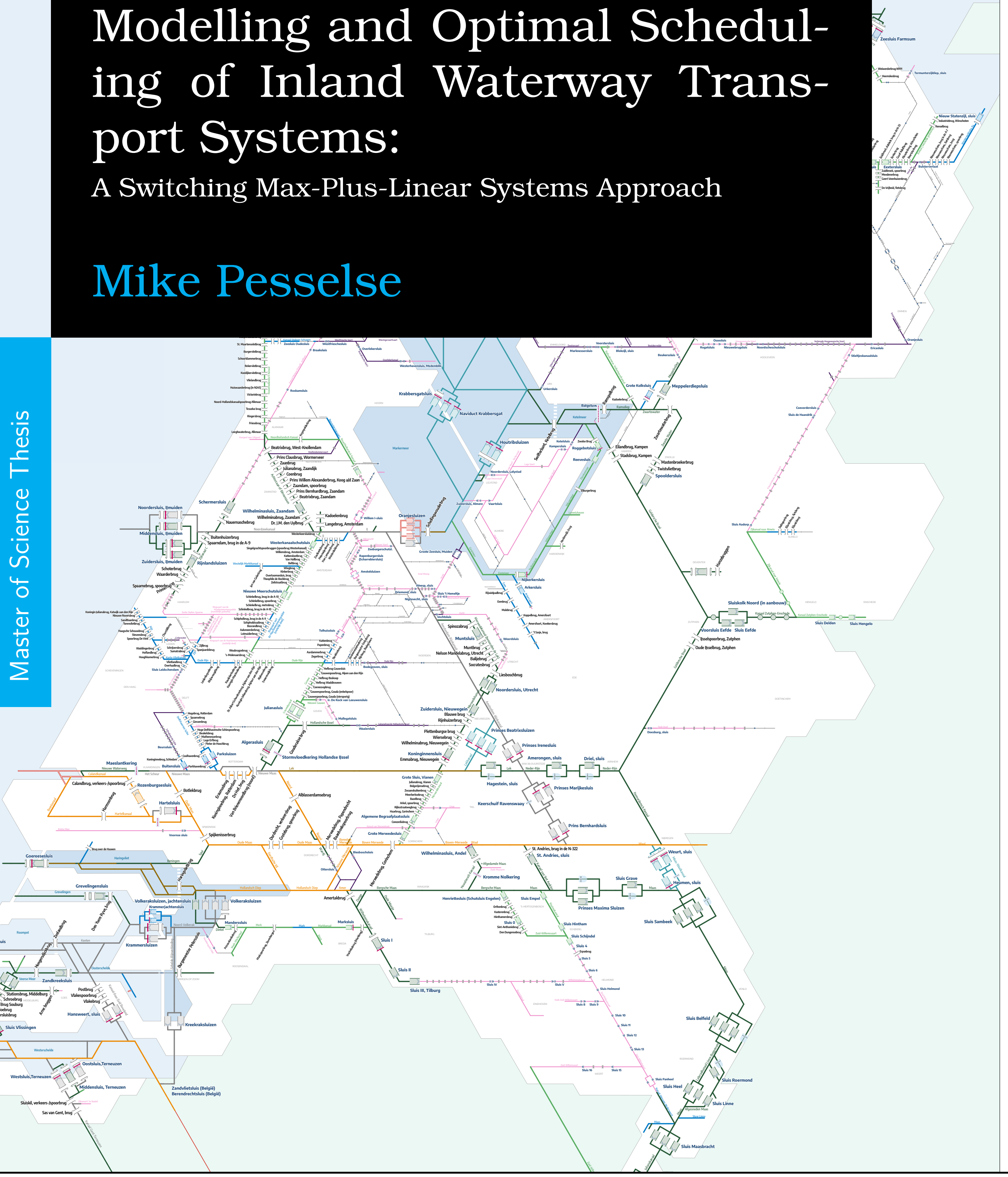
Rijkswaterstaat
Ministerie van Infrastructuur en Waterstaat

Modelling and Optimal Scheduling of Inland Waterway Transport Systems:

A Switching Max-Plus-Linear Systems Approach

Mike Pesselse

Master of Science Thesis



TU Delft
Delft University of Technology

Delft Center for Systems and Control

Modelling and Optimal Scheduling of Inland Waterway Transport Systems:
A Switching Max-Plus-Linear Systems Approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Mike Pesselse

February 19, 2022

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was done in close collaboration with the department of Maritime and Transport Technology and NOVIMOVE partners. Their cooperation is hereby gratefully acknowledged.

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Modelling and Optimal Scheduling of Inland Waterway Transport Systems:

by

Mike Pesselse

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: February 19, 2022

Supervisor(s): ______________________
Dr.ir. A.J.J. van den Boom

____
Dr.ir. V. Reppa

____
Dr.ir. P. Segovia Castillo

Reader(s): ______________________
Dr.ir. A. Dabiri

Abstract

Inland waterways form a natural network infrastructure with the capacity for waterborne transport of people and goods for moving freight from seaports to the hinterland. Recently, Inland Waterway Transport (IWT) has been promoted more extensively by the European Union and various governments as it plays a crucial role in reducing road congestion and CO2 emissions from transport. However, the advantages of IWT are not fully exploited due to inefficiencies in the logistics system, such as long waiting times at locks and sub-optimal navigation on waterways. Currently, no scheduling at infrastructures or routing optimisation of the overall waterway network is happening. The scheduling of vessels through a lock is usually performed on a First In First Out basis, providing an opportunity for improvement. Hence, this thesis aims to design a scheduling strategy for generating an optimal plan for sending inland vessels through a waterway network with minimal delays, yielding a significant positive impact on the modal shift towards IWT.

A promising approach to scheduling problems is by using Switching Max-Plus-Linear (SMPL) systems. SMPL systems have proven to be effective in various Discrete-Event Systems and transportation networks. Using SMPL models is convenient since non-linear scheduling problems can be described linearly using Max-Plus operators without compromising on the system dynamics. Moreover, as the SMPL systems can be transformed into Mixed-Integer-Linear-Programming (MILP) problems, it is also possible to use fast optimisers for solving the scheduling problems.

This thesis will show how one can describe IWT systems, consisting of; waterways, vessels and locks, as SMPL systems. The optimal schedule for the inland vessels is determined based on multiple input parameters, including waterway network lay-out, the sailing speeds of vessels and arrival deadlines of the vessels. The scheduler will return the individual vessel routing and overall vessel order in the waterway network. This routing and order selection is defined using binary control variables, turning the IWT scheduling problem into a MILP problem, which will allow finding the solution to large scale IWT scheduling problems in a reasonable computation time. Furthermore, this thesis will show how the goal of minimising the cumulative arrival times of all vessels in a network can be achieved. This is done for different types of waterway network cases, for which the results are shown and analysed.

Table of Contents

Preface & Acknowledgements

Dear reader,

This thesis marks the last stage of my study time at the Delft University of Technology; it has been a challenging final year to an almost eight year journey. I am incredibly grateful for the time, and I would like to express that with a few words.

First of all, I would like to thank my supervisors Dr.ir. A.J.J. van den Boom, Dr.ir. V. Reppa, and Dr.ir. P. Segovia Castillo, for their guidance and thorough support during the past year. Which I can imagine not always being easy, in these unusual pandemic times.
Ton, I really appreciate the opportunity you gave me to research the more practical side of the Max-Plus and combine it with my interest in the maritime industry. Your enthusiasm for inland vessels has always motivated me enormously. I enjoyed our conversations where you could tell me exactly at which stretches of the Scheldt-Rhine Canal you expected inland vessels to overtake. I also appreciate how you took the time and had the patience to explain Max-Plus algebra to me.
Vasso, I can clearly remember when I reached out to you for the first time just over a year ago. After which we planned our first online meeting where you explained several thesis topics on inland vessels. I do not think you had this direction in mind back then, however, I am very grateful for the opportunity you gave me and to do the thesis work under your supervision. Thank you for your trust and confidence in exploring this Max-Plus and inland waterway transport combination. I hope that I have succeeded a little in making it clear. Your supervision helped me a lot, especially to keep myself on track in the final months of the thesis.
Pablo, I have always felt that you were genuinely interested in understanding the field of Max-Plus and were open to using it. This was motivating and kept me going in improving my understanding, so I could explain it better. Moreover, you often helped me zoom out and look at the bigger picture. Thanks a lot for your availability and the motivating words during the more challenging parts of the thesis. Also, I would like to thank Dr.ir. A. Dabiri, for her effort in reading and evaluating my thesis.

Moreover, a massive thank you to my parents and sister for their guidance and full support in every decision I made in the past years. I could not have accomplished this without you. Another shout out to my friends, ranging from; student associations, DreamTeams, housemates, exchanges and summer projects. Thanks for always being there during; fun, crazy, and difficult times. Although it is time to move on, it would be an easy decision if I could turn back the time. You have made my time in Delft unforgettable!

At last, I would like to dedicate this thesis to my grandpa, who sadly passed away during the first few months of starting the master program. Who has been a great inspiration and showed me anything is possible with hard work. I know he would have been immensely proud of me achieving my Engineering degree by writing a thesis on his beloved maritime industry. I like to believe he notices.

Delft, University of Technology — Mike Pesselse
February 19, 2022

"Hou de bal in de ploeg"
— *W.Chr. Pesselse*

Chapter 1

Introduction

Inland Waterway Transport (IWT) plays a crucial role in the reduction of road congestion and the reduction of $CO2$ transport emissions. Unfortunately, the advantages of IWT are not fully exploited due to inefficiencies in the system, such as long waiting times at locks and sub-optimal navigation on waterways. In this thesis *'Modelling and Optimal Scheduling of Inland Waterway Transport Systems: A Switching Max-Plus-Linear systems approach'*, a method for designing a scheduling strategy that allows inland waterway vessels to sail through a waterway network in an optimal way with minimal delays, was studied.

Chapter 1 will start with an extensive introduction to the research topic and is structured as follows: First, in order to grasp the problem, some background information on the research topic is given in Section 1-1. Next, the relevance of the thesis on an academic and applied level is described in Section 1-2. Thereafter, in Section 1-3, the problem description and scope are presented, which includes the research questions and research approach. At last, the outline and structure of the thesis are given in Section 1-4.

1-1 Background information

On Tuesday morning on 23 March 2021, one of the largest container vessels in the world, the MS Ever Given, got stuck in the Suez Canal while on its way to the Port of Rotterdam [2]. The vessel completely blocked the canal for over six days, resulting in a traffic jam of over four hundred vessels. With over 12% of global trade passing through the Suez Canal, the event had a significant impact on the global economy. The clearing of the canal resulted in an unprecedented high number of arriving vessels in ports, which caused massive congestion and delays at terminals and on inland transportation routes. This event highlights the vital importance and especially the fragility of the maritime logistics system.
As a result of globalisation in the past century, consumer and producer distance has increased enormously. This was mainly made possible by advancements in transportation methods and economies of scale. Enormous container vessels are now voyaging the oceans, delivering goods

and bulk from port to port across the globe. These vessels cannot reach inland destinations due to their size, and thus goods and bulk have to be transferred to various inland transportation methods (e.g. trucks, trains, inland vessels) for hinterland transportation. These three transportation methods are usually referred to as modalities that each have their advantages, disadvantages and modal share of the complete transport to the hinterland. Figure 1-1 shows an example of this transportation chain starting from the seaport on the left towards the inland destination on the right. The middle transportation mode could also be one of the other modalities or a combination of the three.

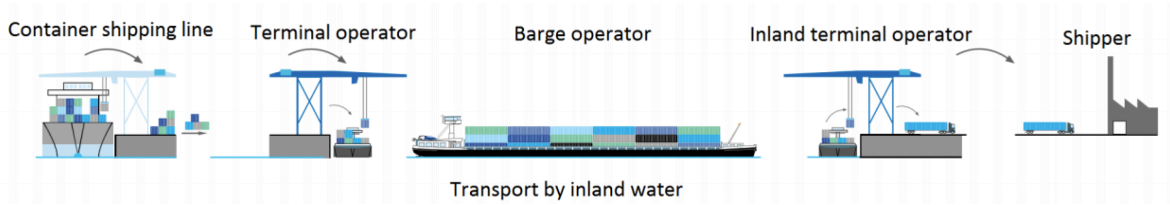


Figure 1-1: Inland container transport using inland shipping (i.e. hinterland vessel transport) [15]

Institutions like the European Commission [19, 23] and the Port of Rotterdam [38] are heavily promoting a modal shift towards IWT in order to make better use of this natural network of inland waterways. The main reasons for this shift are; cost efficiency [53], environmental friendliness in terms of goods transported [15, 37], high degree of safety [20], and reduction of road congestion [18]. The Port of Rotterdam has set the goals to increase the modal share percentage and absolute volume of IWT. Table 1-1 shows the goal for 2035 and progress of reaching these goals measured in 2013.

Table 1-1: Port of Rotterdam inland container transport modal share progress overview defined in 2013, including goals for 2020 and 2035 [39]

Modalities	2009	2013	2014	2020	2035
Inland shipping	39%	35%	36%	41%	45%
Rail transport	13%	11%	11%	17%	20%
Truck transport	48%	54%	53%	42%	35%

Next to the Netherlands and the European Union, also, the United States of America has seen its inland waterborne transportation increase from 1500 million tons in 1970 to 2600 million tons in 2006 [45]. The cargo volume passing through the Panama Canal has been growing at 3% annually since 2005 and is expected to continue to grow [35]. As a result, a potential expansion of the New York Canal System has been thoroughly researched [4]. In China, the freight moved on the Yangtze River has increased more than expected, resulting in the famous Three Gorges lock already reaching its maximum capacity of 100 million tons in 2011 [33]. This maximum capacity was reached 19 years earlier than predicted and has currently made the Three Gorges lock a bottleneck on the Yangtze River, limiting further IWT development. This last point about the Three Gorges ship lock becoming a bottleneck on the Yangtze River is something that the European inland waterways, Panama Canal and New York Canal System are slowly running into as well due to their increased interest for

IWT.
This problem can be solved by increasing these locks' capacity; however, this has high costs, and expansion is not always possible due to environmental or spatial reasons. Another solution could be optimally scheduling these locks' operation to ensure efficient passing of vessels. This scheduling could have a significant advantage in moving freight from seaports to the hinterland without the high investment costs of building additional infrastructure, which will be the focus of this thesis. Due to its conservative nature, the inland shipping industry has always been somewhat slow with embracing new technologies; as a result, these recent scheduling developments have not found a way into inland shipping yet.

