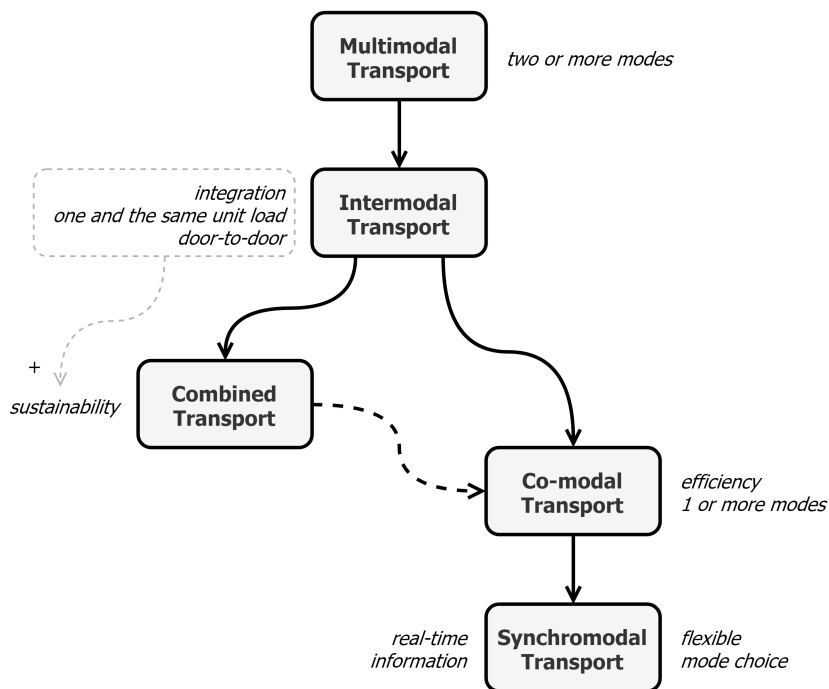


Figure 2.1: Sequential relations between freight transport related concepts, from [27]

The most widely used method to transport freight from one place to another is *intermodal freight transportation*. Intermodal freight transportation involves the transportation of freight in containers of standardized dimensions, using multiple modes of transport (e.g., truck, barge or train), without any handling of the freight itself when changing modes. The fundamental idea of intermodal freight transportation is to

consolidate freight for efficient long-haul transportation (i.e., the transportation of containers between two different ports), while taking advantage of the efficiency of local pick-up and delivery operations by truck [1]. For example, the freight may be picked up by truck at the origin, then be placed on a barge or train, and then travel the last part of the journey by truck again.

Synchromodal freight transportation is positioned as the next step after intermodal transportation. The main difference is the ability to respond to uncertain disturbances (e.g., congestion, accidents, low water levels or maintenance) or other possible stochastic elements within the process, that could lead to delays and money losses. Therefore, a planning is made using real-time information. In other words, the plan will become subject to change when disturbances occur, but also, robust plans are made taking potential future disturbances into account. It must be possible to re-evaluate plans at any moment [16].

The paper of Montreuil [25] defines the *Physical Internet* as an open global logistics system founded on physical, digital, and operational interconnectivity, through encapsulation, interfaces and protocols. The paper starts with the assertion that the way physical objects are currently transported, handled, stored, realized, supplied and used throughout the world is economically, environmentally and socially inefficient and unsustainable. Evidence supporting this assertion is exposed through a set of key unsustainability symptoms. The vision that is presented in the paper is to evolve towards a Physical Internet as a solution to the global logistics sustainability grand challenge. In the Physical Internet freight is moved in a similar way as data is transferred in the Digital Internet: smart, seamless and making use of the network of others. The concept of *synchromodality* is one of the road-maps of the Physical Internet [10].

The paper of Saddik [30] defines a *digital twin* as a digital replica of a living or non-living physical entity. By bridging the physical and the virtual world, data is transmitted seamlessly allowing the virtual entity to exist simultaneously with the physical entity. Digital twins facilitate the means to monitor, understand, and optimize the functions of all physical entities, and provide humans with continuous feedback to improve quality of life and well-being. For example, digital twin technology can help by the decision making process because it has the opportunity to simulate the application before going live. The lessons learned and opportunities uncovered through a digital twin can then be applied to the physical environment. Moreover, digital twins can revolutionize healthcare operations as well as patient care. A digital twin of a patient or organs allows surgeons and health professionals to practice procedures in a simulated environment rather than on a real patient. Sensors with the size of bandages can monitor patients, thus producing digital models, which is overseen by artificial intelligence, thus improving care [24].

Due to the improvements in data technology (e.g. digital twin), the interest in synchromodality has increased. However, synchromodality faces several challenges that keep it from being adopted in practice. The paper by Pfoser et al. [26] identifies seven critical success factors of synchromodality which ensure the effective implementation of it.

- Network, Collaboration and Trust. One of the most important aspects in a synchromodal network is the collaboration and trust between agents. However, many agents are unwilling to cooperate with competitors. A rethinking process is required to achieve a network in which the agents are aware of the advantages of cooperation instead of competition.
- Sophisticated Planning. Logistics networks are complex. To be able to deal with the preferences of the customers, unexpected disturbances and available

resources of locations and transportation modes, forecasts and simulations are needed in order to optimize the network performance.

- ICT and ITS Technologies. Sharing and mutually exchanging data is the key to create a synchromodal network. However, issues dealing with data security and data protection have to be addressed.
- Physical Infrastructure. The locations and capacity of facilities and available resources influence the performance of the synchromodal network. Therefore, the design of the network (e.g., strategic level decisions) forms the basis of an effective implementation of synchromodality.
- Legal and Political Framework. What agents will be held responsible in case of delay, loss or damage? When the transportation mode is switched unplanned, such liability issues are not always clear.
- Awareness and Mental Shift. The willingness of the customers, to leave the decision concerning which transportation modes to use up to the LSP, is another important success factor. Therefore, awareness among customers has to be raised on the advantages of synchromodal transport.
- Pricing, Cost and Service. The pricing of synchromodal services is quite complex. Since the transportation mode(s) and the exact route is not known in advance, it is difficult to set a price. But customers require assurance on the price in advance, so new difficulties arise.

The results of Pfoser et al. suggest that there is quite a uniform agreement upon importance and feasibility of various success factors with cooperation being the most crucial success factor. Although the process to achieve a network in which there is collaboration and trust between agents is outside the scope of this thesis, it is not self-evident that such a network will be achieved in the future. Therefore it is necessary to understand how much information is actually available and shared, and what kind of optimization objective is aimed for.



Figure 2.2: Framework of the four different systems, from [18]

As described in [18], both the information and optimization objective could take a *local* or a *global view*. In case of a local view, only own information is known and the objective is only individually optimized. In case of a global view, information over all agents is available and the goal is to achieve an optimal outcome for the entire network. We may distinguish four different systems of a synchromodal network in a framework, which are shown in Figure 2.2.

Our focus will be on synchronodal planning at operational level from the perspective of a logistics service provider (i.e., sophisticated planning), who is interested in reducing its own overall costs and has some (stochastic) knowledge about the decision making process of other agents in the network. Using Figure 2.2, the system could be seen as a selfish one in which the information is mainly locally available. In addition, the LSP utilizes some services of the other agents, which adds some uncertainty into the model.

2.2 INTEGER LINEAR PROGRAMMING

This section is meant to give the reader some intuition about (integer) linear programming and the branch-and-bound method. As mentioned before, it is the main mathematical tool that will be used to solve the planning problem. For formal proofs, theorems and definitions, we refer the reader to one of the many books about (integer) linear programming that have been written, such as [5].

A *linear programming problem* may be defined as the problem of maximizing or minimizing a linear function, called the *objective function*, in real variables subject to linear constraints. The constraints may be equalities or inequalities. The concepts are illustrated by means of the following example.

Example 2.2.1. Maximize the linear function $f(x_1, x_2) = 5 \cdot x_1 + 6 \cdot x_2$ subject to the (inequality) constraints

$$\begin{aligned} x_1 + x_2 &\leq 5, \\ 4 \cdot x_1 + 7 \cdot x_2 &\leq 28, \\ x_1, x_2 &\geq 0. \end{aligned} \tag{2.1}$$

Every point $(x_1, x_2) \in \mathbb{R}^2$ that satisfies the four constraints is called *feasible*. In addition, the set of all the feasible points is called the *feasible region*, which is shown in Figure 2.3 and denoted by $S \subset \mathbb{R}^2$.

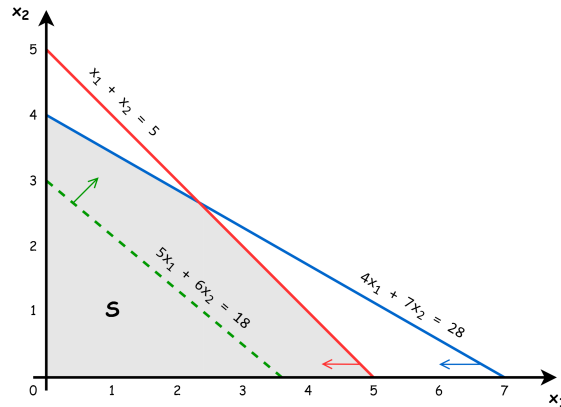


Figure 2.3: The feasible region S

Observe that for any pair of points in the feasible region, the line segment between those two points completely lies within the region, implying that the set of feasible points is *convex*. In general, the set of feasible points for an arbitrary linear program is a convex set (because each feasible region is the intersection of a finite number of *halfspaces* and *hyperplanes*). The objective function f maps every point $x \in S$ to a value. A feasible solution $x^* \in S$ is called *optimal* if $f(x^*) \geq f(x)$ for all $x \in S$ (assuming the objective function is maximized). The goal of linear programming is to establish if an optimal solution exists and to find one (or all of them). If a linear programming problem has a finite optimal objective function value, then an

optimal feasible solution is found on (at least) one of the vertices of the convex set¹. Since there are only two variables, we can solve this problem by graphing the set of feasible points S in the plane and then finding which point(s) of this set maximizes the value of the objective function f . The function $5 \cdot x_1 + 6 \cdot x_2$ is constant on lines with slope $-6/5$, i.e., the line $5 \cdot x_1 + 6 \cdot x_2 = 18$ as plotted in Figure 2.3. As we move this line further from the origin up and to the right, the value of the objective function increases. Therefore, we seek the line of slope $-6/5$ that is farthest from the origin and still touches the feasible region S . This occurs at the intersection of the lines $x_1 + x_2 = 5$ and $4 \cdot x_1 + 7 \cdot x_2 = 28$, which is the point $(7/3, 8/3)$. The value of the objective function is $27 \frac{2}{3}$.

In general, a commonly used method for linear programming problems is the *simplex method*, which is an algorithmic method that seeks a vertex corresponding to an optimal solution [11].

However, if we add the condition that x_1 and x_2 have to be integers, the problem changes into an *integer linear programming (ILP)* problem and the solution methods used for the linear programming problems are no longer applicable, because the set of feasible points is no longer a convex set, as it can be seen in Figure 2.4a.

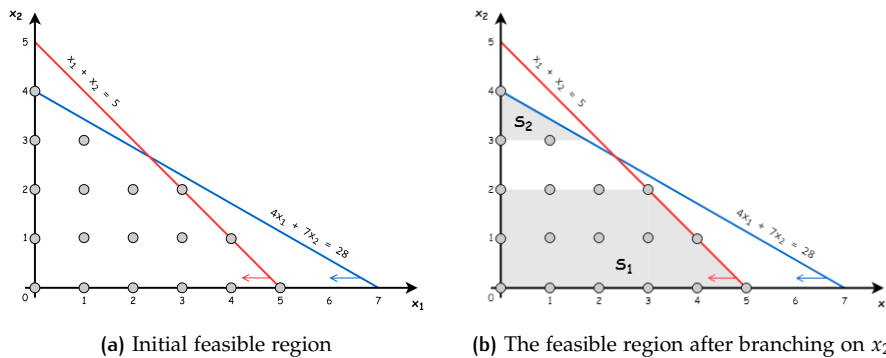


Figure 2.4: Feasible regions before and after applying the B&B method

Just rounding down the optimal solution found in the linear programming problem is not the way to solve the problem, because $(\lfloor 7/3 \rfloor, \lfloor 8/3 \rfloor) = (2, 2)$ is not the optimal solution for the ILP. The *branch-and-bound (B&B)* method can help us to find the best integer solution for this problem. It is a solution approach that can be applied to a number of different types of problems. The B&B approach is based on the principle that the total set of feasible solutions can be partitioned into smaller subsets of solutions.

The B&B method starts by relaxing the condition that x_1 and x_2 are integers, changing the problem into a linear programming problem. As determined above, the solution for the problem is $(7/3, 8/3)$ which gives the objective value $27 \frac{2}{3}$. However, the solution is non-integer, hence infeasible. Therefore we need to branch on one of the variables x_1, x_2 that is non-integer. In this case, both variables are non-integers, so we may randomly pick one, say x_2 . The current value $x_2 = 8/3$ is infeasible, so the variable has to be less than or equal to 2 or the variable has to be greater than or equal to 3, i.e., $x_2 \leq 2$ or $x_2 \geq 3$. In this way, the problem is divided into two subproblems. Both subproblems include the original constraints and, in addition, the new constraints $x_2 \leq 2$ and $x_2 \geq 3$ are added to subproblems 1 and 2, respectively. As can be seen in Figure 2.4b, the feasible regions S_1 and S_2 are disjoint subsets of the original feasible region S .

Using the simplex method (or just the graphical method) the optimal solution for subproblem 1 turns out to be $(3, 2)$ which gives the objective value 27, as illustrated in Figure 2.5.

¹ Assuming the feasible region S is *pointed*, i.e., S has a vertex. This is always the case if the linear program is in *standard form*.

The outcome of the linear programming problem can be considered as an lower bound for the ILP problem. Moreover, observe that the objective function maps every integer point (x_1, x_2) to an integer value, implying that the outcome of the integer linear programming problem can be at most 27.

Because no better result can be achieved in terms of the objective value, there is no reason to proceed. However, for the sake of understanding the entire process of the B&B method, the process is continued along the other branch $x_2 \geq 3$ until integer valued points (x_1, x_2) are found. Although the point $(3, 2)$ is an optimal solution, for now it is called a lower bound to the problem, because it is assumed that there might be another solution which results in a better objective value. Again, using the simplex method (or the graphical method) the optimal solution for subproblem 2 turns out to be $(7/4, 3)$ which gives the objective value $26 \frac{3}{4}$. Since the point is not feasible yet, because x_1 is non-integer, the process is continued, as shown in Figure 2.6.

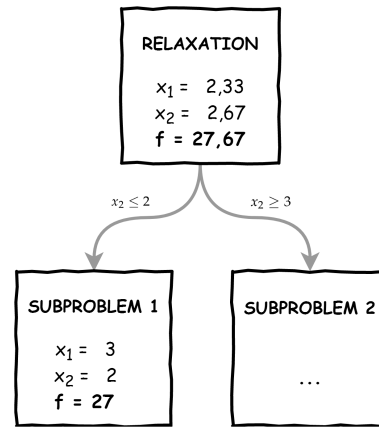


Figure 2.5: B&B method

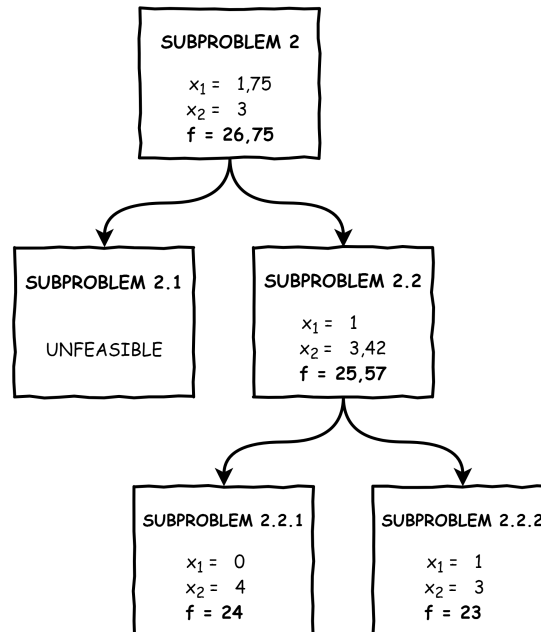


Figure 2.6: Continuation of the B&B method

At the end of the branch, the integer valued points $(0, 4)$ and $(1, 3)$ are found, giving objective values 24 and 23, respectively. These solutions are not better than the one previously found. We may conclude that $(3, 2)$ is the optimal solution of the ILP. Although we figured that out in the very beginning, you do not always get that immediate solution in your first attempt. By constructing the entire tree, we enumerate all feasible solutions to the original problem, implying that we are guaranteed to find the optimal solution.

In general, stop investigating a subproblem if **(i)** the optimal solution is integer, **(ii)** the objective value of the optimal solution is worse than the best integer solution found so far or if **(iii)** the subproblem is infeasible.

3

LITERATURE REVIEW

Before we start exploring solution methods for the problem described in Chapter 2, we take a look at the existing literature on intermodal and synchronodal problems. We distinguish four different problems, as shown in Table 3.1. In addition, it is also important to properly define the planning problem under consideration. Therefore, we start by introducing a framework in which the model choices in our synchronodal planning problem can be classified, based on literature.

	Deterministic	Stochastic
Container scheduling	I	II
Container scheduling & vehicle routing	III	IV

Table 3.1: Problems in synchronodal transport

3.1 FRAMEWORK FOR SYNCHROMODAL PROBLEMS

When solving an optimization problem, it is important to first properly define it. In [18] a framework for synchronodal transportation problems on a tactical or operational level is presented. The framework aims to capture the essential model-making decisions done in the model built, to represent the problem. Within the framework *resource* and *demand elements* are considered. The resource elements refer to available modes of transportation, which are mostly barges, trucks and trains. The demand elements are elements related to demand, which are mostly freight containers. The elements within the network can have different behaviour, we distinguish five.

- *Controllable*: Since we model a decision problem, at least one element of the system must be in control and must take decisions. For example, the container-to-mode assignment.
- *Fixed*: A fixed element does not change within the scope of the problem.
- *Dynamic*: A dynamic element might change over time or due to change in the state of the system (e.g., the amount of containers might change the travel time of a barge), but this change is known and computable beforehand.
- *Stochastic*: A stochastic element is not necessarily known beforehand. For example, it is not known when an incoming order arrives, but the arrivals occur according to a Poisson process.
- *Irrelevant*: It might occur that for certain problems not all elements are taken into consideration to model the system. Then these elements are irrelevant.

We will accurately follow the list of elements presented in [18] in order to describe all the elements occurring in the decision making process of the LSP. However, the framework does not cover all the elements, so small clarifications are mentioned where necessary.