However, an increasing interest in better understanding complex man-made systems has led to automation, optimisation, and digitisation developments that rapidly improve human lives across many industries and sectors.These complex systems can generally be described in either continuous-time models or discrete event models.
Conventional system theory involves modelling the behaviour of a system over continuous time. The evolution of these continuous-time systems is governed partially or solely by the progress of time itself. However, it might be more convenient to model the evolution of a system over a set of discrete events in some cases. Consider a baggage handling system, where a state of the system represents the number of luggage items in a storage room that are waiting to be moved. The evolution of the number of luggage items in the storage room is only affected by the incoming and outgoing of individual luggage pieces. When a dynamic model like the baggage handling system is only involved with a series of such events and not the dynamics between them, the system is called a Discrete Event System (DES).
Modelling, analysing, and controlling of complex man-made DESs has attracted more and more interest for applications such as; railway traffic management[25, 28], urban rail transit networks [59], printer systems [1], manufacturing systems [51], legged locomotion [34] and other various DESs systems [16].
Most of the mathematical models that describe the behaviour of DESs are nonlinear in conventional algebra. There is; however, a class of DESs that can be described by a model that is linear by using the so-called Max-Plus algebra. Modelling DESs using Max-Plus algebra has the advantage that the resulting system can be considered as a Max-Plus Linear (MPL) system which has a solid analogy to conventional linear system theory. Moreover, these MPL systems can be transformed to Mixed Integer Linear Programming (MILP) problem which allow for optimal scheduling. Therefore, describing DESs using Max-Plus algebra is promising. Describing the IWT system using this Max-Plus algebra will be the main focus of this thesis. After having given the background information to the problem, Section 1-2 will highlight the relevance of the research.

1-2 Relevance of research

The next section will address the relevance of the research, and will list where value on an applied and theoretical level will be added.
As mentioned, governmental institutions are promoting a modal shift towards IWT because of; cost efficiency [53], CO2 reduction [15, 37], high degree of safety [20], and the reduction of road congestion [18]. However, as a result of inefficiencies in the system, caused by long waiting times at locks and sub-optimal navigation on inland waterways, these benefits of

IWT are not fully taken advantage of. The scheduling strategy researched in this thesis will contribute to solving these inefficiencies, by generating an optimal schedule with minimal delays. Multiple European projects have recently started addressing these problems, for instance, the NOVIMOVE (Novel inland waterway transport concepts for moving freight effectively) project is aimed at solving inefficiencies on the IWT Rhine-Alpine corridor. The project is funded by the Horizon H2020 Programme of the European Union and consists of a collaboration between 4 EU member states and two associate countries. NOVIMOVE wants to improve the IWT system by condensing it. This should improve container load factors and reduce delays. This condensing will be done by improving voyage routing and ensuring effective scheduling through bridges and locks. Additionally, innovations such as an intelligent waterway navigation system and intelligent vessel and infrastructure interactions will be developed. The goal is to increase the quantity of freight moved using IWT along the Rhine-Alpine corridor by 30% with respect to 2010 data. The scheduling strategy researched in this thesis could provide a basis for further developing these vessel and infrastructure interactions.
Besides European projects, there are currently exist tools for managing IWT, for instance, VisuRIS is a tool built for the management of the Flemish waterways, which aims to help inland vessels with real-time information for planning trips and traffic reports. VisuRIS presents waterway information on a practical map, including operating times of bridges and locks and the expected the expected vessel traffic at infrastructures structures. Most IWT management tools seem to provide information only and lack a global network optimisation. The scheduling strategy researched in this thesis could provide a basis for improving these tools.
In summary; the modal shift to IWT, contribution to European IWT development projects and contribution to IWT management tools, means that research on possible scheduling strategies is required. In this thesis, we will address a promising scheduling strategy by considering the IWT as a DES and modelling it as a Switching Max-Plus Linear (SMPL) system; therefore, this research is relevant at an applied level.

Next, the relevance of the thesis on a theoretical level is to further develop the research on scheduling strategies, in particular, scheduling using SMPL systems. First of all, only a limited number of DESs have been described using the SMPL framework. Therefore, modelling the unique properties and characteristics of the IWT system will help with improving this modelling process, which will be useful for modelling new systems in the future. Secondly, the IWT system could be generalised to a production system in which products arrive from opposite directions and compete for the same resource. This interaction will be described for the first time in research to the best knowledge of the author. Next, currently available Max-Plus research is mainly applied to systems with a maximum of three job families (i.e. types of jobs). In this thesis, the vessels will be considered as jobs, and as all vessels have their own characteristics (e.g. variable speed due to loading), there will be a large number of possible jobs. At last, to the best knowledge of the author, until now, no research was performed on Max-Plus systems where jobs are allowed to overtake each other, which will be dealt with in this thesis and thus could be used in the modelling of future systems. In summary, the novel IWT SMPL system will improve the field of scheduling using Max-plus algebra, and therefore, this research is also relevant on a theoretical level.
After having introduced the relevance of the research, the next section will describe the problem that is tackled in the thesis.

1-3 Problem description

This section will outline the problem of the thesis by describing the research objective, research scope, research questions and research approach.

1-3-1 Research objective and scope

As mentioned before, with this thesis, we want to contribute to solving the inefficiencies in IWT systems by creating an effective vessel schedule. For a schedule to be effective, it is crucial that the actors (i.e. vessels and locks) involved in the system cooperate and behave according to the created schedule. This is not the case for the current IWT systems as these consist of non-autonomous individual vessel skippers who all have their own interests and are highly unlikely to cooperate without opposing. Therefore, the scope of this thesis will be on a theoretical best-case scenario in which autonomous inland vessels operate as imposed. Additionally, only vessel and lock interactions will be considered to keep the scope manageable, as these cause the most significant delays and uncertainties in the system. Although essential, other infrastructure objects such as moveable bridges, ports and berths will not be considered. At last, this research will not investigate the problem of intermodal logistics or the allocation of required inland vessels to transport a certain amount of cargo.
Therefore, this research aims to design a scheduling strategy that enables multiple autonomous inland waterway vessels to optimally sail through a waterway network with minimal delays and show a proof of concept of this strategy. The proposed strategy will be validated in computer simulations, not in actual experiments or real data. This research objective has been formulated into the following main research question:

'How to design and implement a Switching Max-Plus Linear scheduling strategy that allows multiple autonomous inland waterway vessels to optimally sail through a waterway network?'

1-3-2 Research questions

In order to answer the main research question, sub research questions have been formulated. In the remaining part of this thesis, the sub research questions will be answered by literature studies and computer simulations. The sub research questions have been formed in such a way that the main research question will be answered after all sub research questions have been answered. The sub research questions will also give structure to the report, in each chapter the sub research question that will be addressed is mentioned in the introduction. The list of sub research questions is as follows:

1. What is the current state of research on IWT scheduling?
    (a) Which actors play a role in IWT systems and how do they operate?
    (b) What research has been done on IWT scheduling?
    (c) What area of IWT scheduling can be improved?
2. How do SMPL systems work and what is required for modelling and scheduling?

(a) What is the motivation for SMPL modelling over other methods?
(b) What are the operators and definitions used in Max-Plus algebra?
(c) What is the system description of SMPL systems?
(d) What is required to model DESs as a SMPL systems?
(e) How to use SMPL systems for scheduling?
(f) How to transform SMPL systems to MILP problems?

3. How to model IWT systems as SMPL systems?
   (a) Why are SMPL systems useful for modelling IWT systems?
   (b) What assumptions have to be made to model IWT systems as SMPL systems?
   (c) What would the SMPL IWT system for different IWT cases look like?
4. How to transform SMPL IWT systems to MILP models?
   (a) How to realise the objective of the scheduler?
   (b) What assumptions have to be made to transform SMPL IWT systems to MILP problems?
   (c) What would the MILP IWT model for different IWT cases look like?
5. How can we verify the designed scheduling strategy for IWT systems?
   (a) What does the overall scheduler optimisation architecture look like?
      (i) What information is required by the scheduler?
      (ii) What information is produced by the scheduler?
   (b) What are the results of the scheduling strategy for different IWT cases?

For the research questions *3(c)*, *4(b)* and *5(b)*, the different cases refer to the following four:

(i) Uni-Directional Fixed Routing case

(ii) Uni-Directional Variable Routing case

(iii) Bi-Directional Fixed Routing case

(iv) Bi-Directional Variable Routing case

These four cases are used to work towards a model, which can cover various IWT system lay-outs. This approach will be further elaborated on in the next Section 1-3-3

1-3-3 Research approach

In order to address the research questions defined in Section 1-3-2 the following approach will be used. First, it is essential to get an understanding of the system we are dealing with; therefore, we will explore IWT systems and their important actors. Moreover, state-of-the-art research and literature on IWT scheduling will be consulted to comprehend the current status of IWT scheduling. Secondly, the theory on modelling of SMPL systems and scheduling with SMPL systems has to be thoroughly studied such that it is exceptionally well understood to the level which allows for modelling of novel systems.
After this theoretical foundation is laid, the modelling and scheduling of IWT systems can start. Figure 1-2 shows how this modelling and scheduling part of the research is executed for four different cases, increasing in complexity, and each in three steps. This is done in this way to slowly build up the complexity of the models. The naming of the cases is explained in Section 4-1-2. The approach is to first start with the easiest IWT system possible, which is called the Uni-Directional Fixed Routing (UDFR). The UDFR case consists of two waterways with a single lock in between, and vessels are only allowed to sail a single direction (i.e. downstream). Secondly, when the results for the UDFR case are satisfactory, we will continue with the slightly more complex Uni-Directional Variable Routing (UDVR) case. The UDVR case consists of two parallel UDFR systems in which vessels can be scheduled on either route but still sail only in a single direction. Again, if the results are satisfactory, the next Bi-Directional Fixed Routing (BDFR) case will be modelled. The BDFR case resembles the UDFR case; however, vessels are allowed to sail in both directions (i.e. downstream and upstream). At last, naturally, the same goes for the Bi-Directional Variable Routing (BDVR) case, which is the same as the UDVR case but with both sailing directions. The latter, is the most generic case and should allow all other possible networks to be created as well. Note that the dotted lines show the feedback loop in the modelling process; in case the schedule results are not as expected, the SMPL models will be revised.

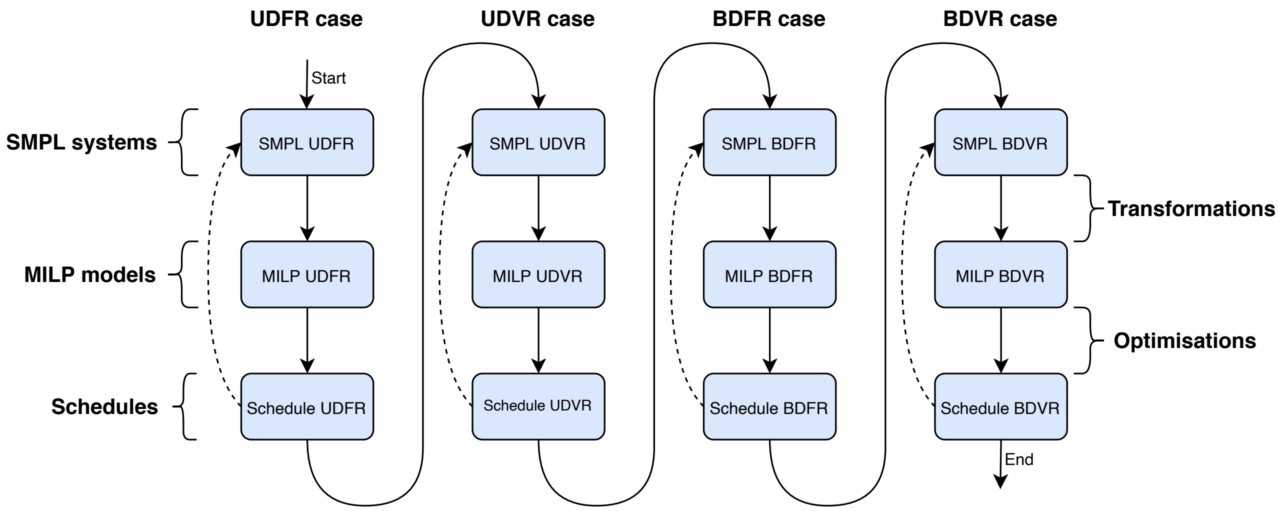


Figure 1-2: Schematic overview of the modelling and scheduling part of the research

Finally, after the developing the models and analysing the scheduling results, the research questions will be answered and some concluding remarks will be given.

After having elaborated on the research approach the next Section 1-4 will present how this research is documented in the remaining part of the thesis.

1-4 Outline of the thesis

The next section presents the organisation and outline of the remaining part of this thesis. A schematic overview of this is also shown in Figure 1-3. It is important to note that the structure of the thesis does not chronologically follow the research approach presented Figure 1-2. Instead of describing each case from SMPL model to scheduling result, the thesis will first discuss the modelling of the four SMPL systems for the four cases in Chapter 4, next all four cases are transformed MILP models in Chapter 5, after which Chapter 6 will present the scheduling results. This reporting structure was chosen because, this makes understanding the research more straightforward and intuitive. Furthermore, Chapter 2 will introduce the IWT system and list state of the art research. Chapter 3 will describe all the theoretical knowledge required for modelling IWT systems as SMPL systems and transforming these to MILP models. At last, Chapter 7, will conclude the research and present some recommendations.

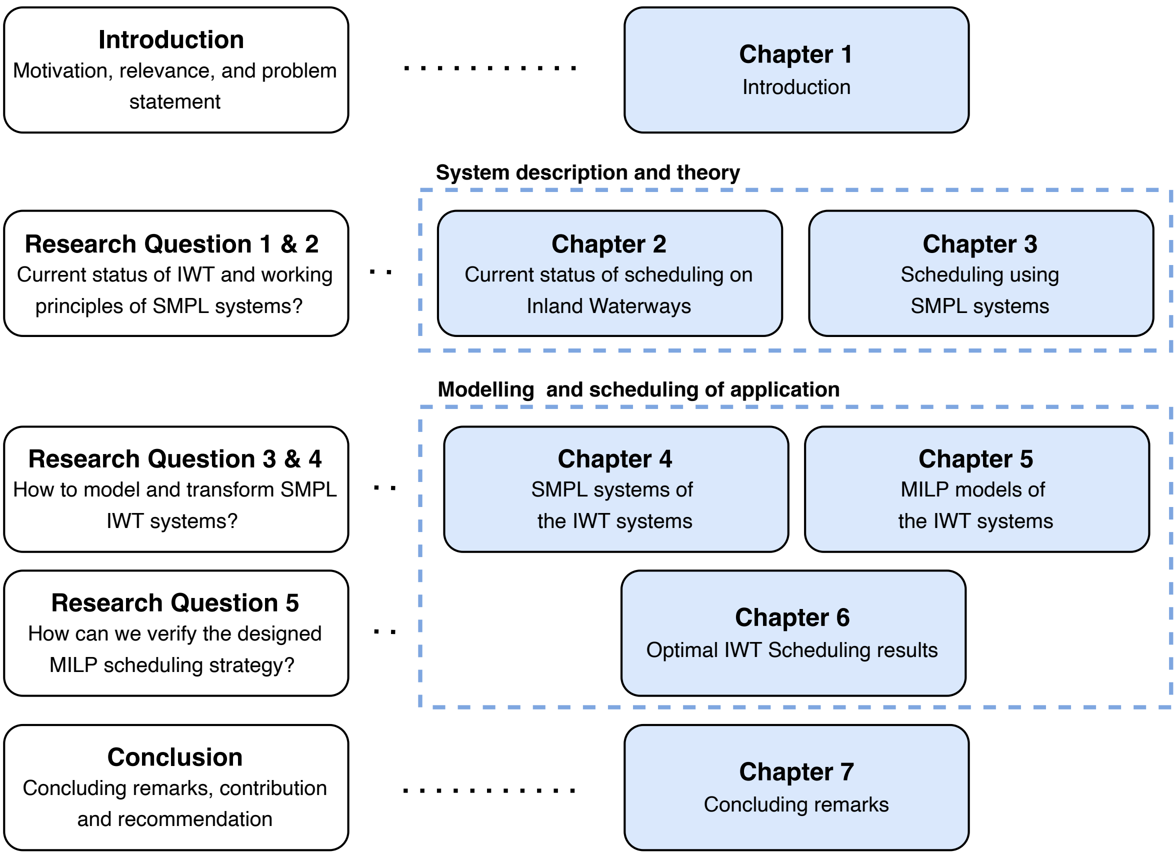


Figure 1-3: Schematic overview of thesis outline

Chapter 2

Current status of scheduling on Inland Waterways

After having introduced the problem statement in Chapter 1, we will continue with laying the foundation for modelling Inland Waterway Transport (IWT) systems. In order order to model and optimally schedule IWT, we first need to have a better understanding of the current status of IWT systems. Therefore, the goal of Chapter 2 is to give a general introduction to IWT systems. To keep the scope manageable, this is mainly done for the Netherlands and also includes a bit of Europe. This system description is done by giving background information on the most important agents active in the IWT system by describing their properties and characteristics. Moreover, the chapter gives a brief summary of state of the art research on scheduling on Inland Waterways.
As introduced in Chapter 1, this chapter will answer research question:

What is the current state of research on IWT scheduling?