Resource elements

- *Resource Type*: In the problem under study, a finite number of barges (of different known capacities) and an infinite number of trucks is available.
- *Resource Features*:
 - *Resource Origin (RO)*: At the beginning of the MPTW the initial locations of the barges are fixed. However, thereafter the resource origin is controllable, because it is a result of the decision made at the previous CTW.
 - *Resource Destination (RD)*: The barges do not have any restrictions, so the resource destination is controllable.
 - *Resource Capacity (RC)*: Each resource has a fixed capacity.
 - *Resource Departure Time (RDT)*: Both the direction and the departure time of a resource is part of the decision the LSP has to make. Therefore, it is a controllable element.
 - *Resource Travel Time (RTT)*: For both barges and trucks, the travel time between any two locations is fixed.
 - *Resource Price (RP)*: Only costs for the use of trucks will be charged, which is a fixed price.
- *Terminal Handling time (TH)*: At the two terminals that are operated by the barge planner handling time is taken into account, both for unloading and loading containers. The handling time is fixed, and independent of the number of containers that have to be (un)loaded.

Demand elements

- *Demand Type*: In the problem under study, only one type of container of standardized dimensions can be transported. Therefore, it is a fixed element.
- *Demand-to-Resource allocation (D2R)*: The assignment of containers to the barges (and trucks) is part of the decision the LSP has to make. Therefore, it is a controllable element.
- *Demand Features*:
 - *Demand Origin (DO)*: When the LSP receives an incoming order, the pick-up location (i.e., origin) is immediately known and fixed.
 - *Demand Destination (DD)*: Similarly, the destination is a fixed element.
 - *Demand Volume (DV)*: The number of containers (i.e., volume) is passed on as well, so it is a fixed element.
 - *Demand Release Date (DRD)*: We distinguish two types of orders: in- and outland orders. Inland orders have a stochastic pick-up time, and outland orders have a fixed pick-up/release time.
 - *Demand Due Date (DDD)*: Again, distinguishing the two types of orders. Inland orders have a fixed delivery/due time, and outland orders have a stochastic delivery time.
 - *Demand Penalty (DP)*: Refers to costs that are incurred when the delivery time at the destination for a container is not met. For each container, a fixed penalty is taken into account.

Using the compressed framework notation presented in [18], most of the elements occurring in the problem under consideration can be summarized as follows¹.

$$\bar{R}, [RD], [RDT] \mid \bar{D}, [D2R], \widehat{DRD}, \widehat{DDD} \mid \text{selfish}(1+) \mid \text{isolated}$$

¹ For further details, we refer the reader to Juncker et al. [18].

3.2 CONTAINER SCHEDULING

In Problem I and II the transportation modes have fixed timetables. In that case, the only goal is to solve a container-to-mode assignment such that all containers reach their destination on time against minimal total cost.

In Problem I every element in the model is deterministic, i.e., every container has got a prespecified release and due time that is known in advance and no disturbances are assumed to occur. The problem can be solved to optimality in a very short time by solving the *multi-commodity minimum cost flow problem (MCMCFP)* on an appropriate time-space graph [16, 19], as explained in detail in the upcoming subsection.

In Problem II, the goal is the same but almost any element could be stochastic, e.g., travel times, release times or the requested calls that have to be confirmed. In the work of Kooiman [23] the *time-stamp stochastic assignment problem (TSAP)* has been studied. The study concerns the collection and delivery of containers between a container terminal and inland ports by barges, where the barges make fixed round trips, starting at the container terminal. At each terminal, the barges must decide how many containers to load here, given that they may have to reserve space for the next terminals, but not knowing exactly how many containers will be released at those terminals by the time they arrive. To deal with the stochastic release times of the containers a simulation algorithm is developed to decide whether a container is assigned to a barge or not.

The work of Rivera & Mes [29] also addresses future assignments. They investigate the container transportation from the eastern part of the Netherlands to the port of Rotterdam, and vice versa, in fixed long-haul round trips starting from the single inland terminal. While delivering containers, the same barge picks up containers from the same, and other terminals, and transports them back to the inland terminal where it started. They formulate the container-to-mode assignment problem as a *Markov decision process (MDP)* and approximate the solution by means of approximate dynamic programming (ADP).

Multi-Commodity Minimum Cost Flow Problem

In general, the *multi-commodity flow problem (MCFP)* deals with the assignment of commodity flows from source to destination in a network. MCFPs are highly relevant in several fields including transportation and telecommunications. The *multi-commodity minimum cost flow problem (MCMCFP)* is a type of problem that is related to capacitated networks. A capacitated network consists of various nodes and arcs, which can be seen as locations, and the waterways and roads connecting them. Each of these arcs has associated costs and capacities. As mentioned before, the schedule of the vehicles is known beforehand. In this network, commodities are transported from one node to another within a certain period, depending on the release and due times of a container. Therefore, it is necessary to model time variables appropriately. In literature there are two ways to model time: continuous and discrete approaches. In [16, 19] a discrete approach is obtained by introducing a *time-space graph*.

Definition 3.2.1. Let X be a set of locations, and let $T \subset \mathbb{Z}^+$ be a finite set of time stamps. We call a graph $G = (V, E)$ a *time-space graph* if its node set V is of the form $T \times X$, and every (directed) arc $((t, x), (t', x')) \in E$ satisfies $t < t'$. We refer to the node (t, x) as location $x \in X$ at time $t \in T$.

Moreover, considering multiple vehicles $w_1, w_2, \dots, w_n \in W$ having the same travel time between locations x and $x' \in X$, then a *time-space multigraph* has to be used to distinguish arcs with the same begin and end node. The set of nodes

remains unchanged, but the set of (directed) arcs is somewhat extended. Every arc $e \in E$ is of the form

$$e = [(t, x), (t', x'), w] \in (T \times X) \times (T \times X) \times W, \quad (3.1)$$

where W is the set of vehicles and $t < t'$.

Example 3.2.1. In the base instance the set of locations is given by

$$X = \{T_{Origin}, T_{Rot}, T_1, T_2, T_3\}.$$

Let $T = \{0, 3, 6, \dots, 33, 36\} \subset \mathbb{Z}^+$ be a finite set of time stamps. In this example, we have access to two barges, denoted by B_{blue} and B_{green} , having capacity 20 and 10, respectively. The schedule of the barges is fixed. One barge, B_{blue} , will start at the container terminal and depart at time $t = 9$ to the single inland terminal, where it arrives at time $t = 30$. The other barge, B_{green} , will start at the single inland terminal and depart at time $t = 12$ to the container terminal, where it arrives at time $t = 33$. Furthermore, an unlimited number of trucks having capacity one is available. Although trucks can be used at any point in time, only the ones that might be useful are shown in Figure 3.1 (the red dashed arcs).

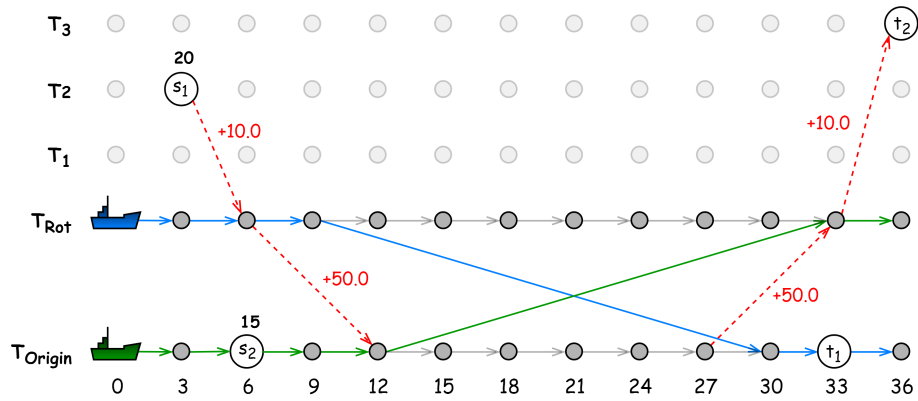


Figure 3.1: Example of the MCMCFP

We need to transport two orders k_1 and k_2 consisting of 20 and 15 containers, respectively. The source node s_1 marks the pick-up time and location of the inland order k_1 , and the sink node t_1 marks the due time and destination of the order. Equivalently, the source node s_2 marks the release time and location of the outland order k_2 , and the sink node t_2 marks the delivery time and destination of the order. We may assume that the cost for transporting a container by barge is zero, while cost is added per container that is transported by truck. Due to the fact that the model under consideration only has deterministic parameters, the only goal is to minimize the cost such that the containers reach their destination on time.

Every MCMCFP can be solved to optimality by rewriting the problem as an *integer linear programming (ILP)* problem. In contrast to the *single commodity flow problem*, where the linear programming relaxation always has an integer solution, the MCMCFP is NP-hard. NP-hardness can be no problem in practice, but the problems might become too big very quickly. Each problem consists of a time-space graph containing all the nodes and arcs in the network, and a set of commodities, denoted by K . Without loss of generality, we may assume that every commodity $k \in K$ has one source node s_k and one sink node t_k . For every arc $e \in E$, the parameter c_e is defined as the capacity of the arc. The parameters $f_{e,k}$ are defined as the per container cost of commodity $k \in K$ transported via arc $e \in E$. The parameters $d_{v,k}$ equal d_k if $v = s_k$, $-d_k$ if $v = t_k$ and zero otherwise, where d_k is the quantity of commodity k (i.e., the number of containers). The variables $x_{e,k}$ represent the magnitude of the

flow of commodity k on arc $e \in E$. The objective function is to minimize the cost using the parameters $f_{e,k}$.

Before stating the problem as an ILP, some notation is introduced. For every node $v \in V$, we define the set of *incoming* and *outgoing arcs* as

$$\delta^-(v) := \{e \in E \mid e = (v', v) \text{ for some } v' \in V\}, \quad (3.2)$$

$$\delta^+(v) := \{e \in E \mid e = (v, v') \text{ for some } v' \in V\}. \quad (3.3)$$

By making use of the notation, we can define the so-called *flow conservation constraints*, given by

$$\sum_{e \in \delta^+(v)} x_{e,k} - \sum_{e \in \delta^-(v)} x_{e,k} = d_{v,k}, \quad (3.4)$$

for all $v \in V$ and $k \in K$. The flow conservation constraints ensure that what flows into the node, must also flow out, except for the source and sink nodes.

The MCMCFP can now be written as an optimization problem in ILP form, as shown in (3.5).

$$\begin{aligned} \min_{x_{e,k}} \quad & \sum_{k \in K} \sum_{e \in E} f_{e,k} \cdot x_{e,k} \\ \text{s.t.} \quad & \sum_{e \in \delta^+(v)} x_{e,k} - \sum_{e \in \delta^-(v)} x_{e,k} = d_{v,k} \quad \forall v \in V, \forall k \in K \\ & \sum_{k \in K} x_{e,k} \leq c_e \quad \forall e \in E \\ & x_{e,k} \in \mathbb{Z}_{\geq 0} \quad \forall e \in E, \forall k \in K \end{aligned} \quad (3.5)$$

The problem described in Example 3.2.1 can be modeled and solved by means of the presented ILP. As illustrated in Figure 3.2, the container transport from the deep-sea terminal T_2 to the container terminal T_{Rot} , and from the container terminal T_{Rot} to deep-sea terminal T_3 have to be done fully by truck. The long trip, however, could be done largely by barge. Only five containers corresponding to the outland order have to be transported by truck to the container terminal T_{Rot} , because the size of the order exceeds the capacity of barge B_{green} . We end up with an objective value of 600.

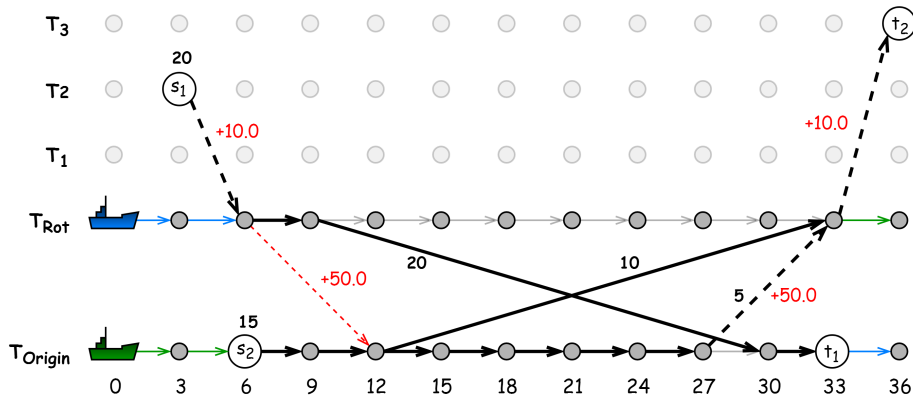


Figure 3.2: The optimal flows of commodities k_1 and k_2 (black arcs) on a time-space graph of the MCMCFP of Example 3.2.1

3.3 CONTAINER SCHEDULING AND VEHICLE ROUTING

The goal in problems III and IV is the determination of both the transportation timetables (i.e., vehicle routing) and container-to-mode assignment. In other words, the LSP must tell the barges and other vehicles where to go and when, and what containers to take with them. The only restrictions for the transportation modalities are the capacity and the initial position in the network.

In Problem III there are no unknowns, i.e., every container has got prespecified release and due time that are known in advance and no disturbances could occur. In other words, the field of intermodal freight transportation. The problem could be modeled as a *multi-commodity network design problem (MCNDP)* on a time-space graph [19], which is explained in detail in Section 4.2.

The MCNDP is only one of the variants of network design problems. In the paper of Johnson et al. [17] a universal definition of a network design problem is formulated: "Given a weighted undirected graph, we wish to find a subgraph which connects all the original vertices and minimizes the sum of the shortest path weights between all vertex pairs subject to a budget constraint on the sum of its edge weights."

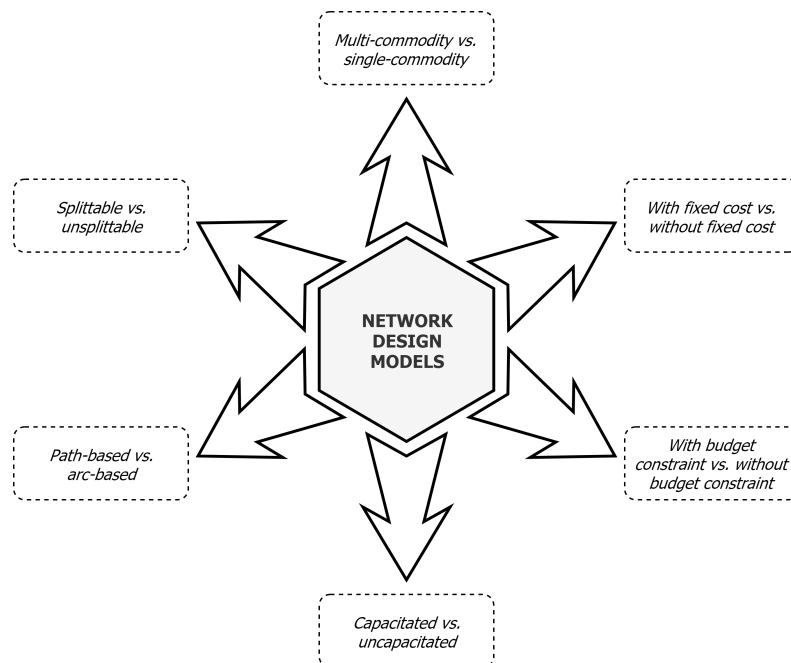


Figure 3.3: Different classifications of network design models, from [34]

As shown in Figure 3.3, network design models can be classified in many different ways [34]. Some models are uncapacitated, whereas some of them impose shared capacity on all of the commodities or capacities on each commodity. Some of the network design models have fixed cost and some of them have only the variable cost. In some models just one commodity must flow through the network, but in others we might have several commodities. Furthermore, some network design models are *path-based* [7, 8] and some of them are *arc-based* [12]. These two models are equivalent to each other in terms of the objective function. The objective of the path-based model is similar to the arc-based model, but the variable cost is the sum of flows on paths rather than arcs. That is to say, in the arc-based models there is a decision variable for every arc between any two nodes, while in the path-based model there exists a decision variable between every flow path from origin to destination. In addition, another variant is the unsplittable problem, in which each commodity must follow exact one route from the origin to the destination.

The network design problem we are interested in, is the multi-commodity, capacitated, arc-based and splittable variant without a budget constraint, but with fixed cost (e.g. for the use of trucks). To refer to the universal definition of Johnson et al., the weighted undirected graph in Problem III coincides with the graph containing every possible route of every vehicle (and every container). We wish to find a subgraph which connects every source (pick-up) and sink (delivery) node in the graph, such that the sum of the cost incurred is minimized. Between any two nodes the edge weight corresponds to the cost incurred if vehicle $w \in W$ travels that arc $e \in E$. Instead of a budget constraint, capacity constraints are taken into account.

There is theoretical evidence that capacitated network design problems are NP-hard [20, 21]. NP-hardness can be no problem in practice, but the problems we are interested in get too big very quickly. Therefore approximate methods are needed to solve them. However, such a solution method cannot guarantee the optimality of the solution found. *Heuristics* and *metaheuristics* are the two classes of approximate methods.

In the paper of Chouman and Crainic [4] an MIP-based heuristic for the designed-balanced capacitated multi-commodity network design problem is introduced. The heuristic combines a cutting-plane procedure that efficiently computes tight lower bounds and a variable-fixing procedure feeding an MIP solver. The idea of the heuristic can be used for all types of ILPs, but it is only effective if most variables turn out to be zero in feasible solutions of the ILP.

A variation to the MIP-based heuristic is proposed in the work of Kalicharan [19]. He introduces the α *B&C-and-fix* heuristic, which does branch and cut instead of only adding cutting planes. The first phase of the heuristic consists of doing branch and cut for the ILP, while saving the variables that are used in every node of the branch-and-cut tree. In the second phase the problem is solved restricted to the variables that were non-zero in at least α nodes of the branch-and-cut tree.

In Problem IV, the goal is the same but almost any element could be stochastic. In the work of Chiscop [3] a robust formulation is proposed to be able to do simultaneously vehicle routing and container-to-mode assignment uncluding uncertainty in the release times. It is assumed that the release times of the containers belong to an uncertainty interval, and no further statistical information is available. The robust solutions found with the model correspond to transportation plans which remain feasible for any realization of the release times within the prespecified uncertainty interval.

In the work of Rivera and Mes [28] a planning problem of selecting services and transfers in a synchronomodal network has been studied. Freight has to be transported from its origin to its destination, while minimizing the costs over a multi-period time window. At each decision moment, the planner can make three possible decisions for available freights at each location, where each decision incurs some form of cost. The possible decisions are given by

1. transport the freight to its destination,
2. transport the freight to an intermediate terminal (i.e., switch service),
3. postpone the transport of the freight.

The optimal balance between direct and future costs guarantees the best performance for the multi-period horizon. However, anticipating future cost is challenging. To model this stochastic and multi-period tradeoff, Rivera and Mes propose a *Markov decision process (MDP)* model. Moreover, to overcome the computational complexity of solving the MDP, an approximate dynamic programming (ADP) approach is proposed.

The paper by Fragkos et al. [12] generalizes the basic MCNDP to a multi-period setting, where demand for each commodity expands dynamically over a discrete time window. Arc activation decisions have to be taken in every single-period, and

once an arc is activated it can be used throughout the remaining horizon to route several different commodities. Though it is one of the few papers that introduces a multi-period setting, the decision making process is done at a strategic level instead of operational level. The decisions at strategic level have long-term goals, while we are interested in short-term decision making where plans can be changed last minute. Furthermore, each period the same commodities have to be transported from their origin to their destination, only the size of the commodities expands dynamically in time.

In the paper of Han et al. [13] a robust scenario approach is presented for the vehicle routing problem (VRP) with uncertain travel times in case where the exact probability distributions are not known, but the multiple range forecasts and the probability of the occurrence of each range are available. In the approach, a scenario is defined as a random travel time of each arc of the network being realized only in one single range. For each realization of a scenario, they first find the robust route that protects the solution against the worst case within the given ranges. Then the optimal route with respect to the minimum expected worst case cost is determined, where an expectation is taken over all scenarios. A branch-and-cut algorithm is proposed to solve the problem.

3.4 CONTRIBUTION MADE BY THIS WORK

The deterministic version of many network design problems has been studied extensively over the last decades. Due to the recognized practical importance of incorporating uncertainty, the uncertain version of network design problems has also attracted increasing attention. Various problems have been formulated depending on the uncertainty under consideration. For example,

1. uncertainty in customer presence,
2. uncertainty in demand,
3. uncertainty in travel time.

A comprehensive overview can be found in Cordeau et al. [6], and Häme and Hakula [15]. Our work extends the traditional multi-commodity network design problems by introducing a multi-period time window setting. Although such multi-period network design problems can provide useful input for strategic and tactical decisions, finding their optimal solution is computationally challenging. We propose a simulation based approach that explicitly seeks routes that are *resistant to change*² (i.e., vehicle routing), and implicitly effectuates a container-to-mode assignment. As mentioned before, only few papers are published that introduce a multi-period setting. Most of the papers that do introduce the multi-period setting, assume a fixed (time) schedule for the transportation modes in their network, which coincides with Problem II. The work of this thesis is related to Problem IV.

We attempt to obtain new insights and knowledge into synchromodal planning problems, including stochastic elements, by proposing a simulation based approach on a multi-period time window. To the best of our knowledge, this research is the first to address both vehicle routing (explicitly) and a container-to-mode assignment (implicitly), including uncertainty in the pick-up and delivery appointments, by generating potential future scenarios in order to obtain the best decision(s) that is *resistant to change*. The uncertainty element in our model is based on probability distributions, which has the benefits of incorporating distributional information and hence results in less moderate solutions than the classical robust optimization approaches where probability distributions are ignored, e.g., Han et al. [13].

² See Definition 5.4.1 in Chapter 5.

4

OFFLINE OPTIMIZATION

In the field of operations research a well known and commonly used distinction is made between *offline* and *online* optimization. In *offline* optimization all relevant information is known when solving the optimization problem involved, i.e., there is no uncertainty about any input data relevant to the problem. In *online* optimization the input data come in sequentially and decisions have to be taken while part of the relevant information is still lacking, since it will only become available after the decision has been made. Only *ex post*, when all information has become available, the truly optimal solution can be computed offline [23]. Since the offline solution can be used as a lower bound, we explore the field of offline optimization. The gap between the offline and online solutions can be used to assess the quality and practical relevance of the solution methods.

In this chapter we present a mathematical model to solve the optimization problem offline. Moreover, the mathematical model can be used as a starting point for solving the optimization problem online. Before doing so, an explanation is given about generating the instances that can be used to test the performance of the different solution methods.

4.1 CREATING IN- AND OUTLAND ORDERS

The multi-period time window into consideration covers 72 time steps, where each time step corresponds to three hours in real life. In other words, the goal is to transport orders from their pick-up location to their destination within a time frame of nine days. As stated in the problem description, each in- and outland order has its own characteristics. For an order $k \in K$, we will denote the pick-up location by $\chi_{pu}(k)$, the pick-up time by $\tau_{pu}(k)$, the delivery location by $\chi_{del}(k)$, the delivery time by $\tau_{del}(k)$ and the size of the order by $|k|$. Given any number of in- and outland orders, denoted by N^{in} and N^{out} respectively, the corresponding characteristics are created as follows.

Let $k \in K^{in}$ be an inland order. Notice that the delivery location is fixed, and the pick-up time consists of both a requested and a confirmed time. The pick-up location $\chi_{pu}(k)$ is chosen between T_1, T_2 and T_3 with equal probability. The requested pick-up time $\tau_{pu}^{req}(k)$ is discrete uniformly chosen between 1 and 59, i.e.,

$$\tau_{pu}^{req}(k) = \sim \mathcal{U}\{1, 59\}. \quad (4.1)$$

Thereafter, the confirmed pick-up time $\tau_{pu}^{conf}(k)$ is chosen using the probability vector

$$\mathbf{p} = (p_{-1}, p_0, p_{+1}, p_{+2}, p_{+3}, p_{+4}, p_{+5}), \quad (4.2)$$

where p_i corresponds to the probability that the confirmed appointment time is scheduled at time $\tau_{pu}^{req}(k) + i$. The due time $\tau_{del}(k)$ is discrete uniformly chosen

between $\tau_{pu}^{req}(k) + 13$ and $\tau_{pu}^{req}(k) + 20$, and may not exceed the right border of the multi-period time window¹, i.e.,

$$\tau_{del}(k) = \sim \mathcal{U} \left\{ \tau_{pu}^{req}(k) + 13, \tau_{pu}^{req}(k) + 20 \right\}. \quad (4.3)$$

In addition to that, the size of the order k is discrete uniformly chosen between 1 and 5, i.e., $|k| = \sim \mathcal{U}\{1, 5\}$.

Let $k \in K^{out}$ be an outland order. Notice that the pick-up location is fixed, and the delivery time consists of both a requested and a confirmed time. The delivery location $\chi_{del}(k)$ is chosen between T_1, T_2 and T_3 with equal probability. The pick-up time $\tau_{pu}(k)$ is discrete uniformly chosen between 0 and 62, i.e.,

$$\tau_{pu}(k) = \sim \mathcal{U}\{0, 62\}. \quad (4.4)$$

The requested delivery time $\tau_{del}^{req}(k)$ is discrete uniformly chosen between $\tau_{pu}(k) + 9$ and $\tau_{pu}(k) + 15$, and may not exceed the right border of the multi-period time window², i.e.,

$$\tau_{del}^{req}(k) = \sim \mathcal{U}\{\tau_{pu}(k) + 9, \tau_{pu}(k) + 15\}. \quad (4.5)$$

Thereafter, the confirmed delivery time $\tau_{del}^{conf}(k)$ is chosen using the probability vector \mathbf{p} , and may not exceed the right border of the multi-period time window. In addition to that, the size of the order k is discrete uniformly chosen between 1 and 5, i.e., $|k| = \sim \mathcal{U}\{1, 5\}$.

Although it has been mentioned before, notice again that the generated times correspond to the time steps in the model. For example, $\tau_{pu}^{req}(k) = 13$ is equivalent to saying that the requested pick-up time is 39 hours after the initial time at time step 0.

High and low priority orders

When the due time of an inland order turns out to be $\tau_{pu}^{req}(k) + 13$ and the confirmed appointment time is scheduled 5 time steps later, then the order has to be transported within 8 time steps, which can be seen as an order having *high priority*. When the due time turns out to be $\tau_{pu}^{req}(k) + 20$ and the confirmed appointment time is scheduled 1 time step earlier, then the order has to be transported within 21 time steps, which can be seen as an order having *low priority*.

In a similar way, when the requested delivery time of an outland order turns out to be $\tau_{pu}(k) + 9$ and the confirmed appointment time is scheduled 1 time step earlier, then the order has to be transported within 8 time steps, which can be seen as an order having *high priority*. When the requested delivery time turns out to be $\tau_{pu}(k) + 15$ and the confirmed appointment time is scheduled 5 time steps later, then the order has to be transported within 20 time steps, which can be seen as an order having *low priority*.

4.2 MULTI-COMMODITY NETWORK DESIGN PROBLEM

In this section a mathematical model is introduced to solve the optimization problem offline. As mentioned in the previous chapter, Problem III can be modeled as a *multi-commodity network design problem (MCNDP)*, in which both a container-to-mode assignment and vehicle routing is determined, and that is exactly the problem we are interested in. That is to say, solving the base instance described in Section 1.2

¹ If $\tau_{del}(k) > 72$, the order including all its characteristics is disregarded and a new order is generated.

² If $\tau_{del}^{req}(k) > 72$, the order including all its characteristics is disregarded and a new order is generated.

disregarding the uncertainty elements that appear in both the requested calls and the unannounced orders. First, the general MCNDP is presented as an integer linear program, then some practice-oriented constraints are added and, finally, the objective function is obtained using some practical information.

4.2.1 General MCNDP

The MCMCFP introduced in Section 3.2 can be extended in order to solve the MCNDP as offline problem. In the work of Kalicharan [19] an extra layer of constraints is introduced to provide a schedule for the barges of minimum cost which satisfies both the strict pick-up and delivery appointments at the deep-sea terminals, and the release and due times at the single inland terminal.

Because we assume that at any point in time, an infinite number of trucks is available at every terminal, we may disregard the routes of the trucks. The objective ensures that only the highly necessary truck arcs will be activated. The routes of the barges, however, are not known in advance. Therefore, we need to extend the edge space of the time-space graph. Let $\Delta_{x \rightarrow x'}$ denote the travel time (in terms of time steps) from location x to x' by barge. That is to say, in the base instance we have

$$\Delta_{x \rightarrow x'} = \begin{cases} 1 & \text{if both } x, x' \in T_{Rot} \cup D \text{ or } x = x', \\ 7 & \text{if one of the locations is the origin } T_{Origin}. \end{cases} \quad (4.6)$$

Now, the edge space of the time-space graph could be extended as follows. For every barge (or non-truck vehicle) $w \in W_{barge}$, for every time step $t \in T$, and for all locations $x \in X$ we need to add the arcs

$$e = [(t, x), (t + \Delta_{x \rightarrow x'}, x'), w], \quad (4.7)$$

for all $x' \in X$ if $t + \Delta_{x \rightarrow x'} \in T$. Thus, for each barge a huge number of possible arcs that could be travelled are added to the time-space graph. For the vehicle routing problem we aim for selecting the right arcs such that the network performance is optimal. Therefore, for every arc $e \in E_{barge} := \{e \in E \mid w \in W_{barge}\}$ we define *design variables* y_e , that are equal to 1 if the service at edge e is used, and 0 otherwise. In other words, a design variable is equal to 1 if and only if the corresponding barge travels the corresponding edge. Within the offline optimization process it will be decided which design variables are equal to 1, i.e., which route the barge should travel.

The previous model is extended by including the so-called *design-balanced constraints*. These constraints make sure that if a barge arrives at a certain location, then it will also depart from it, except for the source and sink nodes of the barge. The parameters $b_{v,w}$ equal 1 if v is the source node and -1 if v is the sink node³ of barge w , and 0 otherwise. The design-balanced constraints are given by

$$\sum_{e \in \delta^+(v) \cap E_{barge}} y_e - \sum_{e \in \delta^-(v) \cap E_{barge}} y_e = b_{v,w}, \quad (4.8)$$

for all $v \in V$ and $w \in W_{barge}$.

³ There is no restriction imposed for the sink node of a barge, but this can easily be solved by introducing a virtual sink node, where each possible sink node of barge w in the time-space graph is connected to the virtual sink node.

The MCNDP can now be written as an optimization problem in ILP form, as shown in (4.9).

$$\begin{aligned}
\min_{x_{e,k}} \quad & \sum_{k \in K} \sum_{e \in E} f_{e,k} \cdot x_{e,k} \\
\text{s.t.} \quad & \sum_{e \in \delta^+(v)} x_{e,k} - \sum_{e \in \delta^-(v)} x_{e,k} = d_{v,k} & \forall v \in V, \forall k \in K \\
& \sum_{e \in \delta^+(v) \cap E_{barge}} y_e - \sum_{e \in \delta^-(v) \cap E_{barge}} y_e = b_{v,w} & \forall v \in V, \forall w \in W_{barge} \\
& \sum_{k \in K} x_{e,k} \leq c_e \cdot y_e & \forall e \in E_{barge} \\
& \sum_{k \in K} x_{e,k} \leq c_e & \forall e \in E_{truck} \\
& x_{e,k} \in \mathbb{Z}_{\geq 0} & \forall e \in E, \forall k \in K \\
& y_e \in \{0, 1\} & \forall e \in E
\end{aligned} \tag{4.9}$$

Except the design-balanced constraints, other capacity related constraints are introduced. The capacity of an edge $e \in E_{barge}$ is zero if the barge does not travel the edge (i.e., if $y_e = 0$), and the capacity is equal to the capacity of the barge c_e if the barge does travel the edge (i.e., $y_e = 1$). In other words, the capacity of an arc depends on the design variables.

4.2.2 Adding practice-oriented constraints

Although the MCNDP can now be modeled and solved by means of the presented ILP, some extra constraints are required to be able to solve more practice-oriented problems. The main reason to add those extra constraints is to ensure that freight cannot suddenly switch vehicles. At the controlled terminals, constraints are added in the form of handling time, whereas at the region \mathcal{D} , another type of constraints is added to ensure that at most one barge can visit an appointment. We may distinguish between constraints for the origin, the container terminal and the deep-sea terminals within the region \mathcal{D} .

In the work of Kalicharan [19] only unloading time is taken into account. In case a barge has capacity left, it is still possible to pick-up containers at a terminal without any handling time. At the container terminal in Rotterdam it could happen that empty barges (or non-empty barges having capacity left) arrive, only load some freight and leave immediately. Therefore, both unloading and loading time have to be taken into account, implying some additional constraints are required. Before stating the unloading and loading constraints, Definitions 3.2 and 3.3 on the incoming and outgoing arcs have to be adapted a bit. For every node $v \in V$ and every vehicle $w \in W$, we define the set of *incoming arcs of type $w \in W$* and the *outgoing arcs of type $w \in W$* as

$$\delta_w^-(v) := \{e \in E \mid e = (v', v, w) \text{ for some } v' \in V\}, \tag{4.10}$$

$$\delta_w^+(v) := \{e \in E \mid e = (v, v', w) \text{ for some } v' \in V\}. \tag{4.11}$$

Given these adapted definitions, for all time-space nodes $v = (t, x)$ with $t \geq 1$, for all commodities $k \in K$ and for all vehicles $w \in W$, we add the *unloading constraints* to the model, given by

$$\sum_{e \in \delta_w^+(v)} x_{e,k} - \sum_{e \in \delta_w^-(v) \setminus \{\hat{e}_1\}} x_{e,k} \geq d_{v,k}, \tag{4.12}$$

where $\hat{e}_1 = [(t-1, x), (t, x), w]$. The constraints ensure that if a container wants to switch from vehicle $w \in W$ to $w' \in W \setminus \{w\}$, vehicle w has to wait at least one time step at the corresponding terminal having the container on board.

Similarly, for all time-space nodes $v = (t, x)$ with $t \geq 1$, for all commodities $k \in K$ and for all vehicles $w \in W$, we add the *loading constraints* to the model, given by

$$\sum_{e \in \delta_w^+(v) \setminus \{\hat{e}_2\}} x_{e,k} - \sum_{e \in \delta_w^-(v)} x_{e,k} \leq d_{v,k}, \quad (4.13)$$

where $\hat{e}_2 = [(t, x), (t+1, x), w]$. The constraints ensure that if a container wants to switch from vehicle $w \in W$ to $w' \in W \setminus \{w\}$, vehicle w' has to wait at least one time step at the corresponding terminal having the container on board. Observe that in case a container wants to switch from one barge to another, it has to wait at the container terminal at least two time steps, no matter what. Both barges cannot fulfill their duty to wait at the same time step, because the waiting process includes the restriction to have the container on board. For trucks, however, no handling time is taken into account. Thus switching from barge to truck, or vice versa, forces to wait only one time step.

At the origin it is sufficient to regard unloading (or loading) time only because we may assume no empty barges will arrive at the origin. What we try to avoid is that a barge, including some freight, could arrive at the origin and leave immediately taking some other freight. Although constraint 4.12 could be used, there is a more efficient way to implement the handling time at the origin. If design variable y_e is equal to 1 for some edge of the form $e = [(t, T_{Rot}), (t+7, T_{Origin}), w]$ for some barge $w \in W_{barge}$, then design variable $y_{e'}$ has to be forced to be equal to 1 as well for edge $e' = [(t+7, T_{Origin}), (t+8, T_{Origin}), w]$. In general, for all time steps $t \in T$ with $t+8 \in T$ and for all barges $w \in W$, we add the constraint given by

$$y_{[(t, T_{Rot}), (t+7, T_{Origin}), w]} \leq y_{[(t+7, T_{Origin}), (t+8, T_{Origin}), w]}. \quad (4.14)$$

The constraint ensures that a barge is forced to wait at least one time step at the origin after arrival.

Although handling time could be added to the pick-up and delivery appointments at the deep-sea terminals in the region \mathcal{D} , we may disregard that aspect in the model. The (discrete) time steps of three hours include handling time, waiting time and other unefficient components that vehicles face in the port of Rotterdam. On the other hand, constraints are needed to ensure that freight cannot suddenly switch vehicles at an appointment. Furthermore, the number of barges that may visit an appointment is restricted to 1 because of practice-oriented reasons. Therefore, for all time-space nodes $v = (t, x) \in V_{appointment}$, we add the constraint given by

$$\sum_{e \in \delta^-(v)} y_e \leq 1. \quad (4.15)$$

The constraint ensures that at most one of the design variables corresponding to the incoming arcs is equal to 1, i.e., at most one barge could visit the node.