This Chapter is structured as follows: First, Section 2-1 will describe the important properties of the IWT system. Next, Section 2-2 will give an overview on what research already has been done on Switching Max-Plus Linear (SMPL) scheduling. At last, Section 2-3 will summarise the findings and will state what area of IWT scheduling can be improved.

2-1 The Inland Waterway Transport system

This section will describe the different aspects and properties of the IWT system, which is mainly based on [41] unless mentioned otherwise. First, the different types of waterways are shown in Section 2-1-1. Section 2-1-2 presents the inland vessel fleet. Next, the operation and important characteristics of locks are described in Section 2-1-3. As mentioned in Section 1-3-1, this research will only focus on the interaction between locks and inland vessel. Although important, other infrastructure objects such as moveable bridges, ports and berths will not be modelled and therefore not discussed in this section.

2-1-1 Waterways

Inland waterways form a natural network structure. The European waterway network comprises over 30.000 kilometres of navigable canals, rivers and lakes, of which approximately 10.000 kilometres is the core network located in the North-Western part of Europe (The Netherlands, France, Germany, Belgium and Austria) [24]. This extensive waterway network allows IWT to reach many destinations in Europe. The Dutch waterways are located at the mouth of several European rivers (the Rhine, the Meuse and the Scheldt); because of this the Dutch IWT system is considered to be the gateway to the European hinterland and provides the Netherlands with an exceptional network for IWT.

To ensure the waterway is used efficiently (i.e. guarantee a certain vessel speed on a route), a corridor and network approach must be taken, whereby waterway management authorities look further than their own area. In this context, a corridor is defined as a cluster of waterways connecting two or multiple (economic) centres [41]. In order to ensure consistency in this network, the European inland waterways are categorised into CEMT classes, which was determined in 1992 by the Conférence Européenne des Ministres de Transport. These classes streamline the waterways' dimensions for the whole of the European network by listing the maximum inland vessel size. This classification makes it immediately clear which waterways are navigable or not navigable due to draft and manoeuvrability [41].

For the Dutch IWT system, the Dutch Ministry for Public Works and Water Management (e.g. Rijkswaterstaat) distinguishes four types of waterway profiles [41], which are: trunk routes, key-waterways, other main waterways and other waterways, which can be seen in Figure 2-1. The trunk routes connect the vital transport hubs of Rotterdam and Amsterdam with the international hinterland (e.g. Germany and Belgium), and the key waterways connect economic areas in the Netherlands with the trunk routes. The waterway status depends on the goods transported on that particular waterway; for instance, trunk routes and key-waterways should transport over 5 million tonnes a year. At last, waterways can have different waterway profiles, which determine the number of vessel lanes on a waterway and if overtaking is allowed.

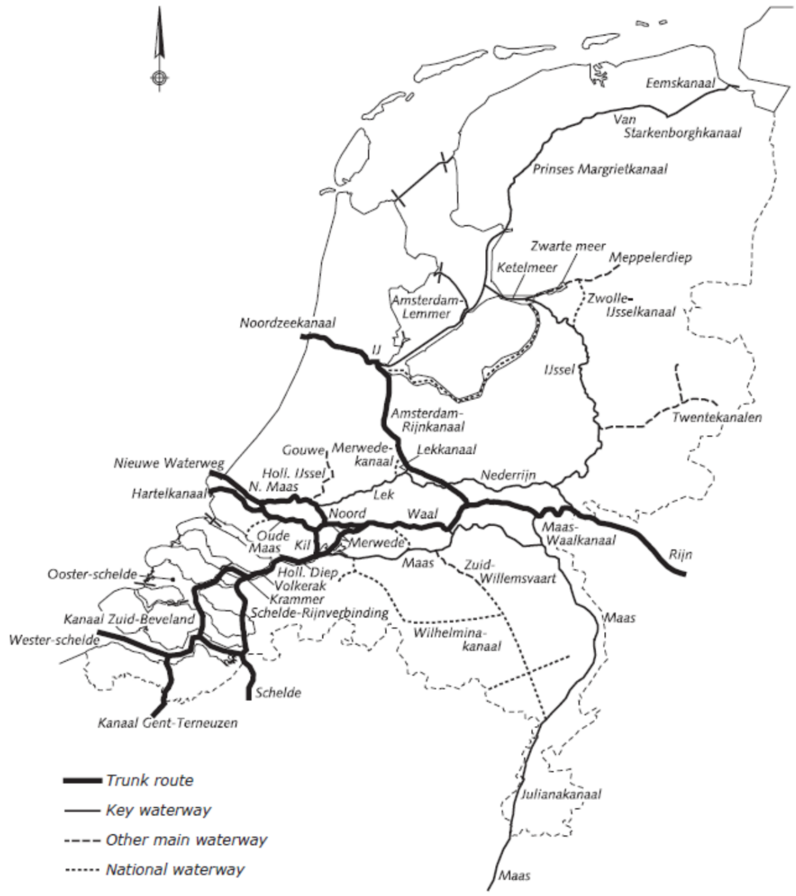


Figure 2-1: Waterway types as defined by the Dutch Ministry Policy Strategy for Infrastructure and Spatial Planning in 2012, according to [41]

2-1-2 Inland vessels

The European inland vessels can be divided into the CEMT-classes (runs from I to VIIa) and RWS-Classes (runs from M0 to M12) [41]. Generally, in the Netherlands, the RWS classification is used to represent the vessel types. The vessel type names are derived from the largest vessel for which the dimensions of the waterway are suitable. These vessels are called the reference vessels and are the largest vessels that can safely and smoothly navigate that particular waterway. Properties of these vessels are summarised in Table 2-1. The cargo type can be divided into motor cargo, pushed convoys, and coupled units. Individual vessel velocity is highly dependent on the vessel type, the cargo loading and the water level.
Vessel navigation and communication has become much more convenient through systems as GP, radar and digital maps. These systems are used to determine the location of a vessel on a waterway which reduces collisions and provides for more safe navigation. More recent advanced communication systems like the Automatic Identification System (AIS) not only allow vessels to communicate with one another but also with independent onshore organisations

Table 2-1: European inland shipping fleet, based on [15, 41]

CEMT-Class	RWS-Class	Vessel type	Length (LOA) [m]	Witdh [m]	Draft [m]	Deadweight [tons]
-	M0	Various smaller vessels				<250
I	M1	Spits (Peniche)	38.5	5.05	2.5	251-400
II	M2	Campine vessel (Kempenaar)	50-55	6.6	2.6	401-650
III	M3	Hagenaar	55-70	7.2	2.6	651-800
III	M4	Dortmund-Ems canal vessel	67-73	8.2	2.7	901-1050
III	M5	Elongated Dortmund-Ems canal vessel	80-85	8.2	2.7	1051-1250
IV	M6	Rhine-Herne canal vessel	80-105	9.5	2.9	1251-1750
IV	M7	Elongated Rhine-Herne canal vessel	105	9.5	3	1751-2050
Va	M8	Large Rhine vessel	110	11.4	3.5	2051-3300
Vb	M9	Elongated large Rhine vessel	135	11.4	4	3301-4000
Vb		Push convoy with 1×2 longitudinal bins	170-190	11.4	3.5-4.0	3951-7050
VIa	M10	Two lighter pushing unit	110	13.5	4	4001-4300
VIa	M11	Gauge vessel	135	14.2	4	4301-5600
VIa	M12	Rhine max vessel	135	17	4	>5601
VIb		Push convoy with 2×2 bins	185-195	22.8	3.5-4.0	6400-12000
VIc		Push convoy with 3×2 bins	270	22.8	3.5-4.0	9600-18000
VIIa		Push convoy with 2×3	195	34.2	3.5-4.0	14500-27000

LOA: Length overall

like waterway management tools (e.g. phone applications, websites), lock management tools and bridge management tools. The AIS includes information like; vessel identity, navigation status, rate of turn, velocity, position, course, and draught [9].

2-1-3 Locks

Inland waterways feature many artificial infrastructure works such as locks, bridges, ports and junctions. These various infrastructures limit the flow of IWT, affect the utilisation of the waterway, and to a large extent, determine the maximum size of the vessels that can navigate the waterway. Since, the scope of the thesis only includes the modelling of lock, only the working principles and unique characteristics of the lock are described in this section.
The difference in water levels between waterways, rivers, canals and lakes shows the necessity for locks. A lock is used for raising and lowering vessels between these different water levels. Locks are the infrastructure types that receive the most attention as they are usually the main bottlenecks in waterway networks [17]. For commercial inland vessels, the lockage sequence is usually based on the First-Come First-Out principle, as lock operators normally only know the vessel arrival times within a close distance to the lock. There is no set of general guidelines that describe all locks since the locks in the trunk routes and key-waterways are always customised to their environmental surroundings. Due to this, many different types of locks exist. This section will focus on a particular lock form called the chamber lock as these are the most common lock forms in the Netherlands and operate on the busiest waterways.
A schematic overview of a lock complex is shown in Figure 2-3. The usable chamber length and width mainly determine the lock dimensions (this being the distance between the stoplines L_k), the sill depth at the reference low water level (MLWS) and the headroom under the lift gates and any bridges over the lock [41]. A chamber lock comprises a chamber with gates at both ends, where the chamber is the space between the two lock gates. The busiest chamber locks often have multiple chamber for vessels and sometimes also separate chambers for recreational vessels. Some properties of these largest locks in the Netherlands are summarised

in Table 2-2 including, the corridor on which the lock is located, the vessel CEMT-class that can pass through the lock, and the number of yearly passages.
After a lock operation has started, the chamber is unavailable for some time as it processes the vessels. During this processing time, the gates are closed. First, the water is levelled with the other side of the lock. Next, the gates open again, allowing the vessels to sail out of the lock. In addition, some extra time is needed to moor the vessels to the lock quay. Vessels on the other side of the lock might be waiting already for their turn to be processed. A chamber can also process without any vessel inside to be ready for inland vessels on the other side. Obviously, a lock can also wait and not process any vessel.

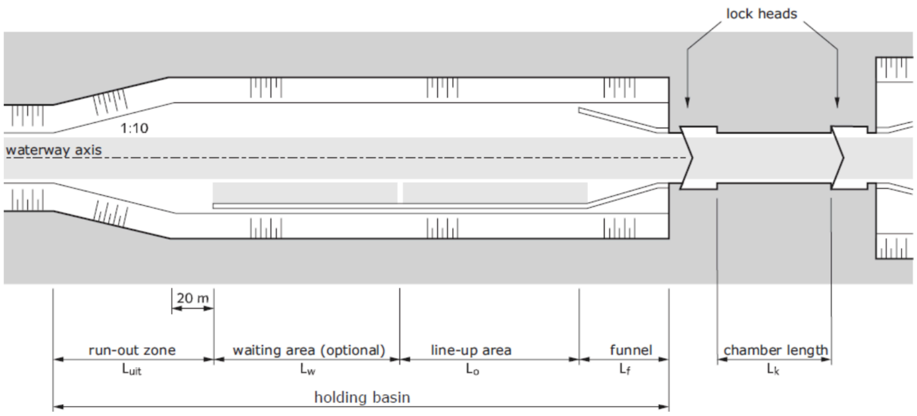


Figure 2-2: Schematic representation of a lock complex including the holding basin (i.e. waiting area) [41]

Table 2-2: Locks in the Dutch IWT system with over 30.000 passages per year excluding recreational vessels [15]

Lock	Corridor	CEMT Class	Chamber	Passages
Volkerak lock	Scheldt–Rhine Canal	VIb	3 + small lock	110331
Kreekrak lock	Scheldt–Rhine Canal	VIb	2	68234
Terneuzen lock	Ghent–Terneuzen Canal	VIb	3	55668
Prinses Beatrix lock	Lek Canal	Vb	3	48984
Hansweert lock	Canal through Zuid-Beveland	VIb	2	43559
Krammer lock	Scheldt–Rhine Canal	VIb	2 + 1 small lock	42211
Oranje lock	the Binnen-IJ	VIa	1 + 2 small locks	41318
Prinses Irene lock	Amsterdam–Rhine Canal	VIb	2	35131
Prins Bernhard lock	Amsterdam–Rhine Canal	VIb	2	32220
Houtrib lock	IJsselmeer	Va	2	31055
Weurt lock	Maas–Waal Canal	Vb	2	30320

In order to reduce the average passage time of vessels, the lock processing efficiency should be increased. This can be improved in different ways such as; speeding up the water levelling, larger chambers, more locks, or faster gate closing. Another method is optimising the order of vessels passing; in other words, the vessel queue must be effectively ordered. In order to do this, the following decisions must be determined; which vessel should be assigned to

which lock chamber? What is the order of sailing into the locks? What times should the lock levelling operation be initiated? These decisions depend on many other future arriving vessels and, in addition, also on their decisions - this creates a rather complex problem.
At last, to show it is important to show that the number of delays at locks is indeed a concern. The bottlenecks at a lock are often analysed using the I/C ratio, which represents the ratio of actual vessel traffic volume I to theoretical lock vessel capacity C expressed in millions of tonnes of passing cargo capacity per year [41]. When the I/C ratio increases, so do the delays. Rijkswaterstaat aims to limit the I/C ratio to 0.6 and will launch a future capacity study for a lock at an I/C ratio of 0.5. The I/C ratio for different locks in the Netherlands can be seen in Figure This shows that the number of delays at locks is already a problem and will only increase [40].