Instead of restricting the number of barges, a more general variant can be implemented. The flow conservation constraints 3.4 could be somewhat extended for the time-space nodes $v = (t, x) \in V_{appointments}$. The constraints are given by

$$\sum_{e \in \delta_w^+(v)} x_{e,k} - \sum_{e \in \delta_w^-(v)} x_{e,k} = d_{v,k}, \quad (4.16)$$

for all $v \in V_{appointment}$, $k \in K$ and $w \in W$. The constraints allow multiple barges at an appointment, but do ensure that freight cannot suddenly switch vehicles anymore. As mentioned before, we consider the previous restriction, both for practice-

oriented and computational complexity reasons. Restriction 4.15 consists of only $|\delta^-(v)|$ variables, and only $|V_{\text{appointment}}|$ constraints have to be added. Restriction 4.16, however, consists of $|\delta_w^+(v)| + |\delta_w^-(v)|$ variables, and

$$|V_{\text{appointment}}| \cdot |K| \cdot |W|$$

constraints have to be added.

4.2.3 Determining the cost structure

The objective function is based on the costs incurred during the decision making process. The ratio between the potential costs incurred in the process has to represent the actual costs faced by the Dutch logistics service provider in practice. In collaboration with the Dutch LSP, an estimate of the costs per TEU⁴ for the main actions within the process are known. Since the actual costs were not allowed to make public because of confidentiality, the cost ratios are presented in Figure 4.1.

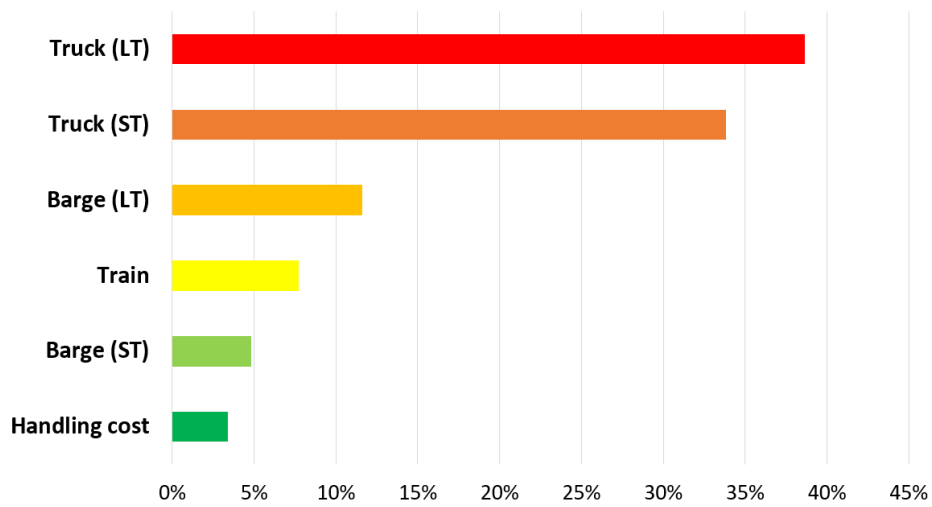


Figure 4.1: Cost per TEU incurred for the main actions. The abbreviations LT and ST stand for *long trip* and *short trip*, respectively.

As can be seen in the figure, trucks are relatively expensive, implying that the use of trucks should be minimized. Notice that the use of trains is disregarded in the model. The departure times of the train are fixed, so only an assignment of the containers has to be done. Furthermore, the train departs only once or twice a week, so does not impact the network that much.

Only additional costs are taken into account in the model, which means no costs are charged for the use of barges, because the barges will travel through the system regardless of the incoming orders. Only the use of trucks is taken into account. In practice, we could distinguish between 20, 40 and 45 ft containers, i.e., 1, 2 and 2 1/4 TEU respectively. In the base instance, however, no distinction is made between any type of containers. Every container has the same standardized dimensions, and the capacity of each vehicle is just measured in terms of the total number of containers.

Due to computational complexity reasons the handling cost at the container terminal is disregarded. Disregarding handling costs does not impair the representativeness of the model, since switching vehicles at the container terminal does implicitly incur additional costs, since the model does charge (handling) time.

⁴ TEU stands for Twenty-foot Equivalent Unit which can be used to measure a ship's cargo carrying capacity. The dimensions of one TEU are equal to that of a standard 20 ft shipping container, i.e., 20 feet long (≈ 6 meters) and 8 feet tall (≈ 2.5 meters).

Since the actual costs were not allowed to make public, the actual costs incurred for the use of trucks for long and short trips, say $C_{truck,LT}$ and $C_{truck,ST}$, have been multiplied by a random number ζ and rounded to a near integer. In the remainder of this thesis, the assumed costs for the use of trucks for long and short trips are given by

$$\bar{C}_{truck,LT} = 200.0 \approx C_{truck,LT} \cdot \zeta, \quad (4.17)$$

$$\bar{C}_{truck,ST} = 175.0 \approx C_{truck,ST} \cdot \zeta. \quad (4.18)$$

Given the practical information and assumptions made, the objective function can be formulated. The goal is to minimize the costs over all long and short trips travelled by truck, where a long trip corresponds to the transport of a container from the origin to the container terminal in Rotterdam or vice versa, and a short trip corresponds to the transport of a container between any two terminals in the port of Rotterdam. Therefore, the objective function is given by

$$\sum_{k \in K} \left(200 \cdot \sum_{e \in E_{long}} x_{e,k} + 175 \cdot \sum_{e \in E_{short}} x_{e,k} \right). \quad (4.19)$$

The set of flow arcs corresponding to the long trips travelled by truck is denoted by E_{long} , and equivalently the set of flow arcs corresponding to the short trips travelled by truck is denoted by E_{short} .

4.3 APPROACHING THE LOWER BOUND

After implementing the ILP described in the previous section, the offline optimal solution can be computed for various randomly generated instances and used as a lower bound for the problem. However, the computational time significantly increases for all instances considered when the number of orders and the length of the time window increases. In other words, as the number of decision variables increases significantly, it becomes quite time consuming to compute the offline solution. Moreover, we could ask ourselves if the offline solution can be helpful at all because of the substantial presence of the uncertainty elements. The uncertainty in the model is twofold. On the one hand, there is uncertainty in the requested appointment times that have to be confirmed. On the other hand, there is even more uncertainty in the orders that are not known yet. Since there is so much uncertainty involved in the model, it is unlikely that the online solution would come even a little bit close to the offline solution. Although we succeeded to gain the results for the offline solution, a different benchmark approach is proposed in Section 5.2 in order to approximate the lower bound. To investigate the benchmark approach is a well-considered choice because of three reasons. The approach

1. is more realistic in terms of computational time,
2. is more realistic in terms of usefulness, i.e., to assess the quality of the models,
3. may provide advantageous information by dynamically changing the amount of uncertainty.

5 | ONLINE OPTIMIZATION

In this chapter several solution methods will be explored to solve the base instance described in Section 1.2. In the base instance a MPTW of 72 time steps (i.e., nine days) is considered, where the data comes in sequentially only after the decision for the upcoming CTW has been made. Before doing so, some remarks on online optimization will be stated, whereafter a benchmark approach is proposed which is more realistic in terms of both computational time and usefulness.

5.1 SOME REMARKS ON ONLINE OPTIMIZATION

The idea of the proposed online optimization approach is as follows. Instead of solving the MPTW at once (by solving an immense ILP), multiple SPTWs need to be solved iteratively. Observe that after solving a SPTW, we move forward on the MPTW only the length of a CTW (as explained and visualized in Figure 1.5).

In the proposed benchmark approach, for example, the solution of each SPTW will be obtained by solving a single ILP. Although each ILP provides a planning for the upcoming SPTW, only the part within the CTW is actually performed. Each ILP (or SPTW) provides a piece of the solution and eventually the final solution is obtained. Observe that the ILPs (or SPTWs) cannot be solved simultaneously, because the input of each ILP (or SPTW) depends on the output of the previous one.

By solving a single ILP, we are guaranteed to find the optimum. However, by solving multiple ILPs iteratively there is no optimality guarantee. Although each single ILP is solved to optimality locally, it does not mean that the MPTW is solved to optimality globally. In fact, only in the unlikely case that all future realizations of the missing or stochastic input data happen to coincide with the imputed expectations, the online solution will match the offline solution.

5.1.1 The source node for shipped orders

In *offline* optimization, a source and a sink node in the time-space graph correspond, respectively, to the pick-up time and location, and the delivery time and location of an order. In *online* optimization, however, the source node does not always corresponds to the pick-up time and location of an order. It could occur that an order has been transported from the pick-up location to an intermediate location, implying that the source node corresponds to that intermediate location and the arrival time. In that case, such a source node might have some extra restrictions, e.g., if it is located on some barge or what its previous location was.

5.1.2 Varying the period between two decision moments

As mentioned before, we distinguish SPTWs and CTWs. The length of a SPTW depends on the relevant information known at an arbitrary decision moment. By the construction of the in- and outland orders (as explained in Section 4.1), the length of a SPTW could be at most 32 time steps since every due or delivery time lies in such a time frame. The length of a CTW, however, can be adjusted manually.

It is defined as the period between two consecutive decision moments, i.e., the part of the solution that is actually performed.

For example, the length of a CTW could be one or four time steps, implying that a decision moment is performed every three hours or every half a day, respectively. The first option would probably give better results, but the amount of work that has to be done increases significantly. In case of a decision moment at each time step, we need to solve 72 SPTWs, while in case of a decision moment every half-a-day, only 18 SPTWs need to be solved. Therefore, an appropriate balance between the solution performance and the running time performance needs to be found. The length of a CTW is denoted by δ . Moreover, the period δ between two consecutive decision moments defines what part of the solution, obtained by solving the SPTW, is actually stored.

Let the interval $[t_a, t_b]$ correspond to an arbitrary single-period time window, where $\delta \in \mathbb{N}_{\geq 1}$ such that $t_a + \delta < t_b$. For every arc $e = [(t, x), (t', x'), w] \in E$ satisfying $t \in [t_a, t_a + \delta)$, the flow variables $x_{e,k}$ and the design variables y_e that are non-zero are actually stored. Moreover, the non-zero variables corresponding to the arcs satisfying $t \in [t_a, t_a + \delta)$ and $t' \geq t_a + \delta$ determine the initial nodes of both the barges and the orders (having their pick-up time before $t_a + \delta$ and their delivery time after).

5.1.3 Unloading process

At the container terminal it is allowed to store containers temporarily or switch vehicles. However, handling time is taken into account for both the unloading and loading process. The unloading constraint ensures that if a barge wants to unload one or more containers, it has to wait at least one time step at the container terminal having the corresponding container(s) on board. In *offline* optimization, where plans are made beforehand, such a restriction does not cause any trouble. In *online* optimization, however, the constraint is weakened.

Suppose a certain barge arrives at the container terminal T_{Rot} and, based on the information available, it is decided to unload some containers on board. Then the barge is obliged to wait one time step at the container terminal including the containers on board. After waiting, new information becomes available and a new decision has to be taken. Due to the unloading constraint, it is now allowed to send the barge to any location without the containers under consideration on board. However, it is also allowed to send the barge to any location with the containers under consideration on board, which contradicts the decision taken at the previous time step. Therefore, if it is decided to unload some containers, constraints are needed to ensure that both the barge and the containers wait at least one time step at the container terminal, but without being on board of the barge.

Let the interval $[t_a, t_b]$ correspond to an arbitrary SPTW, and let $I := (t_a, T_{Rot})$ be the initial node at the container terminal within the time-space graph. We define W_I as the set of barges located at the container terminal T_{Rot} at time step t_a . In a similar way, we define K_I as the set of orders located at the container terminal T_{Rot} at time step t_a , where the order could be both on board of a barge $w \in W_I$ or on the quay of the container terminal.

For every barge $w \in W_I$ and for every order $k \in K_I$, the *realistic unloading constraints* are defined by at most three constraints, given by

$$\sum_{x \in X} x_{e_1, k} - x_{e_2, k} = 0, \quad (5.1)$$

where

$$\begin{aligned} e_1 &= [(t_a + 1, T_{Rot}), (t_a + 1 + \Delta_{T_{Rot} \rightarrow x}, x), w], \\ e_2 &= [(t_a, T_{Rot}), (t_a + 1, T_{Rot}), w]. \end{aligned}$$

Furthermore, if order $k \in K_I$ is on board of barge $w \in W_I$, we need to add two more constraints, given by

$$\sum_{x \in X \setminus \{T_{Rot}\}} x_{e_3, k} = |k| \cdot \sum_{x \in X \setminus \{T_{Rot}\}} y_{e_3}, \quad (5.2)$$

$$x_{e_2, k} + x_{e_4, k} = |k| \cdot y_{e_2}, \quad (5.3)$$

where

$$\begin{aligned} e_3 &= [(t_a, T_{Rot}), (t_a + \Delta_{T_{Rot} \rightarrow x}, x), w], \\ e_4 &= [(t_a, T_{Rot}), (t_a + 1, T_{Rot}), \text{wait}].^1 \end{aligned}$$

Constraint 5.1 ensures that if a container is not unloaded at the beginning of the single-period time window, it is forced to stay on the barge at least one extra time step. Constraint 5.2 ensures that order k is forced to stay on board of barge w when the barge decides to leave the container terminal at time step t_a (because it is not unloaded). Finally, constraint 5.3 ensures that the $|k|$ containers corresponding to order k (i.e., the magnitude of the flow of order k) can be fully or partially unloaded when the barge decides to stay at the container terminal.

5.2 THE BENCHMARK APPROACH

As mentioned before, the computational time grows substantially if the number of decision variables increases. Although it is quite time consuming to compute the problem offline, we succeeded to gain the lower bound to the problem. However, in order to better assess the quality of the models and to gain some advantageous information about the impact of the uncertainty element on the system, a benchmark approach model is proposed. In the model, the lower bound is approximated using SPTWs in which there is no uncertainty in the requested appointment times. Only uncertainty is available in the orders not announced yet. The solution of each SPTW is obtained by solving a single ILP, whereafter we move forward on the MPTW the length of a CTW. The length of the CTW is set to one, i.e., $\delta = 1$.

We assumed that orders become known 12 time steps in advance. In practice those orders include uncertainty. However, in order to approximate the lower bound to the problem, we may disregard this uncertainty element, implying that orders are confirmed immediately after they become known: the *B12 model*. Even better approximations can be found when the orders are announced at an earlier stage: the *B16*, *B20*, *B24* and *B28 model*.

To clarify, the *Bx model* coincides with the method in which orders become known x time steps in advance relative to their requested appointment time. The greater the value of x , the more (deterministic) information available at each decision moment, which will presumably lead to better performance (and increasing computational time).

Only *ex post*, when all information has become available, the truly optimal solution (i.e., lower bound) can be computed offline. In other words, the *Bx model* in which

¹ The arc that corresponds to staying at the container terminal for 1 time step without being on board of any vehicle, i.e., on the quay.

orders become known 72 time steps in advance relative to their requested appointment times: the *B72 model*.

Figure 5.1 presents the average cost, obtained by using the objective function (4.19), and the average number of trucks used in 11 randomly generated instances.

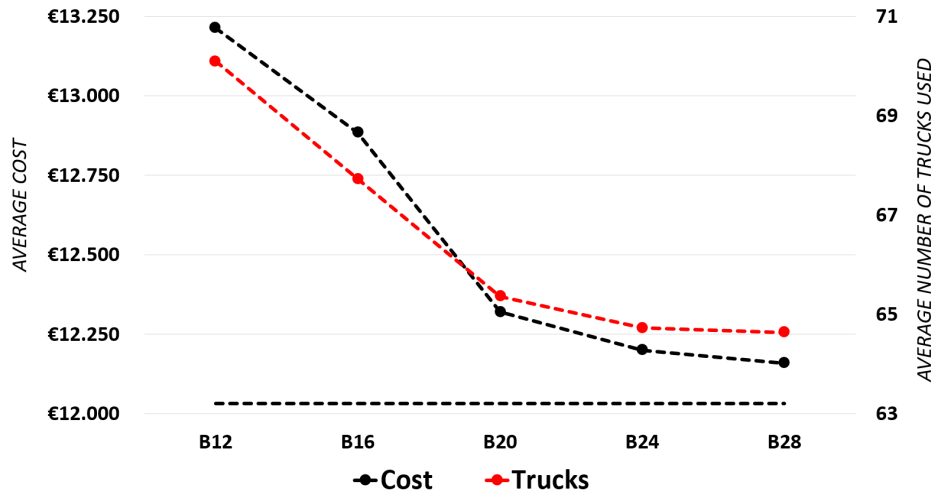


Figure 5.1: Approaching the lower bound by announcing orders at an earlier stage, where the announcement time corresponds to the number of time steps in advance. The horizontal dashed line corresponds to the average cost obtained in the *B72 model*, i.e., the offline solution obtained *ex post*.

As can be seen, the results gradually converge to the results of the *B72 model* (i.e., the offline solution that can be obtained *ex post*). Observe that the number of trucks smoothly decreases, while the average cost suddenly drops down at the *B20 model*. It seems that the focus is shifted more to the use of cheaper trucks (i.e., the short trips). In Chapter 6 a more detailed analysis on the benchmark models is presented.

5.3 CONFIRMATION BASED MODELS

In this section three solution methods will be presented: the *RC*, *EC* and *AC model*. The approach of the models is similar to the *B12 model*, in which the solution of each SPTW is obtained by solving a single ILP, whereafter we move forward on the MPTW the length of a CTW. However, this time the uncertainty element of the requested appointment times has to be taken into account. In the models this uncertainty element is partially disregarded, by assuming that each requested appointment time will be scheduled at a specific time relative to the requested one. In other words, the models are based on a confirmation that is made up by ourselves.

5.3.1 RC model

The *RC model* (*Requested as Confirmed*) is the most obvious method to solve the problem. Here the uncertainty is fully disregarded. In other words, the requested appointment times are assumed to be the confirmed ones. Given that assumption, the solution of each SPTW is obtained solving a single ILP, whereafter the part of the solution within the CTW is actually stored.

Observe that a SPTW could be infeasible when a faulty decision was made at the previous decision moment. This can happen in two different ways. If $\delta \geq 2$, it might be possible that a container arrives at the container terminal T_{Rot} at the confirmed appointment time, implying that the container is too late (i.e., it cannot be delivered on time anymore). This could happen when the container is sent last minute by

barge, relative to the requested appointment time, and the confirmed appointment time is scheduled 1 time step earlier. Another possibility for infeasibility is based on the container assignment at the origin, which is clarified by Example 5.3.1.

Example 5.3.1. Suppose that at decision moment $t = 4$ we decide to send a barge from the origin to the container terminal including three assigned outland orders $k_1, k_2, k_3 \in K^{out}$. The characteristics of the orders that matter are given by

$$\tau_{del}^{req}(k_1) = 10, \quad \tau_{del}^{conf}(k_1) = 12 \quad \text{and} \quad \chi_{del}(k_1) = T_3 \quad (5.4)$$

$$\tau_{del}^{req}(k_2) = 12, \quad \tau_{del}^{conf}(k_2) = 14 \quad \text{and} \quad \chi_{del}(k_2) = T_1 \quad (5.5)$$

$$\tau_{del}^{req}(k_3) = 15, \quad \tau_{del}^{conf}(k_3) = 14 \quad \text{and} \quad \chi_{del}(k_3) = T_2 \quad (5.6)$$

An order is confirmed 8 time steps in advance relative to the requested appointment time, implying that outland order k_3 is the only order not confirmed yet. Observe that the confirmation is done per order, not per ship². Thus, at the decision moment we assume that the appointment times are given by 12, 14 and 15, implying that the orders can be sent by the same barge. In Figure 5.2 the planned route of the barge is visualized.

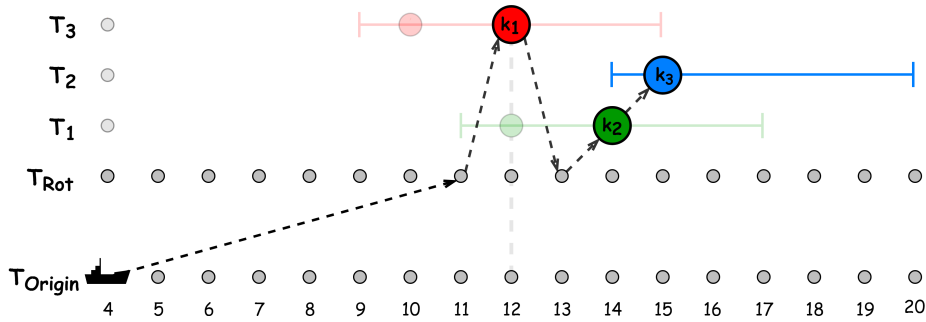


Figure 5.2: The planned route of the barge at decision moment $t = 4$, that will become infeasible if the requested appointment corresponding to order k_3 is confirmed at time step $t = 7$ (and turns out to be one time step earlier).

However, when time passes by, outland order k_3 is confirmed and turns out to be scheduled at the same time as order k_2 , but at different locations. Since there is no time left to unload some containers, the single-period time window at decision moment $t = 7$ (when order k_3 is confirmed) will become infeasible.

If a single-period time window is infeasible, the order of smallest quantity, say $k_{infeasible}$, is sent by truck retroactively and an additional *penalty* is charged. That is to say, an amount of $|k_{infeasible}| \cdot (175.0 + 200.0)$ is added to the objective value since the order is sent by truck from its origin to its destination (i.e., both a long and short trip). In addition, the penalty imposed is equal to the same amount, because we may assume that the LSP has to pay the cost incurred to the client (or the port of Rotterdam) to reimburse the 'damage' (i.e., lateness). In summary, if an order $k_{infeasible}$ causes infeasibility, then $|k_{infeasible}|$ penalties are taken into account. For each penalty, an amount of

$$c_{penalty} = 2 \cdot (175.0 + 200.0) = 750.0 \quad (5.7)$$

is added to the objective function (4.19).

² If an requested appointment time is confirmed, the mode(s) of transportation that will be used to pick-up or deliver the order can still be chosen freely.

5.3.2 EC model

For inland orders, the RC model works fine. For the outland orders, however, the model has some disadvantages. As mentioned before, when the confirmed appointment time of an order turns out to be earlier than the requested one, and the model had decided (based on the requested appointment time) to send the order last minute to its destination, possibly unnecessary costs are incurred because the order has to be trucked. Especially orders of large size may cause problems.

To ensure such problems will not occur, the *EC model* (*Earliest as Confirmed*) is proposed. The model assumes that the confirmation of each requested appointment time is the 'worst case'. In other words, the confirmed appointment time is assumed to be the earliest possible appointment time.

Observe that the problem faced in the RC model does not relate to the inland orders, so it would probably not benefit to assume the earliest possible appointment time for both type of orders. Therefore, the requested appointment time belonging to an inland order is assumed to be the average appointment time.

Although the first possibility of infeasibility is avoided, the second possibility based on the container assignment at the origin could still occur.

5.3.3 AC model

For both the RC and EC model, to deal with the uncertainty, the assumption is made to regard the requested appointment times in the beginning of the uncertainty interval to ensure that the transportation of outland orders by barge is possible (and no unnecessary costs are incurred). However, in most cases (to be precise five out of seven³) the actual confirmed appointment time will be scheduled later than the requested one.

Naturally, we do not charge any cost for being on time, but when an outland order arrives at the container terminal way too early it is not beneficial. The containers corresponding to the order could stay on board of the barge the remaining time or the containers could be (partially) unloaded at the container terminal. The main drawback of the first option is the unnecessary use of the capacity, implying that some other orders cannot be loaded (fully) on the barge. Moreover, a pick-up appointment can only be visited if the delivery time does not collide with the delivery time of the outland order. The second option does avoid those drawbacks, but another disadvantage does appear. In the model, handling time is taken into account for both the unloading and loading processes at the container terminal, implying that the barge has to wait at least one time step after arrival at the container terminal. After the order has been unloaded, the order could be trucked to its destination, implying that cost has to be taken into account, or the order could be loaded on another (or the same) barge at a later moment, and transported to its destination without any cost, implying that this barge is forced to wait at least one time step at the container terminal as well. In other words, the model does not charge cost for being way too early, but the model does charge time, which could (again) lead to extra costs.

Therefore, the *AC model* (*Average as Confirmed*) is added as a third model in order to investigate if shifting to the middle of the interval (i.e., the average) does benefit. Observe that the drawback of the RC model does emerge even more, and infeasibility could occur again in both ways.

³ In the experiments, the probability vector p is uniformly distributed.

5.4 SIMULATION MODEL

Alternatively to the simpler models above, a much more complex algorithm is developed in which future scenarios are simulated for the requested appointment times given their probability vector p . A *future scenario* (or *realization*) is not a specific forecast of the future, but a plausible description of what might happen. By analysing various possible future scenarios, the planning and decision making process will be more efficient. For example, given two potential decisions for some barge (i.e., routing and container assignment), say decision I and II, decision I might perform better for certain scenarios, while decision II achieves better results for some other scenarios.

Within the decision making process, the goal is to find the best decision(s) for each CTW that is *resistant to change*, such that a proper solution is obtained for the entire MPTW.

5.4.1 Definitions

In order to clarify the goal, the emphasized expression is defined in Definition 5.4.1. Moreover, the term (*sub*)*optimal*, mentioned in the definition, is described in Definition 5.4.2 because it is interpreted differently here than in literature.

Definition 5.4.1. Let \mathcal{F} be a set of potential future scenarios (or realizations). A decision is called *resistant to change* if the decision is feasible and (sub)optimal for every potential future scenario $f \in \mathcal{F}$.

Definition 5.4.2. Let N_D be the number of potential decisions at decision moment t , and let \mathcal{F} be a set of potential future scenarios. For every future scenario $f \in \mathcal{F}$, the objective values obtained for the N_D potential decisions⁴ can be reordered as

$$\mathcal{O}_1 \leq \mathcal{O}_2 \leq \dots \leq \mathcal{O}_{N_D}. \quad (5.8)$$

A decision i is called *optimal* if the objective value $\mathcal{O}_i = \mathcal{O}_1$, and decision i is called *suboptimal* if the gap between the objective value \mathcal{O}_i and the optimal objective value \mathcal{O}_1 is less than 10%.

The term *robustness*⁵ can be used to assess the quality of two different models that perform similarly in terms of average cost. Before we define the expression, it is clarified by means of Example 5.4.1.

Example 5.4.1. Suppose experiments for three various instances are performed to assess the quality of two different models, denoted by A_1 and A_2 . The models strive to minimize a cost function. Let A^* be the lower bound to the problem, i.e., the solution obtained *ex post*. The results are presented in Table 5.1.

	A^*	A_1	$A_1 - A^*$	A_2	$A_2 - A^*$
<i>Instance 1</i>	300.0	400.0	100.0	305.0	5.0
<i>Instance 2</i>	400.0	500.0	100.0	425.0	25.0
<i>Instance 3</i>	200.0	300.0	100.0	470.0	270.0
Mean	300.0	400.0	100.0	400.0	100.0
S.D.	100.0	100.0	0.0	85.3	147.6

Table 5.1: Example to clarify the term *robustness*

As can be seen, both models perform similarly in terms of average cost. The mean of model A_i is implicitly equivalent to the mean of $A_i - A^*$, where $i = 1, 2$. The standard deviation, however, does not relate at all. The standard deviation of

⁴ If decision i is infeasible, we set \mathcal{O}_i equal to ∞ .

⁵ The term *robustness* is interpreted differently than in literature.

the models A_1 and A_2 itself is not relevant, but the standard deviation of $A_1 - A^*$ and $A_2 - A^*$ certainly is. It is a measure to quantify the amount of variation or dispersion of a set of data values. As shown in Table 5.1, model A_1 is much more steady than model A_2 . Although model A_2 exceeds the other model for the first two instances, the model gains bad performance for the third one. In case of similar performance in terms of average cost, a more steady model is preferred, i.e., the model that is *more robust*. That is one of the reasons why the results in Chapter 6 will be presented as the difference in results of the benchmark and the proposed solution methods.

Definition 5.4.3. Given two models, say A_1 and A_2 , and a set of corresponding data values. The model A_1 is called *more robust* than model A_2 if

$$\sigma(A_1 - A^*) < \sigma(A_2 - A^*),$$

where σ denotes the standard deviation and A^* the lower bound, i.e., the offline solution obtained *ex post*.

Observe that the term *robustness* is mainly advantageous to compare models that gain similar performance in terms of average cost.

5.4.2 Start of the algorithm

The decision making process consists of both a routing of the barges and a container assignment to the barges (and trucks) such that the total cost is minimized over the entire multi-period time window. Since the problem is twofold (vehicle routing and container-to-mode assignment), the decision space grows rapidly. To be able to manage this immense space, the container-to-mode assignment is not included in the decision space explicitly, but is taken into account afterwards. In other words, the routing problem is solved *explicitly* and the container-to-mode assignment is obtained *implicitly*.

By using a flowchart of the algorithm, shown in Figure 5.3, the simulation based model is presented and carefully explained. Due to complexity reasons, the length of a CTW is set to one, i.e., $\delta = 1$.

Although a MPTW consisting of 72 time steps is solved, it is sufficient to consider only 55 decision moments. At the 55th decision moment (i.e., at time step $t = 54$) no uncertainty is involved anymore. Just by the construction of the in- and outland orders, every requested appointment time is scheduled before or at time step 62, implying that every order is confirmed at time step 54. Therefore, the remaining interval $[54, 72]$ can be solved offline.

In the flowchart, the decision moment is denoted as *period number* (*PeriodNr*), but it is equivalent terminology. For each period number less than 54, the algorithm checks if a decision has to be made at all, which is almost always the case. Only if all barges are on their way from the origin to the container terminal or vice versa, the decision moment can be skipped until a barge arrives at one of the locations.

5.4.3 Decision space

In the base instance, a set of three barges is taken into account. At each decision moment t_a , each barge could be located at the origin (t_a, T_{Origin}) , the container terminal (t_a, T_{Rot}) or one of the deep-sea terminals $(t_a, T_i)_{i \in \mathcal{I}}$ in the region \mathcal{D} . Additionally, a barge could be on the move from the origin to the container terminal or vice versa, in case the initial node of the barge is, respectively, (t, T_{Rot}) or (t, T_{Origin}) for some $t_a + 1 \leq t \leq t_a + 6$. In the latter case no decision has to be made for the barge under consideration.

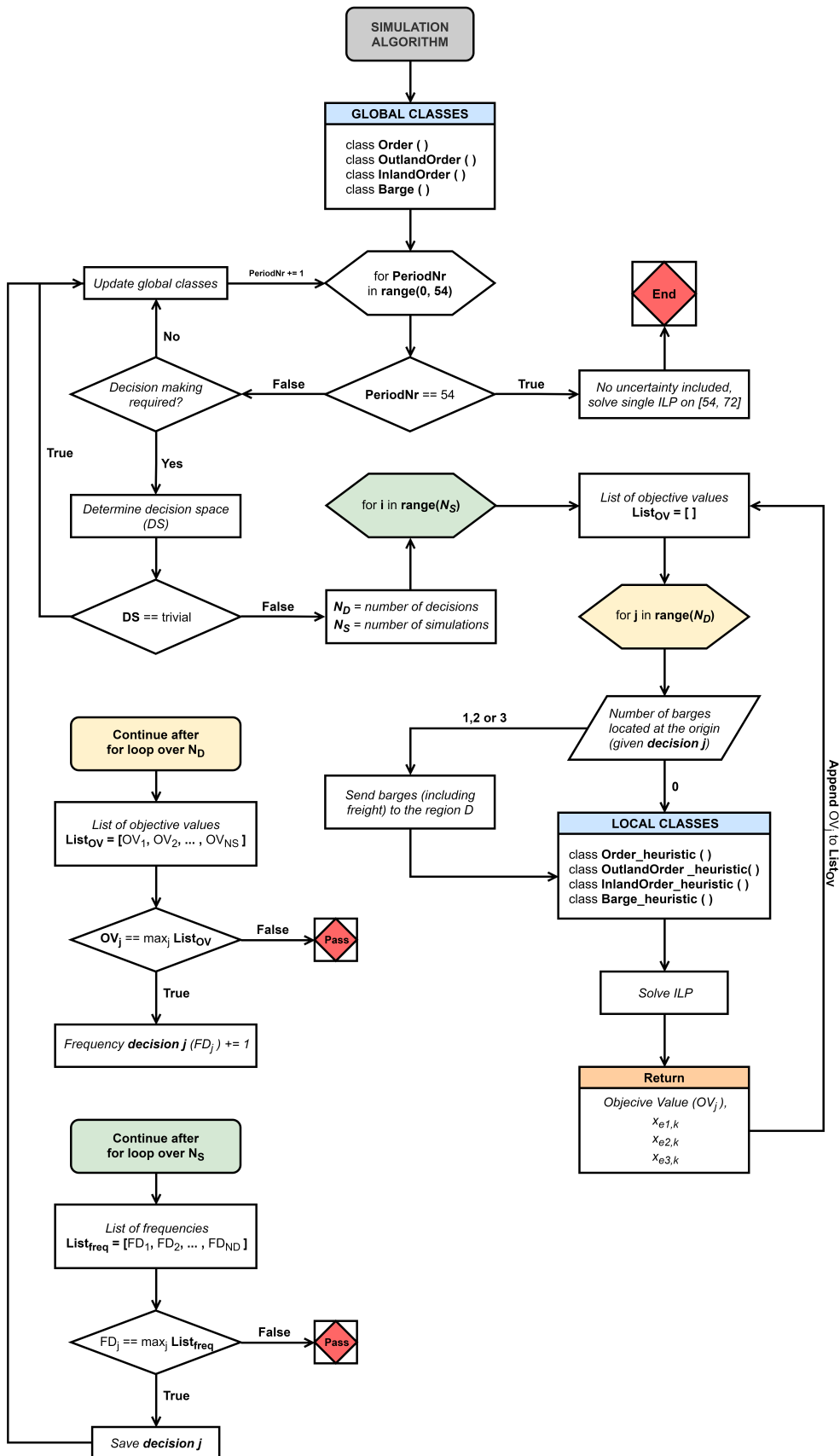


Figure 5.3: Flow chart of the simulation algorithm

If a barge is located at the origin, two possible decisions can be made. The barge could depart to the container terminal or stay another time step at the origin. If a barge is located at the container terminal, at most five possible decisions can be made. The barge could depart to the origin, it could stay at the container terminal or the barge could visit one of the three deep-sea terminals in case an appointment is scheduled at the upcoming time step. In case a barge is located at one of the deep-sea terminals, at most four possible decisions can be made. The barge could depart to the container terminal or it could visit one of the three deep-sea terminals in case an appointment is scheduled at the upcoming time step. From a theoretical point of view it could happen that 5^3 decisions could be made. However, in practice such a scenario would never happen. Moreover, due to symmetry, the number of decisions can be reduced drastically. During the performed experiments, on average 3.14 decisions could be made at each non-trivial decision moment, having a maximum of 20 decisions. In the model, a decision is denoted as a triple

$$(decision_{B1}, decision_{B2}, decision_{B3}), \quad (5.9)$$

where the i -th element corresponds to the (potential) upcoming location of barge i . Observe that the notation excludes the container-to-mode assignment.

Trivial decisions

Quite often there is only one possible direction for a barge, implying that the decision is trivial and fixed on forehand. Decisions could be trivial in different ways. In case no appointment is scheduled at the upcoming time step, a barge located at one of the deep-sea terminals can only return to the container terminal⁶. Moreover, in case a pick-up appointment was scheduled at the current location, the order has to be (partially) assigned to the barge based on its remaining capacity. In case a barge, located at the region \mathcal{D} , does have some containers on board corresponding to an outland order having its delivery appointment at the upcoming time step, the barge is obliged to visit the appointment. Observe that decisions belonging to delivery appointments must be taken at an earlier stage. Furthermore, the decision is trivial when a barge is located at the origin and its previous location was the container terminal, i.e., the barge just arrived. Since a barge must stay at least one time step at the origin to unload the containers on board (and possibly load some new containers), the decision is fixed.

Decision (t_a, T_{Rot}) to $(t_a + 1, T_{Rot})$

Although at first sight this might seem a trivial decision, it is not. At the container terminal a barge is allowed to unload or load some containers, implying that the decision includes the assignment of the containers located at the container terminal and on board of the barge itself. Even orders may be split into suborders, which causes some extra difficulties. Even if there is only one possible decision, the simulation process has to be done to reveal what containers to load and unload. For each potential decision, in which at least one barge decides to stay at the container terminal, we keep track of how frequently an order is transported by that barge. If so, the order could be transported fully or partially. Therefore, the number of containers per order is recorded as well. In the end, orders that occur in more than half of the simulations are taken into account in the final decision. The actual quantity equals the average number of containers transported by barge, rounded to the nearest integer.

⁶ In case a delivery appointment is scheduled at the upcoming time step, but the barge under consideration does not have any containers of the corresponding order on board, the appointment can be neglected.

Decision $(t_a, Origin)$ to $(t_a + 7, T_{Rot})$

At the origin, the only prerequisite is that a barge has to wait at least one time step after arrival (to unload the containers on board). After this, however, no loading time is taken into account, implying that the assignment is not based on the freight on board of the barge. Hence we need to determine what orders are, fully or partially, transported to the container terminal by the barge under consideration. As will be described in Section 5.4.4, the ILP used to find the (estimated) objective value for each possible decision is a heuristic excluding the container flow from the origin to the container terminal. The container-to-mode assignment has to be done manually. Just as for the previous decision, we keep track of the frequency an order is transported by that barge, including the number of containers per order. In the end, orders that occur in more than $\alpha\%$ of the simulations are taken into account in the final decision. This percentage has to be significantly higher than before, because we want to ensure that no conflicting appointments could occur, which might lead to infeasibility. During the experiments, α is set to 95, 90 and 85.