maatgevende maand	2008	2020		2028		2040	
Sluis **I/C Factor -->**		**SE**	**GE**	**SE**	**GE**	**SE**	**GE**
Corridor 2: Amsterdam - Rijn							
Prinses Irene	0.40	0.35	0.45	0.35	0.45	0.35	**0.50**
Prins Bernhard	0.40	0.40	0.45	0.40	0.45	0.40	**0.50**
Amerongen	0.30	0.30	0.35	0.30	0.40	0.30	0.40
Corridor 3: Westerschelde - Rijn							
Volkerak	**0.60**	**0.65**	**0.75**	**0.65**	**0.85**	**0.65**	**1.05**
Kreekrak	**0.60**	**0.65**	**0.75**	**0.65**	**0.90**	**0.70**	**1.10**
Krammer	**0.50**	0.45	**0.55**	0.45	**0.60**	0.45	**0.70**
Hansweert	0.40	0.40	0.40	0.40	0.45	0.40	**0.55**
Corridor 5: Amsterdam - Noord-Nederland							
Oranje	0.40	0.40	0.45	0.40	0.45	0.40	**0.50**
Houtrib	0.35	0.30	0.35	0.30	0.40	0.30	0.40
Prinses Margriet (Lemmer)	**0.55**	**0.60**	**0.65**	**0.60**	**0.65**	**0.60**	**0.70**
Gaarkeuken	0.45	0.45	**0.50**	0.45	**0.55**	0.45	**0.55**
Oostersluis	**0.50**	**0.55**	**0.60**	**0.55**	**0.65**	**0.55**	**0.65**
Corridor 6: Rijn - Oost-Nederland							
Delden	0.45	**0.50**	**0.60**	**0.55**	**0.65**	**0.60**	**0.75**
Corridor 7: Maasroute							
Sluis Weurt	0.45	**0.55**	**0.65**	**0.55**	**0.65**	**0.55**	**0.70**
Sluis Panheel	0.30	0.30	0.30	0.30	0.30	0.30	0.30

Figure 2-3: The I/C ratio for locks in the Netherlands including the prediction for the future [40]

2-2 Research on scheduling of Inland Waterway Transport systems

The goal of this section is to present a broad overview of literature on scheduling methods for the IWT system. This is done by describing the current state of affairs and recognising contemporary challenges. We denote the interaction of vessels with infrastructure as Vessel-to-Infrastructure (V2I). Two earlier performed literature studies on inland vessels were found, however, these do not focus on the V2I interaction [31, 11].
For most infrastructure pieces, the goal of the V2I scheduling is to minimise the time that vessels need to pass through that particular infrastructure piece (e.g. lock, movable bridge, terminal). Vessels passing through infrastructures occupy resources, in this case, both space and time slots. Hence, the main challenge is the optimal allocation of these resources. The scheduling of passing vessels through infrastructures can be formulated as different types of scheduling problems (e.g. Machine Scheduling Problems, Vehicle Scheduling Problem). This section will mainly discuss literature related to V2I scheduling for locks because, as mentioned in Chapter 1, locks are usually the main bottlenecks in waterway networks. These are usually addressed for specific geographical area's (e.g. Mississippi Delta, Yangtze Delta, European

waterway network). This bottleneck problem is less of an issue for infrastructure types such as movable bridges. Moreover, movable bridges can be considered as a simple version of a lock.
A lock can consist of one or multiple chambers. These chambers each have a specific inland vessel capacity and specific lock operation duration, which is both dependent on the water levels and number and types of vessels present in the lock during a particular operation. The scheduling problem that a lock deals with is determining the optimal order of inland vessels passing through that particular lock while meeting these capacity and duration constraints, such that vessel waiting times are minimal.
In [6, 7], the same authors present, respectively, a decision-making tool and a robust simulation tool methods for dealing with congestion on the Upper Mississippi River.
A more generalised and exact method for lock scheduling is discussed in [57]. The complete lock operation of a vessel requires two decision to be made by the lock. First, the lock must determine the position for the vessel; this will be referred to as the Vessel Placement Problem (VPP). Second, the lock must determine the starting time for the lock operation; this will be referred to as the Lock Scheduling Problem (LSP).
Usually, the objective of the VPP is to minimise the number of lock operations needed to place all vessels. This objective is achieved by first deciding the vessel sequence, which is based on the vessel service policy. This policy could be, for instance, a first-come-first-serve policy or a shortest-processing-time-first policy. The first-come-first-serve is what most locks are currently using, in which the vessels are passing through the lock in their arrival order. In [43, 44], it is concluded that the shortest-processing-time-first policy results in less overall network delays than the first-come-first-serve policy.
After deciding the vessel sequence, the second step is determining how the vessels should be arranged in the actual lock chambers while meeting the lock chamber specific capacity constraints. In [54], this chamber arranging problem is recast to a two-dimensional bin packaging problem, in which several rectangular objects (i.e. vessels) need to be placed inside a few larger rectangular bins (i.e. lock chambers) while minimising the total operation time. Additionally, in [55] the same author presents a model for the chamber arranging problem, after which an exact decomposition method and a multi-order best fit heuristic method are compared for finding the best solutions.
As mentioned, the LSP deals with the assigning of chambers and the ordering of a sequence. In [54], these are modelled as a parallel machine scheduling problem where the chambers map to the machines and the lock to the jobs. The authors of [36] introduce the Lock Master's problem. The problem consists of determining the optimal operation plan (i.e. the time when a lock should move up and when it should move down) of a single lock, given a set of upstream and downstream vessels, their arrival times and assuming a constant lock operating time. The objective is to minimise the total sum of ship waiting times. The problem is defined by creating an analogy to the single batching machine scheduling problem. The author develops a dynamic programming algorithm that can solve the problem in polynomial time while taking into account, ship-dependent handling times, water usage, weights and capacities. In [56], a so called late-acceptance algorithm for the lock scheduling problem is proposed that minimise both the waiting time of all inland vessels including the water usage of the lock. Several (meta)heuristics are compared for solving this problem. Research focusing on the vessel order sequence, by making an analogy to the identical parallel machine scheduling problem with sequence dependent setup times and release dates, has been done in [58]. Additional V2I scheduling can be considered at terminals for which the main problems are berth allocation,

quay crane assignment and quay crane scheduling, this is thoroughly reviewed in [42]. Several studies tackle the problem of scheduling inland vessels to berth locations in a port, but not of inland vessels sailing through a waterway network [30]. For instance in [32], a Mixed Integer Linear Programming (MILP) problem for planning rotations of inland Vessels in a large seaport is proposed. At last, when focusing on the whole IWT network, intermodal transport is usually considered instead of more often than V2I interaction [29].
There is limited research on the interaction between infrastructures and inland vessels from the literature. Moreover, communication between lock operators amongst each other and with inland vessel operators is lacking. As a result, there is also minimal research operation and optimisation of infrastructure object. The reviewed literature on V2I scheduling in IWT systems is summarised and compared in Table 2-3.

Table 2-3: Reviewed literature on Vessel to Infrastructure scheduling in IWT systems

Study	Application	Objective	Method
[7]	Mississippi River locks	Reduce congestion	Decision making tool
[6]	Mississippi River locks	Reduce congestion	Simulation tool
[57]	General lock	Ship placement, chamber assignment and lock operation scheduling	MILP
[44]	Single lock	Determine processing policy	SPF vs FCFS
[43]	Multiple locks	Determine processing policy	SPF vs FCFS
[54]	General lock	Ship placement, chamber assignment and lock operation scheduling	Heuristics
[55]	Single lock	Ship placement	Exact & heuristics
[36]	Lock master's problem	Minimise waiting time	Dynamic programming
[56]	Single lock	Minimise waiting time	late-acceptance algorithm
[58]	Single lock	Vessel order and sequencing	(Meta)-heuristics
[42]	Terminal	Berth, Crane scheduling	Literature study

2-3 Conclusion

In this Chapter 2, the general IWT system is described by identifying the properties and characteristics of its agents. Moreover, state of the art research on IWT scheduling is discussed. This information will be used to model IWT systems as SMPL systems later in the research. As stated in Chapter 1, this chapter addressed the research questions:

What is the current state of research on IWT scheduling?

The most important actors active in the IWT system, which will be modelled, are the waterways, vessels and locks. Currently available research on IWT scheduling is mainly focused on one agents (e.g. a single lock) a global network optimisation tool with multiple lock in a row or different routing is still missing. Moreover, most literature is focused on the perspective of the vessel and not the infrastructure object. There is limited research on the interaction between infrastructures and inland vessels. Moreover, communication between lock operators amongst each other and with inland vessel operators is lacking. As a result, there is minimal research on the operation and optimisation of infrastructure object. Therefore, there are still gaps and potential possibilities for future research. A few most notable and relevant gaps and possibilities are summarised:

- *No scheduling at infrastructures*
  Efficient scheduling and coordination of IWT could solve this problem and might yield a significant improvement in making IWT more attractive.

- *No global network optimisation*
  When a delay at a particular lock occurs, the inland vessel's deadline or time window at the next infrastructure could be missed. This deadline missing will in turn, result in even more instabilities to the overall IWT system. Currently, lock operators are not reporting these delays. Scheduling between infrastructures or a global network optimisation could solve this.

As a result, more research regarding the IWT system is relevant. This research extends on the future research possibilities described in [36], where it is stated that the following relevant question would be to look at a problem with multiple locks in series and with vessels sailing downstream and upstream. To deal with this problem, the next Chapter 3 will introduce a promising Discrete Event System (DES) scheduling method based on Max-Plus algebra.

Chapter 3

Scheduling using Switching Max-Plus Linear systems

In order to model Inland Waterway Transport (IWT) systems as Switching Max-Plus Linear (SMPL) systems, we will thoroughly describe the theory on general Max-Plus algebra and SMPL systems in this chapter. Moreover, the chapter provides an extensive introduction to scheduling using these SMPL systems. The Max-Plus algebra's fundamental concepts are presented and it is explained how they can be used to model a specific class of cyclic discrete event systems. Therefore, the goal of this chapter is to lay the foundation for describing the IWT system as an SMPL system.
As introduced in Chapter 1, this chapter will answer the research question:

How do SMPL systems work and what is required for modelling?

This Chapter is structured as follows: First the motivation behind of Max-Plus algebra is given in Section 3-1. Next, the algebra and its new definitions and operators are presented in Section 3-2. Section 3-4 will elaborate on the relationship with graph theory. Before, SMPL systems can be described first an understanding of Max-Plus Linear (MPL) systems is required, which is done in Section 3-3. Next, we will thoroughly describe SMPL systems in Section 3-5 including its most important properties such as routing, ordering and synchronisation. Section 3-6 will show how Max-Plus binary control variables can be parameterised to reduce problem complexity. Furthermore, Section 3-7 will state how SMPL systems can be used for scheduling. To use them for scheduling it is required to transform the SMPL systems to Mixed Integer Linear Programming (MILP) models, which is explained in Section 3-8. At last, Section 3-9 will summarise the findings and answer the corresponding research question.

3-1 Introduction and motivation for scheduling using Max-Plus algebra

As mentioned in the introduction in Section 1-1, for systems whose dynamics are governed by the evolution of events and not time itself, it might be more convenient to describe them as Discrete Event System (DES)s instead of as continuous-time systems; for instance, the in Section 1-1 presented a baggage handling system example. The authors of [8] provide a comprehensive introduction to the DES field, and define a DES as follows:

Definition 1 (*Discrete Event System*)**.** *"A Discrete-Event System is a discrete-state, event-driven system, that is, its state evolution depends entirely on the occurrence of asynchronous discrete events over time."*

Scheduling, in context of this research, deals with the optimisation problem of allocating a set of jobs over certain limited resources that execute these tasks:

Definition 2 (*Scheduling*)**.** *"Scheduling is the process of deciding how to allocate a set of jobs to limited resources over time, in such a way that one or more objectives are optimised"*

Combining these definitions, the scheduling of DESs, in this research, is the process of deciding how to allocate a set of jobs to limited resources over the evolution of events in such a way that one or more objectives are optimised. For instance, DES scheduling deals with determining the route which jobs should take over a particular resource set or determining the order of jobs on a resource (i.e. which job is allowed to use to resource first), while the system evolves as a result of events that happen and not directly as a result of the progression of time. In light of the baggage handling system, the routing could determine which conveyor belt a piece of luggage should take, and the ordering could determine in which order the luggage pieces are deposited on the conveyor. Other questions for other systems are, for instance, selecting which train should go first on a switch track [25], or determining whether an urban train should wait for a passenger transfer [59] or deciding if a piece of paper should be delayed before entering a printer system [1].
Some mathematical models that use conventional algebra to deal with these questions and describe the behaviour of DESs will result in a nonlinear system description due to the Max-operator often occurring in DESs where synchronisation plays a role [3]. Synchronisation in DESs happens when a particular operation can only start when all preceding operations are finished. Thus the starting time of the new operation is equivalent to the maximum time of all individual preceding operations. In the context of the earlier in Section 1-1 mentioned baggage handling system, a plane might only be able to leave when all luggage is loaded. Thus the departure time is equal to the maximum of all the boarding times of all individual luggage pieces. This Max-operator is inherently nonlinear in conventional algebra. A comprehensive example to show this concept of the appearance of the nonlinear Max-operator in conventional algebra DESs with synchronisation is given in Appendix A-1.
There exists a subclass of DESs for which we can get a 'linear' model by only using maximisation and addition as its basic operations. The algebra that only consists of these operations is called Max-Plus algebra [3]. These 'linear' systems are called Max-Plus Linear (MPL) systems. Modelling DESs using Max-Plus algebra has the advantage that the resulting MPL

system can be analysed with some conventional linear system theory tools. Using MPL systems for modelling of DESs, we can analyse the behaviour of the system, which can be used to obtain information about the event evolution. Additionally, there is a close relation between MPL systems and Graph Theory. At last, there are system-theoretical analysis methods for MPL systems in the literature [22], which can be used to find bottlenecks in DESs, for example. In the next sections, we will introduce the basic concepts, definitions, and operators of the Max-Plus algebra.

3-2 Definitions and operators in Max-Plus algebra

First, we will define the Max-Plus algebra. Max-Plus algebra consist of the set $\mathbb{R} \cup \{\varepsilon\}$ denoted by $\mathbb{R}_{\max}$ in which $\mathbb{R}$ is the set of real numbers. Next, we introduce the two neutral numbers of Max-Plus algebra, ε and e, which can be compared to the two neutral numbers in conventional algebra, respectively the zero-element 0 and unit-element 1. These are defined as follow:

$$\epsilon \overset{\text{def}}{=} -\infty \quad \text{and} \quad e \overset{\text{def}}{=} 0 \tag{3-1}$$

For $a, b \in \mathbb{R}_{\max}$ we define the so called 'oplus' operator $\oplus$ and 'otimes' operator $\otimes$, also referred to as tropical addition and tropical multiplication respectively in literature:

$$a \oplus b \overset{\text{def}}{=} \max(a, b) \quad \text{and} \quad a \otimes b \overset{\text{def}}{=} a + b \tag{3-2}$$

With these Max-Plus operators and Max-Plus neutral numbers we can show the resemblance of the neutral numbers with conventional algebra by the following operations:

$$a \oplus \varepsilon = a, \ \forall a \in \mathbb{R}_{\max} \quad \text{and} \quad a \otimes \varepsilon = \varepsilon, \ \forall a \in \mathbb{R}_{\max} \tag{3-3}$$

And for the unit-element:

$$a \otimes e = a, \ \forall a \in \mathbb{R}_{\max} \tag{3-4}$$

The set $\mathbb{R}_{\varepsilon}$, the 'oplus' $\oplus$ and 'otimes' $\otimes$ operations, the zero-element ϵ and the unit-element e define the Max-Plus algebra:

$$\mathcal{R}_{\max} = (\mathbb{R}_{\max}, \oplus, \otimes, \varepsilon, e) \tag{3-5}$$

Max-Plus algebra shares many similarities with conventional algebra. Just as in conventional algebra the 'otimes' $\otimes$ operation happens before the 'oplus' $\oplus$ operation. Which is quite intuitive as can be seen in the following example:

$$\begin{aligned} 5 \oplus 2 \otimes 4 = 5 \oplus (2 \otimes 4) &= 6 \\ \max(5, 2+4) = \max(4, 6) &= 6 \end{aligned} \tag{3-6}$$

Appendix A-2 further elaborates on different properties of the Max-Plus algebra, which are also summarised and compared to conventional algebra in Table 3-1.