Remaining decisions

The container assignment for the remaining decision space is straightforward. For example, if a barge is located at the container terminal T_{Rot} and a potential decision is: departure to the origin. In that case, only the containers on board of that barge are forced to be transported to the origin. Just like for the delivery appointments mentioned above, the assignment of the containers to the barge has to be done at an earlier stage. In a similar way, the container-to-mode assignment for other decisions is based on the freight on board of the barge.

5.4.4 Solving the ILP

If the decision is non-trivial, a prespecified number of simulations N_S is performed to seek the best decision(s) that is *resistant to change*. For every future scenario (or simulation) and every decision, an ILP has to be solved. In case the number of simulations and decisions increases, the computational time significantly increases. Therefore a less time consuming heuristic is preferred. The output of each ILP does not have to be precise, because we are only interested in finding the best decision(s), given a possible future scenario. Therefore, an estimated objective value is sufficient. The time-space graph is modified in two ways: the length of the interval is decreased and the nodes at the origin are merged, which is illustrated in Figure 5.4.

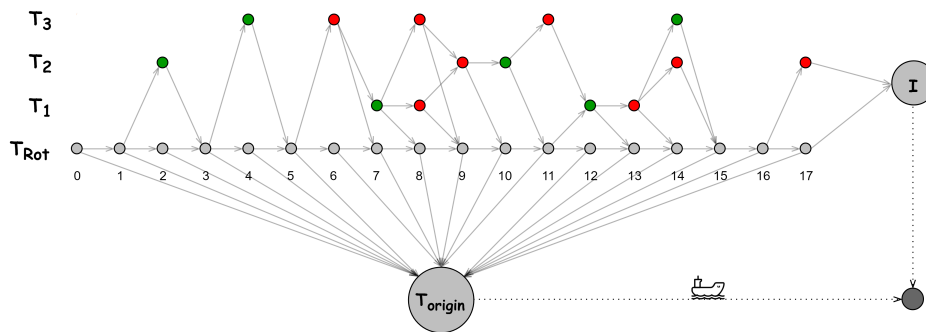


Figure 5.4: Visualization of the network setting in the ILP heuristic. Observe that the node in the bottom right corner corresponds to the virtual sink node of the barges (since there is no restriction imposed for the end point of the barges).

Decrease interval length

Instead of 32 time steps, the interval has shrunk to only 17 time steps, which will be denoted by $[t_a, t_b]$, where $t_b = t_a + 17$. Because of this modification it could occur that the delivery time of some in- and outland orders lie outside the interval. To deal with those orders, an additional (virtual) node I is added to the time-space graph. For the outland orders⁷, this node can be seen as a universal delivery node. The node can be reached both by barge and truck, and cost is taken into account for the use of trucks. The cost c_k depends on the (requested) delivery time $\tau_{del}^{req}(k)$ of the order k . Given the requested delivery time $\tau_{del}^{req}(k) \in [t_a + 18, t_a + 32]$, the cost incurred is given by

$$c_k = \left\lfloor -12.5 \cdot \left(\tau_{del}^{req}(k) - t_a \right) + 400.0 \right\rfloor. \quad (5.10)$$

Because of this dependence, it is ensured that orders are handled in the correct sequence. In addition, the outland orders located at the end of the interval do not affect the outcome that much, since we may assume that these orders are likely to be picked up by one of the barges after the shortened interval. Similarly, the (virtual) node I can be used by inland orders having a delivery time⁸ after $t_a + 24$. Given the delivery time $\tau_{del}(k) \in [t_a + 25, t_a + 32]$, the cost incurred is given by

$$c_k = \left\lfloor -\frac{1}{7} \cdot (200.0 \cdot (\tau_{del}(k) - t_a) + 6400.0) \right\rfloor. \quad (5.11)$$

Again, using this linear cost function it is ensured that orders having a delivery time sooner are more crucial in the model than the ones at the end of the interval.

A minor downside is that a barge may transport both in- and outland orders to the invisible node, even if the due time of the inland order conflicts with the appointment time of the outland order.

Merge the origin nodes

The second modification is merging the nodes at the origin as one universal origin, and removing the arcs from the origin T_{Origin} to the container terminal T_{Rot} . Thus, in the adapted time-space graph it is not possible anymore to route any barge, truck or container from the origin to the container terminal. Therefore, a preprocessing phase has to be carried out, where all the barges are sent to the region \mathcal{D} before solving the ILP. This will be explained in the following subsection.

Although no barges are located at the origin after the preprocessing phase, it could occur that some outland orders still have to be picked up at the origin. That is why the set of outland orders is distinguished into two subsets: outland orders at the region \mathcal{D} , denoted by $K_{\mathcal{D}}^{out}$, and outland orders at the origin, denoted by K_{Origin}^{out} . The outland orders located at the region \mathcal{D} can be treated in the usual way, i.e., the source and sink node correspond to the pick-up time and location, and the delivery time and location of the order. The outland orders located at the origin, however, can never reach their sink node, because the arcs from the origin to the container terminal are removed. In order to deal with those orders, some things have to be adapted to the source and sink nodes.

Suppose an outland order $k \in K_{Origin}^{out}$ has to be delivered at time step $\tau_{del}(k)$. If the order is transported by barge from the origin to the container terminal (and from the container terminal to its destination), the barge under consideration has to arrive at the container terminal before or at time step $\tau_{del}(k) - 1$, implying that the barge has to depart from the origin before or at time step $\tau_{del}(k) - 8$. Including

⁷ Observe that only outland orders located at the region (after ‘sending the barges’) are taken into account, but that is clarified later on.

⁸ Observe that a delivery time for an inland order corresponds to a due time, implying that the order can be delivered at an earlier stage as well.

unloading time at the origin, we may conclude that the barge under consideration has to arrive at the origin before or at time step $\tau_{del}(k) - 9$. Therefore, in the ILP heuristic, the source and sink node of each outland order $k \in K_{Origin}^{out}$ are given by

$$s_k = (t_a, T_{Rot}), \quad (5.12)$$

$$t_k = (\tau_{del}(k) - 9, T_{Origin}). \quad (5.13)$$

Moreover, the capacity of the barge has to be taken into account, but not for the long trip from the container terminal to the origin, because no containers corresponding to an outland order are on board of the barge during such a trip. Therefore, except the capacity constraints for the inland orders, a second capacity constraint is added for each barge, which can be seen as a copy of the capacity what can be filled *virtually*.

Sending the barges to the region

Let W_{Origin} be defined as the set of barges located at the origin at the start of decision moment t_a . Each barge $w \in W_{Origin}$ has an *initial availability*, denoted by $t_{w,0}$ which is an element of the set $\{t_a + 1, \dots, t_a + 8\}$. The initial availability is defined as the initial time step a barge could depart to the container terminal in Rotterdam. For example, if for a certain potential decision, barge w is sent from the container terminal to the origin, then the barge arrives at time step $t_a + 7$. However, the barge has to wait at least 1 time step to unload the freight on board, implying that the initial availability equals $t_{w,0} = t_a + 8$.

For each barge $w \in W_{Origin}$, we need to decide when the barge departs and what containers are assigned to it. In case multiple barges are located at the origin, the order of the decisions is based on the initial availability (and capacity) of the barges. Since the outland orders are announced 12 time steps in advance relative to their release time at the origin, the space of possible departure times is given by $S = \{t_{w,0}, \dots, t_a + 12\}$, because it is not beneficial to depart after time step $t_a + 12$. In order to make a suitable choice, we define the vectors ρ_{cap} and ρ_{loss} consisting both of $|t_a + 12 - t_{w,0}|$ elements.

For every $t \in \{t_{w,0}, \dots, t_a + 12\}$, we define $\rho_{cap}(t)$ as the potential capacity if the barge departs at time step t , and $\rho_{loss}(t)$ as the 0-1 vector that tells us if time step t is the very last option for some order to be transported by barge (i.e., a *critical order*). In other words,

$$\rho_{cap}(t) := \text{the number of containers that could be sent by the barge}, \quad (5.14)$$

$$\rho_{loss}(t) := \begin{cases} 0 & \text{if no critical orders,} \\ 1 & \text{if at least one critical order.} \end{cases} \quad (5.15)$$

The pseudo code of the algorithm to construct the elements $\rho_{cap}(t)$ and $\rho_{loss}(t)$ can be found in the appendix: Algorithm A.1.

Given these two vectors, we aim for a subset $S_{potential} \subseteq S$ of potential departure times for the barge. The subset is obtained by applying Algorithm 5.1. The algorithm seeks the time steps at which the capacity is optimal used without losing any containers, i.e., critical orders.

Example 5.4.2. Suppose we need to determine the space of potential departure times at decision moment $t = 36$, where the initial locations of the three barges $\mathcal{B}_1, \mathcal{B}_2$ and \mathcal{B}_3 (having capacity 20, 10 and 10) are given by

$$(41, T_{Origin}), \quad (36, T_{Rot}) \quad \text{and} \quad (36, T_{Origin}),$$

respectively. Moreover, the previous locations were T_{Rot}, T_{Rot} and T_{Origin} , implying that barge \mathcal{B}_3 may depart immediately, but barge \mathcal{B}_1 has to wait at least one time

Algorithm 5.1: Determine the space of potential departure times $S_{potential}$

```

for all elements  $t$  in  $\{t_{w,0}, \dots, t_a + 12\}$  do
  if  $\rho_{cap}(t) = cap_w$  then
    Append  $t$  to  $S_{potential}$  and break the for loop!
  else if  $\rho_{loss}(t) = 1$  and  $\rho_{cap}(t) >$  highest value so far then
    "Check the previous values of  $\rho_{cap}$ .."
    if  $t = t_{w,0}$  then
      Append  $t$  to  $S_{potential}$ !
    else
      "Check if same capacity can be achieved earlier.."
      while  $\rho_{cap}(t) = \rho_{cap}(t - 1)$  do
         $t \leftarrow t - 1$ 
      end while
      if  $t \notin S_{potential}$ , append!
    end if
  end if
end for

```

step. In other words, the initial availabilities are $t_{B_3,0} = 36$ and $t_{B_1,0} = 42$. Based on the initial availability, we get started with barge B_3 . After applying Algorithm A.1, the following two vectors are found.

$$\rho_{cap} = (6, 7, 10, 10, 10, 9, 9, 4, 8, 8, 8, 9, 9), \quad (5.16)$$

$$\rho_{loss} = (0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1). \quad (5.17)$$

After applying Algorithm 5.1, we find $S_{potential} = \{37, 38\}$. In general, to determine the space of potential departure times for barge B_1 , $|S_{potential}|$ different subproblems are created. Therefore, two different subproblems are created in which the orders transported by barge B_3 at time step 37 and 38, respectively, are removed. By removing the orders transported at time step 37, we get

$$\rho_{cap} = (8, 3, 7, 7, 7, 8, 8), \quad (5.18)$$

$$\rho_{loss} = (1, 0, 0, 0, 1, 0, 1). \quad (5.19)$$

And by removing the orders transported at time step 38, we get

$$\rho_{cap} = (4, 4, 8, 8, 8, 9, 9), \quad (5.20)$$

$$\rho_{loss} = (0, 0, 0, 0, 1, 0, 1). \quad (5.21)$$

Again, by applying Algorithm 5.1, we may conclude that the space of potential departure times is given by

$$S_{potential} = \{(37, 42), (38, 44), (38, 47)\}, \quad (5.22)$$

where each tuple corresponds to the potential departure times of barge B_3 and B_1 respectively. As can be seen, the problem is divided into three subproblems. Instead of solving one *normal ILP*, we need to solve three *heuristic ILPs*, where the outcome having minimum objective value is the actual solution.

Although Example 5.4.2 suggests that the number of heuristic ILPs increases significantly (compared to the normal ILPs), it is not. Given $S_{potential}$, the simulation algorithm keeps track of the outcome of each ILP and stops automatically if the outcome increases too much, implying that the costs incurred by the critical orders influences the outcome too much. During the experiments only 1.36 heuristic ILPs had to be solved on average. A more detailed analysis on the running time profit can be found in Chapter 6.

6

COMPUTATIONAL RESULTS

In order to compare the different solution methods developed, we have conducted experiments for 11 randomly generated instances. In Section 4.1 it was explained how these instances were generated. First, the offline results are presented in which the lower bound is approximated by removing one part of the uncertainty in the model, as described in Section 5.2. Additionally, the lower bound itself is included as well. Thereafter, the online results are presented, interpreted and discussed.

The models described in Chapter 5 have been implemented in Python and solved with the commercial solver CPLEX 12.7 through the Python API. The experiments were conducted using 16 cores of 2.4 GHz each, working with 16 GB of RAM.

6.1 OFFLINE RESULTS

In this section the results are presented on approaching the lower bound to the problem, explained in Section 5.2, in which the uncertainty in the requested appointment times is disregarded. In practice, barge planners announce orders 12 time steps in advance relative to their requested appointment times: the *B12 model*. By shifting the announcement time of the orders to an earlier stage, even better approximations can be found: the *B16*, *B20*, *B24* and *B28 model*. The truly optimal solution (i.e., the lower bound) that be computed offline is denoted by *B72*.

To check how steady the solutions perform, the complete process is repeated 11 times. At each process $N^{in} = 30$ inland and $N^{out} = 30$ outland orders were generated. On average 91.3 containers corresponding to outland orders and 89.5 containers corresponding to inland orders had to be transported from their origin to their destination¹.

The results obtained for the *Bx* models in which only one part of the uncertainty is taken into account, including the *B72* model, are given in Table 6.1.

	<i>Cost (€)</i>	<i>Trucks</i>	<i>Short trips</i>		<i>Long trips</i>	
	Mean	Mean	Out	In	Out	In
B12	13213.6	70.1	17.4 (24.8%)	14.8 (21.1%)	19.5 (27.8%)	18.5 (26.3%)
B16	12884.1	67.7	12.9 (19.1%)	13.5 (20.0%)	20.9 (30.9%)	20.4 (30.1%)
B20	12320.5	65.4	15.8 (24.2%)	14.3 (21.8%)	18.5 (28.2%)	16.8 (25.7%)
B24	12200.0	64.7	16.1 (24.9%)	13.7 (21.2%)	18.3 (28.2%)	16.6 (25.7%)
B28	12159.1	64.6	15.2 (23.5%)	15.5 (24.1%)	17.8 (27.6%)	16.1 (24.9%)
⋮						
B72	12031.8	63.7	13.7 (21.5%)	14.8 (23.3%)	18.5 (29.1%)	16.6 (26.1%)

Table 6.1: Approaching the lower bound by shifting the announcement time to an earlier stage. The average cost incurred using the objective function (4.19), the average number of trucks used in general, and more detailed the average number of trucks used for both short and long trips, and out- and inland orders are given. The distribution of the type of cost in parenthesis.

¹ The sample standard deviation of the number of containers was 10.6 and 6.0, respectively.

As can be seen, the average cost (and number of trucks used) decreases as the orders are announced at an earlier stage, and gradually converge to the B72 model. Moreover, observe that the focus indeed shifted to the cheaper trucks (i.e., short trips) from the B20 model, as perceived in Figure 5.1. Instead of 41.3 long trips, only 35.3 long trips were carried out by trucks.

Although the *optimality gap* between the B28 and B72 model is only 1.06%, there is quite some uncertainty left in the models. To clarify this, the amount of information available at the beginning of the multi-period time window in each model, expressed in percentages, is shown in Table 6.2. As can be seen, more than half of the information is still missing at the initial decision moment in the B28 model.

	B12	B16	B20	B24	B28	B32	B36	B40	B44	B48	B52
%	19.7	26.3	32.8	39.4	46.0	52.5	59.1	65.6	72.2	78.8	85.3

Table 6.2: The percentages of the average number of orders known at the beginning of the MPTW, i.e., at decision moment $t = 0$

Something else stands out in Table 6.1. If we look at the last two columns, corresponding to the long trips, we expected that the number of trucks used for these types of trips would decrease even further in the B72 model. However, the offline results, obtained using the B72 model, show the opposite.

As stated before, the benchmark approach is more realistic in terms of both computational time and usefulness. The computational difficulty is discussed in the following subsection. The B12 model will be used to assess the quality and practical relevance of the solution methods, since it is the most realistic benchmark for the problem. Moreover, we might benefit from analyzing the performance of the even better models (B16, B20, B24, B28 and B72). By implementing common features in the results of these models, the performance of the proposed simulation model might exceed the B12 model. Although, the benefits of the common features probably do not outweigh the impact of the uncertainty element.

Computational time

In order to obtain the results in Table 6.1, multiple SPTWs had to be solved iteratively². As the orders are announced at an earlier stage, the length of the SPTW increases and so does the computational time. Nevertheless, the number of SPTWs that had to be solved reduces. In Table 6.3 the number of SPTWs that had to be solved and the average computational time of the ILPs are presented. Observe that in the benchmark approach solving a SPTW coincides with solving a single ILP (excluding some processing of the in- and output).

	ILP	#SPTWs	MPTW
B12	48.41 s	51	1.00 h
B16	79.45 s	47	1.52 h
B20	122.13 s	43	2.19 h
B24	179.95 s	39	3.27 h
B28	270.06 s	35	4.41 h
⋮			
B72	81063.12 s	1	22.52 h

Table 6.3: The average computational time (in seconds) of the ILPs solved during the experiments, where the length of the SPTW is based on the announcement time of the model. Besides that, the number of SPTWs and the average runtime (in hours) to solve the entire MPTW is given.

² Observe that the length of a CTW is set to one, i.e., $\delta = 1$.

What stands out from the results above is the increasing computational difficulty of the models. Given the results of the B12, B16, B20, B24 and B28 models, we were interested beforehand in the running time of the B72 model. By continuing the procedure of shifting the announcement time, assuming the running time has around the same exponential growth (as in the first five rows), solving the MPTW instance at once (i.e., the B72 model) would probably take around 4 hours. Nevertheless, we expected beforehand already that the running time would probably explode in practice.

Indeed, during the experiments³, solving the B72 model took almost 24 hours! As mentioned in Section 3.3, the NP-hardness of the network design problems can be no problem in practice, but the problems we are interested in get too big very quickly, as experienced.

6.2 ONLINE RESULTS

In this section the online results of the different solution methods are presented, interpreted and compared to the results obtained for the benchmark approach models. As stated before, the results are presented as the difference in results of the B12 model and the proposed solution methods. Based on 11 randomly generated instances of the problem, Tables 6.4 and 6.5 give an overview of the performance of the proposed solution methods.

<i>Cost difference (€)</i>			
	Mean	S.D.	Gap
B72	-1181.8	780.4	-8.9%
⋮			
B28	-1054.5	968.4	-8.0%
B24	-1013.6	1004.9	-7.7%
B20	-893.2	851.9	-6.8%
B16	-329.5	647.9	-2.5%
B12	0.0	0.0	0.0%
SIM	1647.7	1665.1	12.5%
AC	2134.1	1744.2	16.2%
RC	3102.3	2000.2	23.5%
EC	3815.9	1334.6	28.9%

Table 6.4: Comparison of the solution methods as the difference in results of the B12 model and the other models. The average and the standard deviation of the cost incurred using both the objective function (4.19) and the penalty function (5.7), and the gap in terms of percentages are given.

As can be seen in Table 6.4, the difference between the B12 model and the simulation model is positive in terms of the average cost and the average number of trucks used, implying that the B12 model has overall better results. Although there was an impression to exceed the B12 model, the impact of the uncertainty element is too much, as expected.

On the other hand, the simulation model surpasses the performance of the simpler models. With the exception of one instance⁴, the results for the simulation model were within 20% of the B12 model. As shown in the table, the optimality gap was 12.5% on average, where for some instances the simulation model performed even better than the B12 model (containing less uncertainty).

³ Observe that the experiments were conducted using 16 cores of 2.4 GHz each, so using a standard laptop it would probably take several days.

⁴ For one instance the gap was 50.1%. However, the other models RC (64.4%) and AC (60.8%) performed even worse. The uncertainty in the requested appointment times probably affects the model a lot.

Additionally, the AC model performed much better than expected in terms of average cost. Moreover, using Table 6.5, the AC model is the solution method that used on average the least number of trucks for the long trips corresponding to outland orders, which is the part dealing the most with the uncertainty element.

	<i>Trucks diff.</i>		<i>Short trips</i>		<i>Long trips</i>		ϵ
	Mean	S.D.	Out	In	Out	In	
B72	-6.4	5.4	-3.6	0.0	-0.9	-1.8	-
\vdots							
B28	-5.5	5.5	-2.2	0.7	-1.6	-2.4	-
B24	-5.4	5.8	-1.3	-1.1	-1.2	-1.8	-
B20	-4.7	5.0	-1.5	-0.5	-1.0	-1.6	-
B16	-2.4	4.2	-4.5	-1.3	1.5	1.9	-
B12	0.0	0.0	0.0	0.0	0.0	0.0	-
SIM	8.6	8.7	3.5	-0.3	6.1	-0.6	0.0
AC	11.3	9.3	3.6	0.5	4.7	1.0	0.4
RC	16.1	10.7	0.1	2.0	5.3	3.6	1.3
EC	19.2	7.4	0.5	-0.2	13.1	4.7	0.3

Table 6.5: Comparison of the solution methods as the difference in results of the B12 model and the other models. The average and standard deviation of the number of trucks used in general, and more detailed the average number of trucks used for both short and long trips, and out- and inland orders are given. The last column indicates the average number of penalties ϵ .

What further stands out in Table 6.5 is that there is only one positive value in the upper three rows (disregarding the B72 model), corresponding to the detailed trip columns. At first sight, we might conclude that from a certain point, when enough information is available, the focus is shifted to the other type of trips. Although the B72 model refutes the argument, it shows that short trips corresponding to inland orders do not benefit if the amount of information is increased.

As stated in Definition 5.4.3, to assess if model A_1 is *more robust* than model A_2 , we need to regard the standard deviation of the results as the difference in results of the models and the lower bound. So far, the results have been presented as the difference between the B12 model and the solution methods. However, in order to obtain reliable results about the *robustness* of the models, it is more fair to represent the results as the difference between the actual lower bound (i.e., B72 model) and the solution methods, as presented in Table 6.6.

	<i>Cost difference (€)</i>	
	Mean	S.D.
SIM - B72	2829.5	1435.5
AC - B72	3315.9	1461.6
RC - B72	4284.1	1571.3
EC - B72	4997.7	953.9

Table 6.6: The solution methods as the difference in results of the B72 model and the other models in order to compare the robustness more fairly

Observe that the EC model is the most robust solution method, but also has the worst performance in terms of average cost, mainly caused by the long trips corresponding to outland orders. Disregarding the EC model, the simulation model surpasses the simpler models both in terms of average cost and robustness. Besides that, no penalties occurred for the simulation model, whilst for the other models penalties did occur. We may conclude that the simulation model is evidently the best solution method in terms of both performance and robustness.

Trucks can be used for four different types of trips: long and short trips, and trips transporting a container corresponding to an out- or an inland order. In Table 6.7 the distribution of the type of costs incurred is presented in terms of percentages.

	<i>Short trips</i>		<i>Long trips</i>		ϵ
	Out	In	Out	In	
B72	21.5%	23.3%	29.1%	26.1%	-
\vdots					
B28	23.5%	24.1%	27.6%	24.9%	-
B24	24.9%	21.2%	28.2%	25.7%	-
B20	24.2%	21.8%	28.2%	25.7%	-
B16	19.1%	20.0%	30.9%	30.1%	-
B12	24.8%	21.1%	27.8%	26.3%	-
SIM	26.4%	18.5%	32.4%	22.6%	0.0%
AC	25.8%	18.8%	29.7%	23.9%	1.8%
RC	20.3%	19.5%	28.7%	25.6%	5.9%
EC	20.0%	16.4%	36.5%	26.0%	1.2%

Table 6.7: The distribution of the type of costs incurred, using both the objective function (4.19) and the penalty function (5.7)

We already saw that in the benchmark approach models the least number of trucks were used for the short trips corresponding to inland orders. However, the percentages in Table 6.7 corresponding to the solution methods turn out to be even lower. Apparently, including the uncertainty element of the requested appointments, the pick-up of inland orders at the deep-sea terminals is an easy way to minimize the costs at each SPTW, i.e., *locally*. Logically, it makes perfect sense. Visiting a pick-up appointment by barge can be decided last minute (only enough capacity is required), while visiting a delivery appointment (or carrying out a long trip) has to be decided already at an earlier stage because the containers have to be loaded on board of the barge somewhere in between.

In the problem we face, the transportation of containers from the origin to Rotterdam (i.e., long trips for outland orders) is the part dealing the most with the stochastic elements in the model. As can be seen, both the AC and RC models gain better performance than the simulation model relative to that specific trip. This can be explained quite easily, because of the risk factor taken into account in the simulation model. There is a certain trade off between the amount of risk we dare to take and the performance, which is clarified by Example 6.2.1.

Example 6.2.1. Suppose that in both the RC and simulation model, at decision moment $t = 10$, we decide to send the same barge from the origin to the container terminal T_{Rot} . At the origin, three orders $k_1, k_2, k_3 \in K^{out}$ can be assigned to the barge, having characteristics

$$\tau_{del}^{req}(k_1) = 16, \quad \tau_{del}^{conf}(k_1) = 18 \quad \text{and} \quad \chi_{del}(k_1) = T_3 \quad (6.1)$$

$$\tau_{del}^{req}(k_2) = 18, \quad \tau_{del}^{conf}(k_2) = 20 \quad \text{and} \quad \chi_{del}(k_2) = T_1 \quad (6.2)$$

$$\tau_{del}^{req}(k_3) = 19, \quad \tau_{del}^{conf}(k_3) = 21 \quad \text{and} \quad \chi_{del}(k_3) = T_2 \quad (6.3)$$

In the RC model, the uncertainty element is not taken into account and the requested appointment times are just assumed to be the confirmed ones. Hence it is assumed that the orders have to be delivered at time step 18, 20 and 19, respectively, because k_3 is not confirmed yet. Therefore, all three orders are allocated to the barge (and luckily no problems will occur).

In the simulation model, the uncertainty element is taken into account and various future scenarios are generated and analysed in order to make the best decision

that is *resistant to change*. Suppose 25 simulations are performed, and in 6 out of 25 cases the simulated appointment time of order k_3 conflicts with one of the (confirmed) appointments of k_1 and k_2 . In the simulation model, only orders that occur in more than $\alpha\%$ of the simulations are taken into account in the final decision. If $\alpha \geq 76$, then order k_3 is left behind at the origin, and is possibly sent by truck (or later on by another barge).

The example shows that disregarding the uncertainty could lead to better assignment, but in combination with a risk factor. That is why the RC model incurs 1.4 penalties on average per instance, while the simulation model is much more reliable, because no penalties occurred during the experiments. Moreover, the simulation model has better performance overall. We may conclude that the benefits of taking into account the future scenarios outweigh the downside of the safety level α .

6.2.1 Sensitivity analysis on the number of simulations

In the simulation algorithm several parameters can be tuned: the number of simulations performed per non-trivial decision, denoted by N_S , the number of simulations after which a check is performed to throw away zero-frequency decisions (i.e., reduce decision space), denoted by X_{check} , and finally the safety level α .

During the experiments, presented in the tables above, the settings of the simulation model were

$$N_S = 25, \quad X_{check} = 11 \quad \text{and} \quad \alpha = 90. \quad (6.4)$$

In other words, in case a barge decides to route from the origin to the container terminal, then the actual container-to-mode assignment is based on the containers that were assigned to the barge in at least 90% of the simulations. However, what is the minimum number of simulations needed such that performance would not deteriorate? Therefore, an analysis is performed on the number of simulations.

In total, 467 non-trivial decision moments are taken into account. The final (or actual) decision can be categorized into six different classes. At each decision moment, the final decision could be the best decision in 25 out of 25 simulations (including some other decision(s) having non-zero frequency). The final decision could be the best decision in 11 out of 11 simulations (and all other decisions have zero frequency). Additionally, the final decision could be the best decision in 8-11, 12-15, 16-19 or 20-24 out of 25 simulations. In that case we are interested in the simulation number after which the final decision became the best decision and did not lose that position anymore. In Table 6.8 an overview is given of the different classes.

Class	Occurrence	Mean	Best from
25	56	25 / 25	1
11	207	11 / 11	1
[8, 11]	2	8.5 / 25	24.0
[12, 15]	25	14.2 / 25	13.0
[16, 19]	59	17.8 / 25	5.9
[20, 24]	118	22.0 / 25	4.1

Table 6.8: An overview of the different classes. The number of occurrences for each class, the average number of simulations the final decision was the best decision, and the average simulation number after which the final decision became the best decision (and did not lose that position anymore) are presented.

As can be seen, in 81.6% (381 out of 467) of the cases the decision space contains a decision which turns out to be the best decision (or one of the best) in at least 80% of the simulations. Moreover, at 73.7% (344 out of 467) of the non-trivial decision

moments, the final decision moment is already the best one (and stays the best) after running only one simulation.

In order to come up with an appropriate answer, we are mainly interested in the other 26.3% (123 out of 467) of the non-trivial decisions moments were the final decision becomes the best one after at least two or more simulations. In Table 6.9 an overview is given of the different classes where the non-trivial decision moments that were the best after only one simulation are disregarded.

Class	Occurrence	Mean	Best from
[8, 11]	2	8.5 / 25	24.0
[12, 15]	24	14.2 / 25	13.5
[16, 19]	40	17.6 / 25	8.2
[20, 24]	57	21.9 / 25	7.5

Table 6.9: An overview of the different classes disregarding the non-trivial decision moments that we are not interested in

Disregarding the two outliers, we may conclude that it should be enough to perform around 15 simulations. Although, notice that the analysis disregards the container assignment at the origin and the container terminal, that are based on the outcomes of the various future scenarios. Therefore, by reducing the number of simulations, the quality of the container assignment could be affected.

For the same 11 randomly generated instances, the simulation algorithm is carried out again, by setting

$$N_S = 15, \quad X_{check} = 7 \quad \text{and} \quad \alpha = 85 \quad (6.5)$$

In other words, in case a barge decides to route from the origin to the container terminal, then the actual container-to-mode assignment is based on the containers that were assigned to the barge in at least 85% of the simulations.

In Table 6.10 the results are presented as the difference in results of the B12 model and the simulation models with different settings. The *SIM25* and *SIM15 model* correspond to the models in which, respectively, 25 and 15 simulations were performed each non-trivial decision moment.

	<i>Cost diff. (€)</i>		<i>Trucks diff.</i>		<i>Short trips</i>		<i>Long trips</i>		ϵ
	Mean	S.D.	Mean	S.D.	Out	In	Out	In	
SIM25	1647.7	1665.1	8.6	8.7	3.5	-0.3	6.1	-0.6	0.0
SIM15	2215.9	1855.9	11.4	9.5	2.5	-0.2	8.5	0.5	0.0

Table 6.10: Comparison of the simulation models with different settings as the difference in results of the B12 model. The average and standard deviation of the costs and number of trucks used in general, and more detailed the average number of trucks used for both short and long trips, and out- and inland orders are given. The last column indicates the average number of penalties ϵ .

As can be seen, the SIM15 model cannot meet the performance of the SIM25 model. The extra costs are mainly caused by the increase in the number of trucks used for the long trips, as expected. As mentioned before, the reduced number of simulations affects the quality of the container assignment at the long trips. On the other hand, we could benefit in terms of computational time, which is discussed in the upcoming subsection.

6.2.2 Computational time

Tables 6.11 and 6.12 give an overview of the computational time of the solution methods, including the already treated benchmark models. However, some additional information is given and some results are written down again for comparison.

<i>Running time per..</i>					
	MPTW	SPTW	\sum ILPs	% on ILPs	#ILPs
B72	22.52 h	81063.12 s	22.52 h	100.0%	1
\vdots					
B28	2.74 h	270.06 s	2.63 h	96.1%	35
B24	2.05 h	179.95 s	1.95 h	94.9%	39
B20	1.55 h	122.13 s	1.46 h	94.0%	43
B16	1.12 h	79.45 s	1.04 h	93.0%	47
B12	0.75 h	48.41 s	0.69 h	91.6%	51
AC	0.81 h	48.31 s	0.74 h	91.5%	55
RC	0.79 h	46.94 s	0.72 h	91.4%	55
EC	0.76 h	45.26 s	0.69 h	90.7%	55

Table 6.11: Comparison of the computational time of the simpler models, including the benchmark models. The average running time per MPTW and per SPTW – which is equivalent to the average running time of the ILPs – are given. Further, the average running time spent on ILPs each instance including the percentages, and the number of SPTWs that had to be solved each instance is denoted.

As can be seen in Table 6.11, the simpler models are quite similar to the B12 model. Only some more ILPs (i.e., SPTWs) had to be solved because of the stochastic element in the requested appointment times. Furthermore, as the length of the SPTW increases, the percentage spent on solving ILPs converges to 100%.

	<i>Running time per..</i>					<i>Amount of..</i>	
	MPTW	SPTW	ILP	\sum ILPs	% on ILPs	ILPs	Instances
SIM25	11.74 h	0.21 h	12.03 s	10.01 h	85.2%	2995.5	2199.2
SIM15	7.44 h	0.13 h	12.01 s	6.33 h	85.1%	1897.9	1397.6

Table 6.12: Comparison of the computational time of the simulation models. The amount of instances and ILPs correspond to the average number of leaves in the tree structure of the algorithm, where each leaf is again divided into one or more subproblem(s), as explained in Section 5.4.4. In each subproblem an ILP needs to be solved.

To put the running time of the simulation models into perspective, we should realize that the running time per multi-period time window (MPTW) refers to the transportation of containers distributed over the entire network. In actual applications, we are only interested in the running time per single-period time window (SPTW), where a decision needs to be made for the upcoming hours.

Concerning the SIM25 model, almost all SPTWs can be solved within 10 minutes, except for some outliers. In each instance, it might occur once or twice that the decision space is quite large, implying that the number of ILPs that need to be solved increases significantly.

For example, during the experiments, the decision space contained once a maximum of 20 potential decisions. It took 58 minutes to find the best decision, which is too time consuming. Although 15 decisions were discarded after the first phase (i.e., after the check that is performed after 11 simulations), the computational time of the first phase was 44 minutes. In this specific example, the five remaining contenders had non-zero frequency already after three simulations. When the check was performed right then, the computational time of the decision moment had shrunk to 34 minutes, where the first phase took only 12 minutes.

Therefore, it might be a good idea to improve the first phase if the decision space is large. This could be done by performing the check at an earlier stage, by discarding decisions after one or two simulations when the objective value deviates too much from the others or by taking into account the symmetry of some decisions even more.

Additionally, because of the tree structure of the algorithm, the simulation model can be parallelized easily. Around 85% of the computational time is spent on solving ILPs, implying that the computational time on a PRAM computer [2] with infinite many processors and zero communication cost is less than a minute. Although it is unrealistic, it shows that the average running time per SPTW can be reduced drastically.

As can be seen in Table 6.11 the simpler models are advantageous in terms of running time. Solving a SPTW takes only 48 seconds, because only one *normal ILP* has to be solved. Finally, using Table 6.12, the *heuristic ILP* turns out to be almost four times faster in terms of running time. Since the extra amount of ILPs that need to be solved is increased only by a factor 1.36, the algorithm does benefit considerably. Without using the heuristic, running the SIM25 and SIM15 model would take around 28.85 and 18.34 hours, respectively, implying that the algorithm has been speed up by a factor of approximately 2.25.

6.2.3 LP relaxation

The majority of the time, the algorithm is busy solving ILPs. In order to drastically decrease the computational time, we might benefit from changing the ILP into the LP relaxation. At first sight this might be possible because the (potential) decision of each barge (i.e., the route within the CTW) is fixed before actual solving the program. Although, notice that the fixed decision disregards the container assignment at the origin and the container terminal. The assignment at the origin does not occur any problems, because it is done manually. However, the container-to-mode assignment at the container terminal raises problems. If multiple delivery appointments are scheduled at the same time at different terminals (i.e., *conflicting appointments*), the problem might become infeasible if containers corresponding to conflicting appointments are loaded on the same barge. Since in the LP relaxation it is allowed to set design variables y_e equal to $1/2$, the barge can use half of its capacity to visit one appointment and half of its capacity to visit another conflicting appointment at the same time. Therefore, at first sight decisions might seem beneficial, but will eventually lead to infeasibility.

6.2.4 Extending the amount of (uncertain) data

This section is dedicated to the fourth sub-question formulated in Section 1.3. In practice, the barge planners request for an appointment only 12 time steps in advance, since they experienced that it does not benefit to call at an earlier stage. However, it might be worth knowing what will happen if the assumption is relaxed, i.e., when orders are announced at an earlier stage. Naturally, more deterministic information will lead to better performance, but what does happen with the performance if more information, even stochastic, is available?

Only experiments for the simpler models have been carried out. Due to the construction of the simulation algorithm, it was not possible to run experiments. The ILP heuristic is based on the length of the SPTW assuming that orders are announced 12 time steps in advance. By announcing orders at an earlier stage, the simulation model can only be solved using normal ILPs, which would probably take several days.

In Tables 6.13, 6.14 and 6.15 the results are presented for the RC, EC and AC models, respectively, as the difference in results of the B12 model and the simpler models. The announcement time relative to the requested appointment times in parenthesis.