Table 3-1: Properties and operators of Max-Plus algebra and its analogies to conventional algebra, with $\forall\{x, y, z\} \in \mathbb{R}_{\max}$. Table based on [22]

Property	Max-Plus Algebra	Conventional Algebra
Associativity	$x \oplus (y \oplus z) = (x \oplus y) \oplus z$	$x + (y + z) = (x + y) + z$
Associativity	$x \otimes (y \otimes c) = (a \otimes y) \otimes c$	$x \times (y \times z) = (x \times y) \times z$
Commutativity	$x \oplus y = y \oplus a$	$x + y = y + x$
Commutativity	$x \otimes y = y \otimes x$	$x \times y = y \times x$
Distributivity	$x \otimes (y \oplus c) = (x \otimes y) \oplus (x \otimes c)$	$x \times (y + z) = (x \times y) + (y \times x)$
Zero element	$x \oplus \varepsilon = x$	$x + 0 = x$
Unit element	$x \otimes e = x$	$x \times 1 = x$
Absorption	$x \otimes \varepsilon = \varepsilon$	$x \times 0 = 0$

Another analogy with conventional algebra is the method for calculating powers. Which is defined as:

$$x^{\otimes n} \stackrel{\text{def}}{=} \underbrace{x \otimes x \otimes \ldots \otimes x}_{n \text{ times}}, \quad \forall x \in \mathbb{R}_{\max}, \quad \forall n \in \mathbb{N} \setminus 0 \tag{3-7}$$

Where $\mathbb{N}$ is the set of natural numbers and $x^{\otimes 0} = e$. At last, all the properties and operators of Max-Plus algebra and its analogies to conventional algebra are summarised in Table 3-1.

Max-Plus algebra for matrices

Next, we will elaborate on how Max-Plus algebra is applied to matrices. In Max-Plus algebra, the set of $n \times m$ matrices is defined by $\mathbb{R}_{\max}^{n \times m}$, with $n \in \mathbb{N} \setminus 0$. The element in row i and column j is denoted by a_{ij}, where $i \in \underline{n} = \{1, \ldots, n\}$ and $j \in \underline{m} = \{1, \ldots, m\}$, such that matrix $A \in \mathbb{R}_{\max}^{n \times m}$ is defined as:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}$$

The Max-Plus sum of the matrices $A \in \mathbb{R}_{\max}^{n \times m}$ and $B \in \mathbb{R}_{\max}^{n \times m}$ is defined as:

$$\begin{aligned} [A \oplus B]_{ij} &= a_{ij} \oplus b_{ij} \\ &= \max(a_{ij}, b_{ij}) \end{aligned} \tag{3-8}$$

The Max-Plus scalar multiplication for $\alpha \in \mathbb{R}_{\max}$ is defined as:

$$[\alpha \otimes A]_{ij} = \alpha \otimes A_{ij} \tag{3-9}$$

The Max-Plus matrix multiplication requires the same structure as conventional linear algebra. For the matrices $A \in \mathbb{R}_{\max}^{n \times s}$ and $B \in \mathbb{R}_{\max}^{s \times m}$ with $i \in \underline{n}$, $k \in \underline{m}$ and $\underline{s} = \{1, \ldots, s\}$, this is defined as:

$$\begin{aligned} [A \otimes B]_{ik} &= \bigoplus_{j=1}^{s} a_{ij} \otimes b_{jk} \\ &= \max_{j \in \underline{s}} \{a_{ij} + b_{jk}\} \end{aligned} \tag{3-10}$$

The zero-matrix in Max-Plus algebra is denoted as $\mathcal{E}(n,m)$, with all elements equal to ϵ. The identity matrix in Max-Plus algebra is denoted as $E(n,m)$, with e along its main diagonal and ϵ elsewhere:

$$E = \begin{pmatrix} e & \varepsilon & \cdots & \varepsilon \\ \varepsilon & e & \cdots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon & \varepsilon & \cdots & e \end{pmatrix}, \quad \mathcal{E} = \begin{pmatrix} \varepsilon & \varepsilon & \cdots & \varepsilon \\ \varepsilon & \varepsilon & \cdots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon & \varepsilon & \cdots & \varepsilon \end{pmatrix} \tag{3-11}$$

For $A \in \mathbb{R}_{\max}^{n\times m}$ these neutral matrices have the following properties:

$$\begin{aligned} A \oplus \mathcal{E}(n,m) = \mathcal{E}(n,m) \oplus A = A \\ A \otimes E(m,m) = E(n,n) \otimes A = A \end{aligned} \tag{3-12}$$

In addition for $k \geq 1$, we have:

$$\begin{aligned} A \otimes \mathcal{E}(m,k) &= \mathcal{E}(n,k) \\ \mathcal{E}(k,n) \otimes A &= \mathcal{E}(k,m) \end{aligned} \tag{3-13}$$

Max-Plus eigenvalues and eigenvectors are analogues to conventional linear algebra:

$$A \otimes v = \lambda \otimes v \tag{3-14}$$

Where $\lambda \otimes v$ thus means the element-wise addition of the entries of Max-Plus eigenvalue λ to the entries of the Max-Plus eigenvector v. Just as in conventional linear algebra, for $A \in \mathbb{R}_{\max}^{n\times n}$ the eigenvector v has n non-ϵ entries.
At last, we define Max-Plus binary variables which will be elaborated on later in this chapter. Define $u \in \mathbb{B}_\epsilon = \{0, \epsilon\}$ as a Max-Plus variable, and $\bar{u} \in \mathbb{B}_\epsilon$ as the adjoint variable which is defined as follows:

$$\bar{u} = \begin{cases} 0 & \text{if } u = \varepsilon \\ \varepsilon & \text{if } u = 0 \end{cases} \tag{3-15}$$

The Max-Plus algebra field is extensive, and this section has omitted some features and areas for the sake of conciseness. However, this introduction is sufficient to understand the Max-Plus algebra-based systems described later in this chapter.

3-3 Max-Plus linear systems

As shown in [14, 46], DESs can be described as MPL systems, if there only is synchronisation but no concurrency takes place or choice is allowed. In other words, no operations should be executed simultaneously or choices need to be made for determining operation order. A benchmark example with a great fit is a production system in which the routing schedule is fixed. These MPL systems are described *explicitly* as follows:

$$\begin{aligned} x(k) = A \otimes x(k-1) \oplus B \otimes u(k) \\ y(k) = C \otimes x(k) \end{aligned} \tag{3-16}$$

Where $A(k) \in \mathbb{R}_{\max}^{n\times n}$, $B(k) \in \mathbb{R}_{\max}^{n\times m}$, and $C(k) \in \mathbb{R}_{\max}^{p\times n}$. Here we have n as the number of states, m as the number of inputs, p as the number of outputs and event or product counter

k. The state vector $x(k)$ contains all the time instants at which events occur for the kth time or kth product, the input vector $u(k)$ contains all the time instants at which the input events occur for the kth time or kth product and output vector $y(k)$ contains all the time instants at which the output events occur for the kth time or kth product. Defining the input vector $u(k)$ can be done in different ways, it can be predefined like the timetable of the railway system or the departures of vessels or it can also be variable like for the arbitrary timing of materials going into a production system. It could even be possible that there is no input vector $u(k)$, this happen when we are just trying to synchronise events for instance synchronising gait patterns in legged locomotion [34]. Usually, the output $y(k)$ is left out of the MPL system description as the state-vector $x(k)$ is generally the output for DESs, thus$y(k) = x(k)$. As mentioned, the MPL system in Eq (3-16) is in explicit form due to the dependency of $x(k)$ on the previous cycle or product $x(k-1)$. However, we commonly end up with an implicit model after modelling a DES [27]. These MPL systems are described *implicitly* as follows:

$$x(k) = A_0(k) \otimes x(k) \oplus A_1(k) \otimes x(k-1) \oplus B(k) \otimes u(k) \tag{3-17}$$

At last, even more general, we can extend Eq (3-17) by not only having a dependency on the previous cycle or product counter $k-1$, but also on an event in cycle or product counter $k-\mu$. For $\mu \in \{\mu_{\min}, \dots, \mu_{\max}\}$ this yield the most general MPL system:

$$x(k) = \left(\bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu(k) \otimes x(k-\mu) \right) \oplus B(k) \otimes u(k) \tag{3-18}$$

In Appendix A-3 a by hand calculated worked out example of a MPL railway system [22] is shown, to clarify the operation of MPL systems.
The main shortcoming of MPL systems is that the model structure is fixed, and changes in the system cannot be modelled. In other words, this $A(k)$ matrix cannot change for different products k. This can be solved by allowing the system to switch between different modes; this will be explained in Section 3-5. However, as we now have the definitions of MPL systems, first, the relationship between Graph Theory and MPL systems will be described in the next section.

3-4 Graph Theory

This section will give a general introduction to Graph Theory and its definitions, as it is used for describing the IWT networks. Moreover, the relationship of Max-Plus algebra and MPL systems with Graph Theory is shown as is described in [22] on which most of this section is based.
A graph can be associated with any square $n \times n$ matrix T. This matrix T is then called the *topology matrix*, and the directed graph is denoted by $\mathcal{G}(T)$, which is defined as:

Definition 3 (*Directed Graph*). *"A directed graph or digraph is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{D})$ where $\mathcal{V}$ is a finite set of nodes (i.e. vertices) and $\mathcal{D} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of possible arcs (i.e. directed edges) (i, j) from node i to j. The nodes set is denoted by $V(\mathcal{G})$ and the arc set by $D(\mathcal{G})$"*

The arc (i, j) is called an incoming arc at j and an outgoing arc at i, which can be the same node. Now suppose that $(i, j) \in \mathcal{D}$ but $(j, i) \notin D$. Then an arc from node i to j exists but an arc from j to i does not exist. Moreover, if we can give the arcs a weight $\tau_{i,j}$, in that case the graph $\mathcal{G}$ is called a weighted directed graph. However, as we will only be dealing with weighted directed graphs these will be referred to as graph. Now consider a matrix $T \in \mathbb{R}_{\max}^{n\times n}$ and define the following:

$$t_{ij} = \begin{cases} \tau_{i,j} & \text{if } (i,j) \in \mathcal{N}(T) \times \mathcal{N}(T) \\ \varepsilon, & \text{otherwise} \end{cases} \tag{3-19}$$

For example, for the following topology matrix T defined in Eq (3-23):

$$T = \begin{bmatrix} \varepsilon & \tau_{1,2} & \varepsilon \\ \tau_{2,1} & \tau_{2,2} & \tau_{2,3} \\ \tau_{3,1} & \varepsilon & \varepsilon \end{bmatrix} \tag{3-20}$$

We get graph $\mathcal{G}(T)$ shown in Figure 3-1:

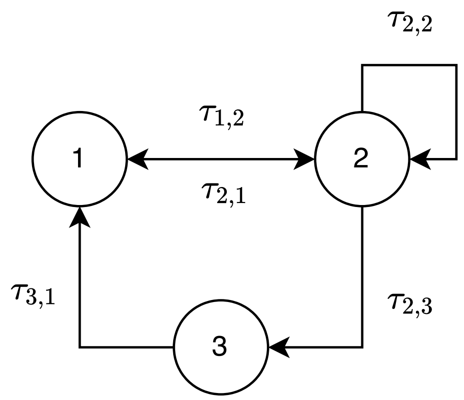


Figure 3-1: Example graph $\mathcal{G}(T)$, with $n\left(\mathcal{V}(T)\right) = 3$ and $n(\mathcal{D}(T)) = 5$

We get the following set of Max-Plus equations:

$$\begin{aligned} x_1(k) &= (\varepsilon \otimes x_1(k-1)) \oplus (\tau_{2,1} \otimes x_2(k-1)) \oplus (\tau_{3,1} \otimes x_3(k-1)) \\ x_2(k) &= (\tau_{1,2} \otimes x_1(k-1)) \oplus (\tau_{2,2} \otimes x_2(k-1)) \oplus (\varepsilon \otimes x_3(k-1)) \\ x_3(k) &= (\varepsilon \otimes x_1(k-1)) \oplus (\tau_{2,3} \otimes x_2(k-1)) \oplus (\varepsilon \otimes x_3(k-1)) \end{aligned} \tag{3-21}$$

Which we can use get the following a MPL system of the form:

$$x(k) = A \otimes x(k-1) \tag{3-22}$$

With:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} = \begin{bmatrix} \varepsilon & \tau_{2,1} & \tau_{3,1} \\ \tau_{1,2} & \tau_{2,2} & \varepsilon \\ \varepsilon & \tau_{2,3} & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \end{bmatrix} \tag{3-23}$$

As can be seen $T^T = A$, which shows the close relationship between Graphs and MPL systems. At last, for any set node set $(i,j) \in \mathcal{G}$, a sequence of connected arcs (i.e. path) denoted by $p = (i_k, j_k)$ is called a path from i to j. By using the definition of a path, we can say there a graph is strongly connected if there is a path from node i to node j for $\forall i, j \in \mathcal{D}$.

3-5 Switching Max-Plus Linear systems

As explained in Section 3-3, only a limited number of DESs can be described as a general MPL system due to the fixed model structure. This fixed model structure means that the system dynamics will always remain the same and cannot be influenced in any way. For example, when event order changes or synchronisation links are broken, the MPL system model is not reliable anymore. A way of dealing with this is to allow for switching between different modes of operation; a MPL system that has this switching ability is called a SMPL system. This is first introduced in [46], a modelling framework is developed in [50], and thoroughly explained for scheduling in [51]. This section is mainly based on these studies unless mentioned otherwise.
The IWT system is not fixed, and the system dynamics can change; for instance, a vessel takes a different route through a waterway network, or the order of vessels on a waterway is changed. Thus, when modelling the IWT system, this switching ability of SMPL systems should be exploited. Therefore, it is highly relevant to thoroughly discuss the general approach for modelling a SMPL system. This section will first present the SMPL model, after which the three decisions which are crucial for scheduling SMPL systems, namely, routing, ordering, and synchronisation, are discussed in the respective subsections. These are eventually combined to describe the full SMPL model.
Thus as mentioned earlier, the SMPL system is a DES that can switch between modes of operation. Let the operation modes be defined as $\ell(k) \in \{1, \cdots, n_m\}$, for discrete event k or product counter k and n_m operation modes. Then the *explicit SMPL system* is of the form as Eq. (3-24):

$$\begin{aligned} x(k) &= A(\ell(k)) \otimes x(k-1) \oplus B(\ell(k)) \otimes u(k) \\ y(k) &= C(\ell(k)) \otimes x(k) \end{aligned} \tag{3-24}$$

Again, we commonly end up with an implicit models after modelling DESs [47]. The *implicit SMPL system* is of the form as Eq. (3-25):

$$x(k) = A_0(\ell(k)) \otimes x(k) \oplus A_1(\ell(k)) \otimes x(k-1) \oplus B(\ell(k)) \otimes u(k) \tag{3-25}$$

At last, even more general, we can extend Eq (3-25) by not only having a dependency on the previous cycle or product counter $k-1$, but also on an event in cycle or product counter $k-\mu$. For $\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$ this yield the most general SMPL system:

$$x(k) = \left(\bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu(\ell(k)) \otimes x(k-\mu) \right) \oplus B(\ell(k)) \otimes u(k) \tag{3-26}$$

To clarify, this means that the dynamics of product k can be influenced by products $k-\mu_{\min}$ up to product $k-\mu_{\max}$.

3-5-1 Routing in SMPL systems

The first type of control decision is routing [51]. To summarise the goal of routing with respect to the proposed framework, a job consists of several operations that have to be executed using different resources. Deciding on how a job follows a sequence of resources is called *routing.*
Consider a system that has to operate M jobs. For each job a specific route through the

system has to be scheduled and resources have to be ordered accordingly. We consider job $j \in \{1, \cdots, M\}$ which consists of p_j operations on the sequence of resources denoted by $\mathcal{R}_j = \left(r_{j,1}, \ldots, r_{j,p_j}\right)$, which is in processing order. Next we consider the corresponding processing times in cycle k to be $\mathcal{T}_j(k) = \left(\tau_{j,1}(k), \ldots, \tau_{j,p_j}(k)\right)$ with $\tau_{j,i}(k) \geq 0 \quad \forall i, j$. For instance, the processing times could represent the duration of manufacturing of a product on a machine, or the travel time from an event to the next event. We assume each operation to be assigned to a unique machine and that it cannot be interrupted. At last, we define the vector with all operation starting times of job j to be $\hat{x}_j(k) = \left[x_{j,1}(k) \ldots x_{j,p_j}(k)\right]^T$. To clarify, the elements (i.e. states) of this vector $\hat{x}_j(k)$ are time instants, thus $x_{j,1}(k)$ is the starting time of event or operation 1 for job j. This will give the following inequalities for all jobs $j \in \{1, \cdots, M\}$:

$$x_{j,m}(k) \geq x_{j,l}(k) + \tau_{j,l}(k), \text{ with } m > l, \quad m, l \in \{1, \ldots, p_j\} \tag{3-27}$$

This inequality shows the following; operation m of job j can only start when operation l of job j is finished. In other words, the starting time of operation l includes the processing time of operation l. This inequality holds for the whole vector which consists of the starting times $x_j(k)$ and all its corresponding processing times. Using the Max-Plus matrix notation described in Section 3-2, we get the following:

$$\begin{bmatrix} x_{j,1}(k) \\ x_{j,2}(k) \\ \vdots \\ x_{j,p_j}(k) \end{bmatrix} \geq \begin{bmatrix} \varepsilon & \varepsilon & \cdots & \varepsilon \\ \tau_{j,1}(k) & \varepsilon & \cdots & \varepsilon \\ \vdots & \ddots & \ddots & \vdots \\ \varepsilon & \cdots & \tau_{j,p_j-1}(k) & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} x_{j,1}(k) \\ x_{j,2}(k) \\ \vdots \\ x_{j,p_j}(k) \end{bmatrix} \tag{3-28}$$

Which can be written as shorter as:

$$\hat{x}_j(k) \geq \hat{A}_0^{\text{job},j}(k) \otimes \hat{x}_j(k) \tag{3-29}$$

If we have all jobs $j \in \{1, \cdots, M\}$, we can collect all the starting time in one state vector $x(k)$ and the following equation is obtained:

$$x(k) = \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \vdots \\ \hat{x}_M(k) \end{bmatrix} \geq \begin{bmatrix} \hat{A}_0^{\text{job},1}(k) & \varepsilon & \cdots & \varepsilon \\ \varepsilon & \hat{A}_0^{\text{job},2}(k) & & \varepsilon \\ \vdots & & \ddots & \vdots \\ \varepsilon & \cdots & \cdots & \hat{A}_0^{\text{job},M}(k) \end{bmatrix} \otimes \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \vdots \\ \hat{x}_M(k) \end{bmatrix} \tag{3-30}$$

Which can be written as shorter as:

$$x(k) \geq A_0^{\text{job}}(k) \otimes x(k) \tag{3-31}$$

The described method has two shortcomings, which will be described and addressed. The first shortcoming is not being able to deal with cyclic jobs. In some applications, the jobs are not fully processed or finished in one job cycle but need one or more job cycles to be completed. For instance, this occurs in a railway network with a cyclic timetable, in which a in cycle $k-1$ departing train will arrive at the next station in cycle k. If this is the case, the state equation is given by:

$$x(k) \geq A_0^{\text{job}}(k) \otimes x(k) \oplus A_1^{\text{job}}(k) \otimes x(k-1) \tag{3-32}$$

The inequality sign is used in Eq. (3-32) because the departing times may also depend on the later discussed ordering and synchronisation, which might delay starting times. Therefore it is upper bounded. To remark, Eq (3-32) will not be used for modelling the IWT system as k will be a product counter (i.e. vessel counter) and thus Eq (3-31) will be sufficient.
The second shortcoming is not allowing different job routes. However, often these alternative routes are available. For instance, a production system that consist of machines which can produce different products wants to switch to another product that requires a different machine order. In this case, an alternative route is used. If we consider to have L alternative sets of routes for the system: then for each set of routes the following matrices can be defined $A^{\text{job}}_{\mu,\ell}(k)$ for $\mu \in \{\mu_{\min}, \dots, \mu_{\max}\}$ and $\ell = 1, \dots, n_\ell$. Next, a set of Max-Plus binary routing variables $(s_1(k), \dots, s_{n_\ell}(k))$ is defined such that when for product k the ℓ-th set of alternative routes for the system is considered, $s_m(k) = e$ and $s_\ell(k) = \varepsilon, \forall \ell \neq m$. Then systems' job matrices can be written for $\mu \in \{\mu_{\min}, \dots, \mu_{\max}\}$ as follows:

$$A^{\text{job}}_{\mu}(s(k)) = \bigoplus_{\ell=1}^{L} s_\ell \otimes A^{\text{job}}_{\mu,\ell}(k) \tag{3-33}$$

These two addressed shortcomings apply to the IWT system. In the context of the IWT system, routing can be used to describe the route the inland vessels take through the infrastructure network and the selection of chambers in an lock.

3-5-2 Ordering in SMPL systems

The second type of control decision that will be explained is ordering [51]. To summarise the goal of ordering with respect to the proposed framework, after the job routes have been determined, conflicts might arise when multiple jobs need to be operated at the same resource at the same time. Then the next decision variable *ordering* comes into play. When conflicts arise, the order of the concurring jobs at resources needs to be determined.
We consider a system with n operations, which are divided over N resources. Using the method and results of the previous subsection, we have L sets of alternative routes, which are a function of the earlier defined Max-Plus binary variables $s(k)$. Next we define the matrix $P_\ell \in \mathbb{B}_\varepsilon^{n\times n}, \ell \in \{1, \cdots, L\}$ with Max-Plus binary entries, where $[P_\ell]_{i,j} = e$ if operation i and operation j are executed on the same resource, and $[P_\ell]_{i,j} = \epsilon$ if operation i and operation j are executed on different resources:

$$[P_\ell]_{i,j} = \begin{cases} e, & \text{if operation } i \text{ and operation } j \text{ are executed on the same resource} \\ \varepsilon, & \text{if operation } i \text{ and operation } j \text{ are executed on different resources} \end{cases}$$

Then the resource allocation matrix $P(w(k))$ is defined as follows:

$$P(w(k)) = \bigoplus_{\ell=1}^{L} w_\ell(k) \otimes P^\ell \tag{3-34}$$

Next, we consider separation time matrix $H(k)$, where $H_{i,j}(k) = h_{ij}(k)$ is the separation time between operations i and j if they may be scheduled on the same resource with $h_{ij}(k) \neq \varepsilon$. If they may not be scheduled on the same resource we have $H_{i,j}(k) = \varepsilon$, thus:

$$[H(k)]_{i,j} = \begin{cases} h_{ij}(k), & \text{if operation } i \text{ and operation } j \text{ may be scheduled on the same resource} \\ \varepsilon, & \text{if operation } i \text{ and operation } j \text{ can never be scheduled on the same resource} \end{cases}$$

To clarify, $h_{ij}(k)$ is the separation time in cycle k between operations i and j. Finally, we consider $Z_\mu(k)$, $\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$ to be order decision matrices with Max-Plus binary entries. Here $[Z_\mu(k)]_{i,j} = e$ is the case when operation i in cycle k is scheduled after operation j in cycle $k+\mu$. The case when operation i in cycle k is scheduled before operation j in cycle $k+\mu$ is $[Z_\mu(k)]_{i,j} = \epsilon$, thus

$$[Z_\mu(k)]_{i,j} = \begin{cases} e, & \text{if operation } i \text{ in cycle } k \text{ is scheduled after operation } j \text{ in cycle } k+\mu \\ \varepsilon, & \text{if operation } i \text{ in cycle } k \text{ is scheduled before operation } j \text{ in cycle } k+\mu \end{cases}$$

If we define $z_\mu(k)$ as the stacked column vector of matrix $Z_\mu(k)$, in the following way $Z_\mu(k) = Z(z_\mu(k))$. Then the following notation $Z_\mu(k) = Z(z_\mu(k))$ can be used, and, at last, the ordering matrices are defined as such:

$$A_\mu^{\text{ord}}\ (w(k), z_\mu(k)) = P(w(k)) \odot Z(z_\mu(k)) \odot H(k) \tag{3-35}$$

Then the constraints for defining the operation order follow:

$$x(k) \geq A_0^{\text{ord}}\ (w(k), z_0(k)) \otimes x(k) \oplus A_1^{\text{ord}}\ (w(k), z_1(k)) \otimes x(k-1) \tag{3-36}$$

To make some concluding remarks, it is important to point out that some values of $z_\mu(k)$ could result in infeasible solutions (i.e. an infeasible schedule) due to cycles in the ordering. For instance, this could occur when there are three operation starting times $\{x_1, x_2, x_3\}$ and the following ordering is made $x_1 > x_2$, $x_2 > x_3$, and $x_3 > x_1$.
Ordering in the context of the IWT system can be used to describe the order of the inland vessels passing through the infrastructures.

3-5-3 Synchronisation in SMPL systems

The third SMPL decision variable is synchronisation [51]. When jobs are running on different resources, they might need synchronisation. To summarise the goal of synchronisation with respect to the proposed framework, synchronisation happens when an operation of a job can only start when the operation of another job has finished. For instance, at a railway network, the arrival of trains needs to be synchronised for when passengers need to change trains. Another example is synchronisation in the legged locomotion, where it is used for creating different gait patterns.
The number of synchronisation modes is defined by $\ell = 1, \ldots, L_{\text{syn}}$, where for every synchronisation mode a system matrix is obtained for $\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$ as follows:

$$\left[A_{\mu,\ell}^{\text{syn}}(k)\right]_{ij} = \begin{cases} 0 \text{ if operation } j \text{ in cycle } k \text{ is to be scheduled behind} \\ \ \text{operation } i \text{ in cycle } k-\mu. \\ \varepsilon \text{ elsewhere} \end{cases} \tag{3-37}$$

Next, the Max-Plus binary variables for synchronisation scheduling are determined as $c(k) \in \mathbb{B}_\varepsilon^{L_{syn}}$. By using this, the synchronisation constraints are given as follows:

$$x(k) \geq \bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu^{syn}(c(k), k) \otimes x(k-\mu) \tag{3-38}$$

where for $\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$:

$$A_{\mu}^{\text{syn}}(c_{\mu}(k)) = \bigoplus_{\ell=1}^{L_{syn}} c_{\mu,\ell}(k) \otimes A_{\mu,\ell}^{\text{syn}}(k) \tag{3-39}$$

To clarify, $[c_{\mu}(k)]_{\ell} = 1$ means that synchronisation l is made and $[c_{\mu}(k)]_{\ell} = 0$ means that synchronisation l is cancelled.
Synchronisation in the context of the IWT system can be used to describe the synchronised arrival of multiple vessels at a multi-chambered lock in order for the inland vessels to pass simultaneously.

3-5-4 Complete SMPL system

Before the overall scheduling of the SMPL model is finalised, it is first explained how a reference signal would fit into the problem. Some DESs have an available schedule that gives a lower bound on the systems operations' starting time. For example, a railway network with an earlier defined timetable. If $r_i(k)$ is the starting time of operation i, according to the earlier defined timetable, then the lower bound on the starting time $x(k)$ can be a constraint as follows:

$$x(k) \geq r(k) \tag{3-40}$$

Finally, the four conditions for the starting time $x(k)$ for the SMPL model were derived in the previous subsections. These are Eq. (3-32), Eq. (3-36), Eq. (3-38) and Eq. (3-40). In addition, a set of SMPL scheduling decision variables is determined; for routing $s_l(k)$, $\ell \in \{1, \ldots, n_{\ell}\}$, for ordering $w_{\mu}(k)$, $\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$, for synchronisation $c_{\mu}(k)$,$\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$. Stacking these in a vector gives, as in [51]:

$$v(k) = \begin{bmatrix} s_1(k) \\ \vdots \\ s_{n_\ell}(k) \\ w_{\mu_{\min}}(k) \\ \vdots \\ w_{\mu_{\max}}(k) \\ c_{\mu_{\min}}(k) \\ \vdots \\ c_{\mu_{\max}}(k) \end{bmatrix} \in \mathbb{B}_{\varepsilon}^{(L_{\text{tot}})} \tag{3-41}$$

In which L_{tot} is defined as the total number of scheduling variables. Accordingly, the SMPL scheduling model can be written as follows:

$$x(k) \geq A_0(v(k)) \otimes x(k) \oplus A_1(v(k)) \otimes x(k-1) \tag{3-42}$$

where for $\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$:

$$\begin{aligned} A_{\mu}(v(k)) &= A_{\mu}^{\text{job}}(s_{\ell}(k)) \oplus A_{\mu}^{\text{ord}}\left(s_{\ell}(k), w_{\mu}(k)\right) \oplus A_{\mu}^{\text{syn}}\left(c_{\mu}(k)\right) \\ &= \bigoplus_{\ell=1}^{L_{\text{tot}}} v_{\ell}(k) \otimes A_{\mu,\ell}(k) \end{aligned} \tag{3-43}$$

Now the system can switch between operation modes by defining $v(k)$. Again, we can generalise Eq (3-42) by not only having a dependency on the previous cycle or product $k-1$, but also on an event in cycle or product $k-\mu$. For $\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$ this yield the most general SMPL system with the vector Max-Plus binary control vector $v(k)$:

$$x(k) = \left(\bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu(v(k)) \otimes x(k-\mu) \right) \oplus B(v(k)) \otimes u(k) \tag{3-44}$$

To clarify, this means that the dynamics of product k can be influenced by products $k-\mu_{\min}$ up to product $k-\mu_{\max}$. At last to remark, the SMPL system can get complex quite fast if predefined system matrices for every mode are needed. This is doable for small systems; however, for large complex systems (e.g. the complete Dutch IWT network), this would result in a large number of possible modes. Due to its multiplicative nature, when scaling the problem, cycle or product k increases rather quickly. For example, 4 routes, 5 orders, and only 2 synchronisations possibilities already results in 40 required predefined system matrices. In order to deal with this the next section will describe parameterisation of the Max-Plus routing binary control variables and the Max-Plus ordering binary control variables.