	<i>Cost diff. (€)</i>		<i>Trucks diff.</i>		<i>Short trips</i>		<i>Long trips</i>		ϵ
	Mean	S.D.	Mean	S.D.	Out	In	Out	In	
RC (12)	3102.3	2000.2	16.1	10.7	0.1	2.0	5.3	3.6	1.3
RC (16)	2215.9	2410.5	10.6	12.4	-3.4	-2.0	7.1	5.3	0.9
RC (20)	2238.6	2731.6	11.1	14.3	-2.5	0.0	6.5	3.8	0.8
RC (24)	1965.9	2061.0	9.8	10.4	0.0	-1.9	5.1	3.0	0.9

Table 6.13: Shifting the announcement time for the RC model

	<i>Cost diff. (€)</i>		<i>Trucks diff.</i>		<i>Short trips</i>		<i>Long trips</i>		ϵ
	Mean	S.D.	Mean	S.D.	Out	In	Out	In	
EC (12)	3815.9	1334.6	19.2	7.4	0.5	-0.2	13.1	4.7	0.3
EC (16)	2002.3	1838.4	10.3	10.1	1.5	-0.1	6.6	0.8	0.4
EC (20)	2904.5	1189.5	13.9	6.1	-2.8	-2.1	13.6	5.2	0.0
EC (24)	3059.1	1434.5	14.8	7.3	-0.8	-3.0	13.7	4.9	0.0

Table 6.14: Shifting the announcement time for the EC model

	<i>Cost diff. (€)</i>		<i>Trucks diff.</i>		<i>Short trips</i>		<i>Long trips</i>		ϵ
	Mean	S.D.	Mean	S.D.	Out	In	Out	In	
AC (12)	2134.1	1744.2	11.3	9.3	3.6	0.5	4.7	1.0	0.4
AC (16)	1815.9	1940.0	9.3	10.8	0.6	0.2	5.6	1.4	0.4
AC (20)	1884.1	1992.3	9.5	10.2	2.3	-2.0	5.1	2.7	0.4
AC (24)	1163.6	1991.1	5.5	11.0	-1.1	-3.1	5.0	2.1	0.6

Table 6.15: Shifting the announcement time for the AC model

Although on average better performance is obtained in the RC and AC models, the EC models fluctuates quite a lot. Furthermore, in terms of *robustness* the models do not perform satisfactory, e.g., for some instances the RC (12) model even exceeds the RC (24) model (and similarly for the AC models). To clarify, the results of a specific instance are shown in Table 6.16.

	Cost (€)
RC (12)	13075.0
RC (16)	12350.0
RC (20)	10625.0
RC (24)	14050.0

Table 6.16: Results corresponding to a specific instance

We may conclude that more information, even stochastic, leads on average to better performance for the RC and AC models, but it is certainly not guaranteed. Just like in Example 6.2.1, in the simpler models there is a certain risk factor. Most of the time (luckily) no problems occur, but when it does, performance will collapse drastically.

6.3 SUMMARY

In this section, the most important findings from the numerical experiments will be summarized, and where necessary, more insight into the results will be provided by closely inspecting the solutions.

The offline results showed that the lower bound can be approached by disregarding one part of the uncertainty and shifting the announcement time of the orders to an earlier stage. Although there is quite some uncertainty left in the B28 model, the optimality gap was only 1.06%, implying convergence. What stood out was the sudden change in the offline results (i.e., B72 model) in the use of trucks for long trips corresponding to outland orders. We expected beforehand that these type of trips would decrease even further, but the offline results showed the opposite.

In the online results the assumption about the focus on inland orders was validated. Relatively, the proposed solution methods did even use more barges for short trips corresponding to inland orders. The more uncertainty is present, the more the focus is shifted to that type of trips. Apparently, including the uncertainty element(s), the pick-up of inland orders at deep-sea terminals is an easy way to minimize the costs at each single-period time window, i.e., *locally*. It makes perfect sense, because visiting a pick-up appointment by barge can be decided last minute (only sufficient capacity is required).

Furthermore, the simulation model SIM25 (in which 25 simulations were performed each decision moment) surpasses the simpler models both in terms of outcomes and robustness. In the simpler models infeasibility incurred due to poor container assignment, while in the simulation model no penalties were given at all. Besides that, with the exception of one instance, the results of the cost function for the simulation model were within 20% of the B12 model. Moreover, the majority of the SPTWs can be solved within 10 minutes. Only the exceptional decision moments where the number of potential decisions is huge (i.e., large decision space), the computational time exceeds the time limit for applicability. Some recommendations were given in order to deal with such exceptions.

A sensitivity analysis on the number of simulations suggested that 15 simulations should be sufficient to obtain an adequate decision that is *resistant to change*, i.e., feasible and (sub)optimal for every potential future scenario that has been simulated. However, the analysis disregarded the container-to-mode assignment at the origin and the container terminal, that are based on the outcomes of the simulations. This was reflected in the results, that showed that the performance in the long trips decreased. On the other hand, a reduction in the number of simulations benefits in terms of the computational time. The average running time per SPTW dropped down from 12.81 to 8.12 minutes. Therefore, a reduction in the number of simulations can be done if quick results are needed, but it has to be taken into account that the quality of the solution is weakened.

In addition, by shifting the announcement time to an earlier stage, better results were obtained for the RC and AC models in terms of average cost and average number of trucks used. However, a downside of the models is the fluctuation within the results. More information, even stochastic, leads on average to better performance, but it is certainly not guaranteed. The EC model even showed that on average the results can fluctuate too.

7 | CONCLUSION

This chapter provides complete answers to the research questions formulated in Section 1.3. Thereafter, some recommendations for future research concerning simulation techniques applied to synchromodality are given.

7.1 CONCLUSIONS

The main goal of the thesis was to seek out the answer to the following research question.

“What kind of solution methods can be used for the single-period time windows, including uncertainty, to find an appropriate schedule and container assignment for the entire multi-period time window (within an acceptable amount of time)?”

In order to answer the question, five different sub-questions were formulated, which will be completely answered below.

1. *How can the base instance described in Section 1.2 be modeled in order to meet all the assumptions made?*

An online optimization approach is proposed, where the input data come in sequentially and decisions have to be taken while part of the relevant information is still uncertain or unknown. At each decision moment, the uncertainty element in the requested appointment times is converted to an offline optimization problem by disregarding the uncertainty or by simulating various potential future scenarios. Subsequently, the offline problem is modeled as a multi-commodity network design problem on a time-space graph, that minimizes the costs incurred by the use of trucks on each single-period time window.

2. *What solution methods can be used to obtain both a schedule and container assignment for every transportation mode in the network?*

Four different solution methods are proposed: the RC, EC, AC and simulation models. The RC, EC and AC models are the simpler ones, in which the uncertainty is partially disregarded. The models assume that the requested appointment times will be confirmed at the requested, the earliest possible and the average appointment time, respectively. Alternatively to the simpler models, a much more complex algorithm is developed in which future scenarios are simulated for the requested appointment times given their probability vector p . The simulation model seeks the best decision(s) that is resistant to change, i.e., feasible and (sub)optimal for every potential future scenarios that has been generated.

3. *What can be said about the quality and practical relevance of the results obtained by the solution methods?*

In order to analyse the performance, experiments were carried out for 11 randomly generated instances. To say something about the quality and the practical relevance, the results were presented as the difference in results of the B12

model and the solution methods. Although the B12 model does not guarantee optimality, the outcome can still be used as a benchmark for the problem. With the exception of one instance, the results of the cost function for the simulation model were within 20% of the B12 model. The gap was 12.5% on average, where some instances performed even better for the simulation model than the B12 model (containing less uncertainty). Although not optimal, the simulation model provides a reliable vehicle routing and a container-to-mode assignment. Within the decision making process, the model finds the best decision(s) that is *resistant to change*.

The practical relevance of the simulation model is restricted in the sense that the model is build on several assumptions. Every order is announced exactly 36 hours in advance, and confirmed exactly 12 hours later. In practice, the announcement and confirmation times are more scattered. Furthermore, the assumption is made that the confirmed appointment can only be scheduled within an uncertainty interval of length seven, where the confirmation is based on the probability vector p that is uniformly distributed. Finally, due to the lack of real-world data, no adequate comparison can be made between the decisions made by the simulation algorithm and the decisions that barge planners would make in practice.

4. *What can be said about results obtained by the solution methods if the assumption on the announcement time of orders is relaxed, i.e., when orders are announced at an earlier stage?*

In practice, the barge planners request for an appointment only 12 time steps in advance, since they experienced that it does not benefit to call at an earlier stage. If the assumption is relaxed, more information, even stochastic, is available at a decision moment. Due to construction of the simulation model, it was only possible to do experiments for the simpler models. On average better results were obtained for the RC and AC models in terms of average cost and average number of trucks. However, a downside of the relaxed models is the fluctuation within the results, implying that some instances benefit, but a considerable number of instances did not benefit or even had worse results. We may conclude that more information, even stochastic, leads on average to better performance for the RC and AC models, but the fluctuation affects the robustness of the models a lot. The EC model even showed that on average the results can fluctuate too.

5. *Does the chosen approach successfully incorporate elements from synchronomodality?*

The online optimization approach that is proposed, definitely incorporates elements from synchronomodality. First of all, the approach incorporates unknown and uncertain network elements that appear in the requested pick-up and delivery appointment times. Furthermore, the proposed solution methods involve the transportation of freight in containers using multiple modes of transport, that can be freely chosen by the barge planner. Finally, the barge planner has the ability to respond to unexpected disturbances. For example, if an order is confirmed at a much later time, the barge planner can decide to unload the corresponding container(s) at the container terminal. It is possible to re-evaluate the plan at any moment. Not a single route of an order is known or fixed beforehand.

7.2 RECOMMENDATIONS

To conclude this research, we discuss some further research directions that may be developed and possibly lead to future success or usefulness.

First of all, the network under consideration can be extended to a more practice-oriented model as shown in Figure 7.1. The extended network takes into account the further away deep-sea terminals located in the Maasvlakte I and II. The blue node in between can be seen as a waiting node on the water located in between the two terminal clusters.

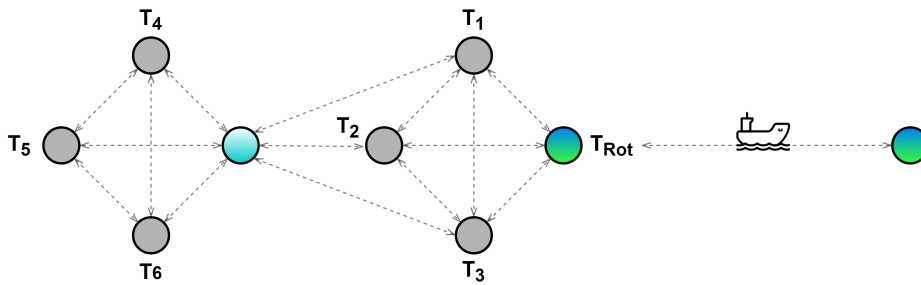


Figure 7.1: The extended network taking into account the Maasvlakte.

Besides that, the model can be generalized in terms of the number of barges and orders or the size of the orders. Even more general, the simulation based model can be applied to any multi-commodity network design problem including uncertainty (based on probability distributions) using the idea of solving *explicitly* a vehicle routing problem and *implicitly* a container-to-mode assignment.

As mentioned before, when the size of the decision space increases, the computational time of a single-period time window grows significantly. Reducing the number of simulations affects the quality of the outcome, but improving the first phase (i.e., the phase before the check after 11 simulations) might be a good idea. This could be done by performing the check at an earlier stage, by discarding decisions after one or two simulations in case the objective value deviates too much from the others, or by taking into account the symmetry of some decisions even more.

Furthermore, a better comparison can be made if the probability vector p becomes more practice-oriented. The most ideal would be when the probability distribution can be extracted from the real-world data. If not, a sensitivity analysis could be carried out for different kinds of probability distributions.

Finally, because of the tree structure of the algorithm, the simulation model can be parallelized easily. By doing so, the algorithm should benefit in terms of computational time.

A

PSEUDO CODE

In Section 5.4 the vectors ρ_{cap} and ρ_{loss} have been defined. The pseudo code of the algorithm to construct the elements $\rho_{cap}(t)$ and $\rho_{loss}(t)$, for some barge $w \in W$ having initial availability at time step $t_{w,0}$, is stated in Algorithm A.1.

Algorithm A.1: Determination of $\rho_{cap}(t)$ and $\rho_{loss}(t)$

```
for  $t = t_{w,0}$  to  $t_a + 12$  do
  for  $k \in K_{Origin}^{out}$  do
    if  $t$  in  $\{\tau_{pu}(k), \dots, \tau_{del}(k) - 8\}$  and  $k$  fits on board then
      if no appointment at time step  $\tau_{del}(k)$  then
        Add  $|k|$  to  $\rho_{cap}(t)$  and set  $\rho_{cap}(\min(\tau_{del}(k) - 8, t_a + 12))$  to 1
      else if appointment at  $\chi_{del}(k)$  at time step  $\tau_{del}(k)$  then
        Add  $|k|$  to  $\rho_{cap}(t)$ 
      else if appointment at different terminal at time step  $\tau_{del}(k)$  then
        "Check if it is possible to unload.."
        for  $t_{check}$  in range( $t + 8, \tau_{del}(k) - 1$ ) do
          if if no appointment at time  $t_{check}$  and  $t_{check} - 1$  then
            Add  $|k|$  to  $\rho_{cap}(t)$  and break current for loop
          end if
        end for
      end if
    end if
  end for
end for
```

BIBLIOGRAPHY

- [1] Bektaş, T. and Crainic, T. G. (2007). A brief overview of intermodal transportation. *CIRRELT*, 03.
- [2] Chatterjee, S. and Prins, J. (2005). COMP 203: Parallel and distributed computing; pram algorithms. *Course Notes*.
- [3] Chiscop, I. (2018). A robust optimization approach to synchromodal container transportation. *Master's Thesis*.
- [4] Chouman, M. and Crainic, T. G. (2012). MIP-based Matheuristic for Service Network Design with Design-Balanced Requirements. *CIRRELT*, 79.
- [5] Cook, W. J., Cunningham, W. H., Pulleyblank, W. R., and Schrijver, A. (1998). *Combinatorial Optimization*. Wiley-Interscience, Netherlands.
- [6] Cordeau, J.-F., Laporte, G., Savelsbergh, M. W., and Vigo, D. (2007). Vehicle Routing. *Handbooks in Operations Research and Management Science*, 14:367–428.
- [7] Crainic, T. G. (2006). Intermodal Transportation. *Transportation*, 14:467–537.
- [8] Crainic, T. G. and Gendreau, M. (2002). Cooperative Parallel Tabu Search for Capacitated Network Design. *Journal of Heuristics*, 8:601–627.
- [9] Crainic, T. G. and Laporte, G. (1997). Planning models for freight transportation. *European Journal of Operational Research*, 97:409–438.
- [10] ETP-Alice (2018). Roadmap: Corridors, Hubs and Synchromodality. <http://etp-logistics.eu>.
- [11] Ficken, F. A. (2015). *The Simplex Method of Linear Programming*. Dover Publications, English.
- [12] Fragkos, I., Cordeau, J.-F., and Jans, R. (2017). The multi-period multi-commodity network design problem. *CIRRELT*, 63.
- [13] Han, J., Lee, C., and Park, S. (2013). A Robust Scenario Approach for the Vehicle Routing Problem with Uncertain Travel Times. *Transportation Science*, 48:313–463.
- [14] Hendriks, M. (2009). Multi-step optimization of logistics networks: Strategic, tactical and operational decisions. *PhD Dissertation*.
- [15] Häme, L. and Hakula, H. (2013). Dynamic journeying under uncertainty. *European Journal of Operational Research*, 225:455–471.
- [16] Huizing, D. (2017). General methods for synchromodal planning of freight containers and transports. *Master's Thesis*.
- [17] Johnson, D., Lenstra, J., and Kan, A. R. (1978). The Complexity of the Network Design Problem. *Networks*, 8:279–285.
- [18] Juncker, M., Huizing, D., Vecchy, M., Phillipson, F., and Sangers, A. (2017). Framework of synchromodal transportation problems. *Lecture Notes in Computer Science*, pages 383–403.
- [19] Kalicharan, K. (2018). Intermodal transport: Routing vehicles and scheduling containers. *Master's Thesis*.

- [20] Karp, R. (1972). Reducibility among Combinatorial Problems. *Complexity of Computer Computations*, pages 85–103.
- [21] Karp, R. (1975). On the Computational Complexity of Combinatorial Problems. *Networks*, 5:45–68.
- [22] Konings, R. (2007). Opportunities to improve container barge handling in the Port of Rotterdam from a transport network perspective. *Journal of Transport Geography*, 15:443–454.
- [23] Kooiman, K., Phillipson, F., and Sangers, A. (2016). Planning Inland Container Shipping: a Stochastic Assignment Problem. *23rd International Conference on Analytical and Stochastic Modelling Techniques and Applications*.
- [24] Marr, B. (2019). 7 amazing examples of digital twin technology in practice. *Forbes Innovation*.
- [25] Montreuil, B. (2011). Towards a Physical Internet: Meeting the Global Logistics Sustainability Grand Challenge. *Logistics Research*, 3:71–87.
- [26] Pfoser, S. (2016). Critical success factors of synchromodality: Results from a case study and literature review. *Transportation Research Procedia*, 14:1463–1471.
- [27] Reis, V. (2015). Should we keep on renaming a +35-year-old baby? *Journal of Transport Geography*, 46:173–179.
- [28] Rivera, A. E. P. and Mes, M. R. K. (2016). *Service and transfer selection for freights in a synchromodal network*. TU Eindhoven, Research School for Operations Management and Logistics (BETA), Netherlands.
- [29] Rivera, A. E. P. and Mes, M. R. K. (2017). Anticipatory freight selection in intermodal long-haul round-trips. *Transportation Research Part E*, 105:176–194.
- [30] Saddik, A. E. (2018). Digital Twins: The convergence of multimedia technologies. *IEEE MultiMedia*, 25:87–92.
- [31] Schmidt, G. and Wilhelm, W. E. (2000). Strategic, tactical and operational decisions in multi-national logistics networks: A review and discussion of modeling issues. *International Journal of Production Research*, 38:1501–1523.
- [32] UNCTAD (2018). *Review of Maritime Transport 2018*. United Nations publication, New York and Geneva.
- [33] Vecchy, M. and Huizing, D. (2016). The current worldwide state of synchromodal transport in academics and practice. *TNO internal documentation*.
- [34] Yaghinia, M. and Akhavan, R. (2012). Multicommodity network design problem in rail freight transportation planning. *Procedia – Social and Behavioral Sciences*, 43:728–739.