3-6 Parameterisation of Max-Plus binary control variables

In this section, we will discuss the reparametrisation of the Max-Plus binary control variables, which is mainly based on the theory presented in [51]. The goal is to reduce the number of Max-Plus binary variables such that the complexity of the MILP is reduced. This is done in Section 3-6-1 for the Max-Plus routing binary control variables and in Section 3-6-2 for the Max-Plus ordering binary control variables.

3-6-1 Routing binary control variables

Firstly, reparametrisation for the Max-Plus routing binary control variables. Recall Eq. (3-33) with Max-Plus binary routing control variables $(s_1, s_2, \ldots, s_L)$ for a system with L alternative routes:

$$x(k) \geq \bigoplus_{\ell=1}^{L} s_\ell \otimes A_{\mu,\ell}^{\text{job}}(k) \tag{3-45}$$

Or when the routing is dependent on foregoing jobs (i.e. vessels) we get:

$$x(k) \geq \bigoplus_{\ell=1}^{L} s_\ell \otimes A_{\mu,\ell}^{\text{job}}(k) \oplus \bigoplus_{\ell=1}^{L} s_\ell \otimes A_{1,\ell}^{\text{job}}(k) \otimes x(k-1) \tag{3-46}$$

We can bound the number of alternative routes L as follows:

$$2^{m-1} < L \leq 2^m \tag{3-47}$$

Next, we define the parameterised Max-Plus binary routing control variables $(\eta_1, \ldots, \eta_m)$ and the adjoint parameterised Max-Plus binary routing control variables $(\bar{\eta}_1, \ldots, \bar{\eta}_m)$. Now, we can use this to m to parameterise the original Max-Plus binary routing control variables.

This will be demonstrated by two examples.
For example for $L = 4$, then $[s_1(k), \dots s_4(k)]$, is parameterised by $[\eta_1(k), \eta_2(k)]$ as follows:

$$\begin{aligned} s_1(k) &= \bar{\eta}_1(k) \otimes \bar{\eta}_2(k) \\ s_2(k) &= \bar{\eta}_1(k) \otimes \eta_2(k) \\ s_3(k) &= \eta_1(k) \otimes \bar{\eta}_2(k) \\ s_4(k) &= \eta_1(k) \otimes \eta_2(k) \end{aligned} \tag{3-48}$$

Where the values of $[\eta_1(k), \eta_2(k)]$ for each alternative route $l(k)$ can be seen in Table 3-2:

Table 3-2: Parameterisation for $L = 4$

$\ell(k)$	$\eta_1(k)$	$\eta_2(k)$
1	ε	ε
2	ε	e
3	e	ε
4	e	e

For example for $L = 8$, then $[s_1(k), \dots s_8(k)]$, is parameterised by $[\eta_1(k), \dots, \eta_8(k)]$ as follows:

$$\begin{aligned} s_1(k) &= \eta_1(k) \otimes \eta_2(k) \otimes \eta_3(k) \\ s_2(k) &= \bar{\eta}_1(k) \otimes \bar{\eta}_2(k) \otimes \eta_3(k) \\ s_3(k) &= \bar{\eta}_1(k) \otimes \eta_2(k) \otimes \bar{\eta}_3(k) \\ s_4(k) &= \bar{\eta}_1(k) \otimes \eta_2(k) \otimes \eta_3(k) \\ s_5(k) &= \eta_1(k) \otimes \bar{\eta}_2(k) \otimes \bar{\eta}_3(k) \\ s_6(k) &= \eta_1(k) \otimes \bar{\eta}_2(k) \otimes \eta_3(k) \\ s_7(k) &= \eta_1(k) \otimes \eta_2(k) \otimes \bar{\eta}_3(k) \\ s_8(k) &= \eta_1(k) \otimes \eta_2(k) \otimes \eta_3(k) \end{aligned} \tag{3-49}$$

Where the values of $[\eta_1(k), \dots, \eta_8(k)]$ for each alternative route $l(k)$ can be seen in Table 3-3:

Table 3-3: Parameterisation for $L = 8$

$\ell(k)$	$\eta_1(k)$	$\eta_2(k)$	$\eta_3(k)$
1	ε	ε	ε
2	ε	e	ε
3	e	ε	ε
4	e	e	ε
5	ε	ε	e
6	ε	e	e
7	e	ε	e
8	e	e	e

The relationship between the number of routes L and the number of required parameterised Max-Plus binary routing control variables can be seen in table Table 3-4.

Table 3-4: The number of routes L, compared with the required number of Max-Plus routing control variables $s(k)$ and the reduced number of parameterised Max-Plus routing control variables m^{job}

L	$s_l(k)$	m^{job}
2	2	1
3	3	2
5	5	3
10	10	4
15	15	4
30	30	5
60	60	6

Thus the number of Max-Plus binary control variables needed can be reduced by substituting the function $w_l(k) = f_l(\eta(k))$ into Eq. (3-46), we get:

$$A_{\mu}^{\text{job}}(\eta(k)) = \bigoplus_{\ell=1}^{L} f_{\ell} \otimes A_{\mu,\ell}^{\text{job}}(k) \tag{3-50}$$

This way the MILP only has to find the optimal values of η_i. Note, there will be to much combinations of η_i and $\bar{\eta}_i$ present when L is not an exact power of 2. In order to describe the allowed set, an additional constraint has to be introduced. This constraint has the following form:

$$\sum_{i=1}^{m} 2^{m-i} \eta_i^c \leq L - 1 \tag{3-51}$$

Where the parameterised conventional control variable η_i^c is related to the parameterised Max-Plus Binary control variable η_i, as follows:

$$\eta_i^c = \begin{cases} 0, & \text{for } \eta_i = e \\ 1, & \text{for } \eta_i = \varepsilon \end{cases} \tag{3-52}$$

This relationship will be further elaborated on in section 3-8-1. For instance, lets say we have 20 different routes, thus $L = 20$. Then using the earlier mentioned bound $2^{m-1} < L \leq 2^m$, we get $m = 5$. As 20 is not an exact power of 2 and $2^5 = 32$ we have to add the following constraint:

$$16\eta_1^c + 8\eta_2^c + 4\eta_3^c + 2\eta_4^c + \eta_5^c \leq 19 \tag{3-53}$$

This will make sure that just the allowed set is described. To conclude, reparametrisation of the Max-Plus routing binary control variables will reduce the number of variables required and thus reduce the complexity and computation time of the problem.

3-6-2 Ordering binary control variables

Secondly, we will do the reparametrisation for the Max-Plus ordering binary control variables. For system with a large number of jobs (i.e. vessels) it is computationally demanding to consider all possible ordering combinations and thus determine a large number of ordering

control variables. Therefore, this section will describe how these Max-Plus ordering binary control variables can be reparameterised [50, 51].
Recall Eq. (3-36) with Max-Plus binary ordering control variables $(w_{\mu_{\min}}, \ldots, w_{\mu_{\max}})$:

$$x(k) \geq \bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_{\mu}^{\mathrm{ord}}\ (s(k), w_{\mu}(k), k) \otimes x(k-\mu) \tag{3-54}$$

For scheduling n jobs (i.e. vessels) we require the following number of Max-Plus binary control variables:

$$w_{\mu}(k) = n(n-1)/2 \tag{3-55}$$

Next, we define the parameterised Max-Plus binary ordering control variables $(\gamma_{\mu,1}, \ldots, \gamma_{\mu,m})$ and the adjoint parameterised Max-Plus binary ordering control variables $(\bar{\gamma}_{\mu,1}, \ldots, \bar{\gamma}_{\mu,m})$. Now, we can use this to parameterise the original Max-Plus binary ordering control variables as defined in [51]. For n jobs (i.e. vessels) we require m^{ord} number of parameterised Max-Plus binary ordering control variables. With m^{ord} defined as:

$$m^{\mathrm{ord}} = \lceil \log_2 n! \rceil \tag{3-56}$$

Lets consider an example with 3 vessels, thus $n = 3$. We have 3 original Max-Plus binary ordering control variables:

$$w_{\mu}(k) = n(n-1)/2 = 3(3-1)/2 = 3 \tag{3-57}$$

Moreover, we have the ordering matrix W:

$$W(w_{\mu}(k)) = \begin{bmatrix} \varepsilon & \bar{w}_{\mu,1}(k) & \bar{w}_{\mu,3}(k) \\ w_{\mu,1}(k) & \varepsilon & \bar{w}_{\mu,2}(k) \\ w_{\mu,3}(k) & w_{\mu,2}(k) & \varepsilon \end{bmatrix} \tag{3-58}$$

Where we have $n! = 6$ possible combinations of $w_{\mu,1}(k)$, $w_{\mu,1}(k)$, and $w_{\mu,1}(k)$. By using equation Eq 3-56, we can then parameterise the orgiginal Max-Plus binary control variables by m^{ord} number of parameterised Max-Plus binary control variables, as follows:

$$m^{\mathrm{ord}} = \lceil \log_2 n! \rceil = 3 \tag{3-59}$$

In Table 3-5, the ordering combinations for the 3 vessels is shown with the corresponding values of $w_{\mu}(k)$ and $\gamma_{mu}(k)$. The last two dummy rows are unused combinations of $\gamma_{mu}(k)$. From Table 3-5 we can derive the following equations to parameterise the Max-Plus ordering binary control variables:

$$\begin{aligned} w_{0,1} &= (\bar{\gamma}_{0,1} \otimes \bar{\gamma}_{0,2}) \oplus (\gamma_{0,1} \otimes \gamma_{0,3}) \oplus (\gamma_{0,1} \otimes \gamma_{0,2}) \\ w_{0,2} &= \gamma_{0,2} \oplus (\bar{\gamma}_{0,1} \otimes \bar{\gamma}_{0,3}) \\ w_{0,3} &= (\bar{\gamma}_{0,1} \otimes \bar{\gamma}_{0,2}) \oplus (\bar{\gamma}_{0,1} \otimes \bar{\gamma}_{0,3}) \oplus (\gamma_{0,1} \otimes \gamma_{0,2}) \end{aligned} \tag{3-60}$$

As can be seen, for just 3 jobs (i.e. vessels) it does not make a lot of sense to parameterise the Max-Plus binary ordering control variables, as we for $n = 3$ the number of original Max-Plus binary ordering control variables is the same as the number of parameterised Max-Plus binary ordering control variables. We can see from Table 3-6, that for $n \geq 4$ the reduction starts to

	$\gamma_{0,1}$	$\gamma_{0,2}$	$\gamma_{0,3}$	$w_{0,1}$	$w_{0,2}$	$w_{0,3}$	$\bar{w}_{0,1}$	$\bar{w}_{0,2}$	$\bar{w}_{0,3}$	ordering
0	ε	ε	ε	ε	ε	ε	0	0	0	1 2 3
1	ε	ε	0	ε	ε	0	0	0	ε	1 3 2
2	ε	0	ε	ε	0	ε	0	ε	0	2 1 3
3	ε	0	0	ε	0	0	ε	ε	0	2 3 1
4	0	ε	ε	0	ε	ε	ε	0	0	3 1 2
5	0	ε	0	0	ε	0	ε	0	ε	3 2 1
6	0	0	ε	0	0	0	0	0	0	dummy
7	0	0	0	0	0	0	0	0	0	dummy

Table 3-5: Max-Plus truth table and corresponding permutation. (the index (k) for $w(k)$ and $\gamma(k)$ was dropped for for convenience and to improve readability)

have an effect.
To conclude, reparametrisation of the Max-Plus ordering binary control variables will reduce the number of variables required and thus reduce the complexity and computation time of the problem. Additionally, there are no infeasible choices for $w_{(\mu)}(k)$, which is especially helpful to the final MILP optimisation problem.

Table 3-6: The number of operations (i.e. vessels or jobs) N, compared with the required nominal number of Max-Plus ordering control variables w_μ and the reduced number of paramterised Max-Plus ordering control variables m^{ord}

N	$w_\mu(k)$	m^{ord}
2	1	1
3	3	3
4	6	5
5	10	7
10	45	22
15	105	41
30	435	108

3-7 Scheduling with SMPL systems

The authors in [13, 51] show that SMPL systems can be used for optimal scheduling by transforming them to MILP optimisation problems. Before we elaborate on MILP models, the concept of general Linear Programming (LP) optimisation problems should first be clear. LP is one of the most common optimisation problem formulations and was defined for the first time in [12] as:

$$\begin{aligned} \max_{x\in\mathbb{R}^n} \quad & c^T x \\ \text{subject to} \quad & Ex \leq b \end{aligned} \tag{3-61}$$

Where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $E \in \mathbb{R}^{m\times n}$ are the model parameters. The entries going into x are decision variables with $x \in \mathbb{R}^n$. The to be optimised objective function is $\max_{x\in\mathbb{R}^n} c^T x$, and $Ex \leq b$ are the inequality constraints that should be satisfied, only then a solution is feasible. If these inequality constraints are not satisfied, a solution is infeasible. If there are limits or bounds on the inputs of the decision variable x, then these should be defined in the constraints. The LP can be described as a minimisation problem by $\max\left\{c^T x\right\} = -\min\left\{-c^T x\right\}$, maximisation and minimisation is therefore considered as the same problem. The LP can be solved using the simplex algorithm [12], which guarantees an optimal solution. Common objective functions in scheduling problems are [10]:

- *Make-span* $[C_{\max}]$:
  The goal is to complete all jobs as soon as possible. Define C_j as the completion time of job j, then the objective is minimising $\max_j\{C_j\}$.
- *Lateness* $[L_{\max}]$:
  The goal is to complete all jobs as close to their deadlines as possible. Define d_j as the deadline of job j and lateness as $L_j = C_j - d_j$. Then the objective is minimising $\max_j\{L_j\}$.
- *Throughput* $[U_{\max}]$:
  The goal is to complete as much jobs on time, before their deadline d_j. Define U_j as the profit for when jobs are completed on time, $U_j = 1$ if $C_j \leq d_j$ and $U_j = 0$ otherwise.
- *Tardiness* $[T_{\max}]$:
  The goal is to minimise the delay in executing of machine operations
  $T_j = \max\{0, C_j - d_j\} = \max\{0, L_j\}$
- *Earliness* $[E_{\max}]$:
  The goal is to finish jobs before due time.
  $E_j = \max\{0, d_j - C_j\} = \max\{0, -L_j\}$
- *Convex function* $[f]$:
  The goal could be a combination of the earlier mentioned completion time, lateness, tardiness, throughput and earliness. In that case, a convex function defined by these parameters must be minimised.

One of the drawbacks of LP is that discrete decision variables cannot be modelled. When the decision variables of the to be modelled problem do require or is restricted to integers $v \in \mathbb{Z}$, such as for the earlier mentioned SMPL systems, then the optimisation problem should be formulated as a Mixed Integer Linear Programming (MILP). The fundamental difference between the two formulations is that the MILP model does allow variables to be integers. For

$$\begin{aligned} \min_{x\in\mathbb{R}^n, v\in\mathbb{Z}^k} \quad & c_x^T x + c_v^T v \\ \text{subject to} \quad & E_x x + E_v v \leq b \end{aligned} \tag{3-62}$$

With $c_x \in \mathbb{R}^n$, $c_v \in \mathbb{R}^k$, $b \in \mathbb{R}^m$, $E_x \in \mathbb{R}^{m\times n}$, and $E_v \in \mathbb{R}^{m\times k}$. Where x and v correspond to the variables of the system model in Eq 3-42. The MILP model can be written for implantation as follows:

$$\begin{aligned} \min_{x\in\mathbb{R}^n, v\in\mathbb{Z}^k} \quad & \begin{bmatrix} c_x^T & c_v^T \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} \\ \text{subject to} \quad & \begin{bmatrix} E_x & E_v \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} \leq \begin{bmatrix} b \end{bmatrix} \end{aligned} \tag{3-63}$$

The usual method for solving MILP problems is with the Branch-And-Bound algorithm, which relies on solving the linear relaxation of the MILP problem. The linear relaxation of the MILP is the LP defined by the same constraints of the MILP but where the v vector is allowed to be non-integer. This is further described in [60]. At last, the SMPL system can also be used for Model Predictive Control, which can also be recast into a MILPproblem; this is extensively shown in [50, 51]. As mentioned in Section 3-2, the Max-Plus binary control variables of the SMPL system are either 0 or ε. For the actual implementation, the infinity value ϵ can not be used. Therefore, we have to transform the Max-Plus binary control variables to conventional binary control variables.

3-8 Transformation of SMPL system to MILP model

This section will describe how SMPL systems can be transformed to MILP models. First, Section 3-8-1 will show how Max-Plus binary control variables are transformed to conventional binary control variables. Next, Section 3-8-2, will elaborate on recasting the Max-Plus equations to MILP constraints.

3-8-1 Transformation of Max-Plus variables to Conventional variables

The Max-Plus Binary control variables described throughout this chapter cannot be used for the actual implementation of a scheduling algorithm due to the earlier mentioned definition of the Max-Plus zero:

$$\epsilon \overset{\text{def}}{=} -\infty$$

Thus a method to transform Max-Plus binary control variables into conventional binary control variables will be described. First, we define an arbitrary Max-Plus binary control variable ζ_i and its adjoint $\bar{\zeta}_i$. Then the conventional binary control variable ζ_i^c will be defined as follows:

$$\zeta_i^c = \begin{cases} 0, & \text{for } \zeta_i = e \\ 1, & \text{for } \zeta_i = \varepsilon \end{cases} \tag{3-64}$$

Next, the adjoint $\bar{\zeta}_i$ can be approximated by:

$$\bar{\zeta}_i \approx \beta\,(1 - \zeta_i^c) \tag{3-65}$$

Where $\beta \ll 0$ is a large negative number. As defined in as defined in [51], we should choose $-\beta$ at least larger than the value of the largest state $x_i(k-\mu)$. Let $x^{\max} \geq \max_{i,\mu}(x_i(k-\mu))$ then we should have $\beta \leq -x^{\max}$, this will make sure the solution with the approximation and original variables will yield the same solution. With these definitions we can now transform a Max-Plus multiplication equation, Max-Plus addition equation and Max-Plus multiplication and addition equation.

3-8-2 Transformation of Max-Plus equations to MILP constraints

The Max-Plus equation to MILP constraint transformation will be described for the 3 possible Max-Plus equation types, namely, Max-Plus multiplication equation, Max-Plus addition equation and the Max-Plus multiplication and addition equation.

Max-Plus multiplication equation

Firstly, we consider a Max-Plus multiplication equation consisting of arbitrary Max-Plus binary variables of the form:

$$\zeta_0 = \zeta_1 \otimes \ldots \otimes \zeta_m \tag{3-66}$$

We can make the following statement from Max-Plus multiplication equation, only if for all Max-Plus binary control variables holds $\zeta_i = e \quad \forall i \in [1, \ldots, m]$ then $\zeta_0 = e$. If $\zeta_i \neq e \quad \forall i \in [1, \ldots, m]$ then $\zeta_0 = \varepsilon$. We can transform this Max-Plus multiplication equation to a set of conventional constraints as follows:

$$\begin{aligned}
&\zeta_0^c \geq \zeta_1^c \\
&\zeta_0^c \geq \zeta_2^c \\
&\vdots \\
&\zeta_0^c \geq \zeta_m^c \\
&\zeta_1^c + \zeta_2^c + \ldots + \zeta_m^c - \zeta_0^c \geq 0
\end{aligned} \tag{3-67}$$

Again, we can make the statement for the conventional constraints, $\zeta_i^c = 0 \quad \forall i \in [1, \ldots, m]$ then $\zeta_0^c = 0$. If $\zeta_i^c \neq 0 \quad \forall i \in [1, \ldots, m]$ then $\zeta_0^c = 1$.

Max-Plus addition equation

Secondly, we consider a Max-Plus addition equation consisting of arbitrary Max-Plus binary variables of the form:

$$\zeta_0 = \zeta_1 \oplus \ldots \oplus \zeta_m \tag{3-68}$$

We can make the following statement from Max-Plus addition equation, if for one or more Max-Plus binary control variable holds $\zeta_i = e \quad \text{for some } i \in [1, \ldots, m]$ then $\zeta_0 = e$. If $\zeta_i = \varepsilon \quad \forall i \in [1, \ldots, m]$ then $\zeta_0 = \varepsilon$. We can transform this Max-Plus addition equation to a set of conventional constraints as follows:

$$\begin{aligned}
&\zeta_0^c \leq \zeta_1^c \\
&\zeta_0^c \leq \zeta_2^c \\
&\vdots \\
&\zeta_0^c \leq \zeta_m^c \\
&\zeta_0^c \geq \zeta_1^c + \zeta_2^c + \ldots + \zeta_m^c - (m-1)
\end{aligned} \tag{3-69}$$

Again, we can make the statement for the conventional constraints, if for one or more Max-Plus binary control variable holds $\zeta_i = 1 \quad \text{for some } i \in [1, \ldots, m]$ then $\zeta_0 = 1$. If $\zeta_i = 0 \quad \forall i \in [1, \ldots, m]$ then $\zeta_0 = 0$.

Max-Plus multiplication and addition equation

At last, we consider a Max-Plus multiplication and addition equation consisting of arbitrary Max-Plus binary variables of the form:

$$\zeta_0 = \underbrace{(\zeta_{1,1} \otimes \ldots \otimes \zeta_{m,1})}_{\zeta_1} \oplus \ldots \oplus \underbrace{(\zeta_{1,n} \otimes \ldots \otimes \zeta_{m,n})}_{\zeta_n} \tag{3-70}$$

Which can be transformed to conventional constraints as follows:

$$\begin{aligned}
&\zeta_1^c \geq \zeta_{1,1}^c \\
&\vdots \\
&\zeta_1^c \geq \zeta_{m,1}^c \\
&\vdots \\
&\zeta_n^c \geq \zeta_{1,n}^c \\
&\vdots \\
&\zeta_n^c \geq \zeta_{m,n}^c \\
&\zeta_{1,1}^c + \ldots + \zeta_{m,1}^c - \zeta_1^c \geq 0 \\
&\vdots \\
&\zeta_{1,n}^c + \ldots + \zeta_{m,n}^c - \zeta_n^c \geq 0
\end{aligned} \tag{3-71}$$

The statements follow the same explanation as elaborated on in the Max-Plus multiplication and Max-Plus addition paragraphs.

3-9 Conclusion

The goal of this Chapter 3 was to lay the foundation and explain the theory for modelling the SMPL IWT systems, which could then be used later on in this research. As stated in Chapter 1, this chapter addressed the research question:

How do SMPL systems work and what is required for modelling?

The motivation behind modelling Discrete Event Systems with Max-Plus algebra is the advantage of being able to consider the resulting system as Max-Plus Linear. These MPL systems have a solid connection to conventional linear systems theory and can be analysed with some of the same tools. Additionally, there is a close relation between MPL systems and Graph Theory which allows for intuitive modelling. Moreover, there are system-theoretical analysis methods for MPL systems that can be used to find bottlenecks in DESs, for example. These MPL systems only consist of the 'oplus' and 'otimes' operators, which respectively correspond to the max-operator en plus-operator in conventional algebra. The main disadvantage of MPL systems is that only a limited number of DESs can be described as a general MPL system due to the fixed model structure. This fixed model structure means that the system dynamics will always remain the same and cannot be influenced in any way. However, the system dynamics can be changed by allowing a MPL system to switch between different modes of operations. These systems are called Switching Max-Plus Linear (SMPL) systems. In order to model these SMPL systems, we require Max-Plus Binary routing control variables and Max-Plus Binary Ordering control variables which will define the mode of the system. These SMPL systems can be used for scheduling by transforming them to Mixed Integer Linear Programming (MILP) problems. This transformation is done by first transforming the Max-Plus binary variables to conventional binary variables using an approximation for the

Max-Plus zero, after which the Max-Plus equations can be transformed to MILP constraints. In the next Chapter 4 we will show the advantage of modelling IWT systems as SMPL systems, and this is done for four different IWTcases.

Chapter 4

SMPL systems of Inland Waterway Transport systems

After the theoretical foundation is laid in the previous chapter by thoroughly describing the theory on Switching Max-Plus Linear (SMPL) systems, we will now model Inland Waterway Transport (IWT) systems as SMPL systems for four different IWT cases. As introduced in Chapter 1, this chapter will answer sub research question 3:

How to model IWT systems as SMPL systems?

Chapter 4 is structured as follows: Section 4-1 will give an introduction to the modelling approach and method. In addition, the motivation for describing IWT systems as SMPL systems will be described. Then we continue with the SMPL systems of the four IWT cases. Firstly, Section 4-2 will describe the Uni-Directional Fixed Routing case. Secondly, Section 4-3 will state the Uni-Directional Variable Routing case. Thirdly, Section 4-4 will present the Bi-Directional Fixed Routing case. Fourthly, Section 4-5 will describe the Bi-Directional Variable Routing case. At last, Section 4-6 will summarise the findings and answer the corresponding research question.

4-1 Introduction to modelling IWT systems as SMPL systems

This section will give a more detailed introduction to how and why we are going to model IWT systems as SMPL systems. First, Section 4-1-1 will set the overall modelling approach and present some definitions which will be used throughout the chapter. Next, in Section 4-1-2 the constraints for Uni-Directional Fixed Routing (UDFR) case are derived in great detail, and the method is thoroughly described. At last, Section 4-1-3 show the usefulness of modelling the IWT system as an SMPL system.

4-1-1 SMPL systems modelling approach and definitions

As mentioned in the Research Approach in Section 1-3-3 four different IWT cases where considered to build up the SMPL IWT system. This is done to slowly build up the completeness of the model and add an additional complexity with every case. However, first, as an addition to Section 3-5-1 and Section 3-5-2, we will clarify the definitions of routing and ordering in the context of IWT systems:

- *Routing* describes the route a vessel takes through the waterway network. If there is only one path from start to end, the routing is fixed. However, routing constraints are still required to capture this single route's dynamics. If there are multiple paths from start to end, the routing is variable, and the scheduler determines the optimal path. Naturally, routing constraints are required to capture the dynamics of all paths.
- *Ordering* describes the order in which vessels pass through a lock. When several vessels arrive simultaneously at a lock, the scheduler will determine the order of the queue in which vessels are allowed to enter the lock. This is what we call ordering.

Next, the SMPL modelling approach. We first start with the easiest IWT system possible, which is called the UDFR case. The UDFR case consists of two waterways with a single lock in between, and vessels are only allowed to sail a single direction (i.e. downstream). Secondly, we will continue with the slightly more complex Uni-Directional Variable Routing (UDVR) case. The UDVR case consists of two parallel UDFR systems in which vessels can be scheduled on either route, but still sail only in a single direction (i.e. downstream). Thirdly, the Bi-Directional Fixed Routing (BDFR) case will be expressed as a SMPL system. The BDFR case closely resembles the UDFR case; however, vessels are allowed to sail in both directions (i.e. downstream and upstream). At last, naturally, the same goes for the Bi-Directional Variable Routing (BDVR) case, which is the same as the UDVR case but with both sailing directions. There is a figure of the topology graph describing the waterway network for each case, as presentend in Section 3-4.
To clarify, the directional component of the four cases can be understood as follows:

- For the *Uni-Directional* cases, vessels are only allowed to sail in a single direction through the network (i.e. downstream), like a one-way street.
- For the *Bi-Directional* cases, vessels are allowed to sail in both directions through the network (i.e. both downstream or upstream), like a two-way street.

For the Bi-Directional cases, the vessels will have a pre-specified starting location and thus a pre-specified sailing direction. It does not make sense to let the scheduler determine the starting location of a vessel, as in real-life scenarios, this cannot be controlled either. Obviously, we cannot control from which locations a vessel will depart.
Next, to clarify, the routing component of the four cases can be understood as follows:

- For the *Fixed Routing* cases, the scheduler does not actively determine the route vessels take as there is only one path to go from start to end.

- For the *Variable Routing* cases, the scheduler does actively determine the route vessels take as there are multiple paths to go from start to end.

To summarise, the four cases, the problems they deal with, and what they describe of the IWT system:

- The *UDFR case* deals with determining the order in which vessels can enter a lock with only vessels coming from a single direction. So a vessel arriving at a lock from the downstream direction is only dependent on other vessels arriving from the downstream direction. No vessels are sailing upstream. Therefore, this ordering dependency does not need to be taken into account, creating a relatively simple case.
- The *UDVR case* deals with determining the order in which vessels can enter locks with only vessels coming from a single direction, and with determining the route vessels should take. This is an extension to the UDFR case scheduler by adding the Variable Routing feature, which determines the best route a vessel can take based on what the other vessels are doing, creating a more complex case than the UDFR.
- The *BDFR case*: Deals with determining the order in which vessels can enter a lock with vessels coming from both sides. So a vessel arriving at a lock from the downstream direction is not only dependent on other vessels arriving from the downstream direction, but also dependent on vessel who are arriving at the other side of the lock that are coming from the upstream direction.
- The *BDVR case*: deals with determining the order in which vessels can enter locks with vessels coming from both sides, and with determining the route vessels should take. This is an extension to the BDFR case scheduler by adding the Variable Routing feature, which determines the best route a vessel can take based on what the other vessels are doing, creating a more complex case than the BIFR. This is the most generic case and should allow all other possible networks to be created as well.

For these four cases, we will first present the setup and assumptions of the case. Next, the Max-Plus Routing and Ordering equations are derived, after which the SMPL systems are completed. This chapter will refer to Max-Plus Routing equations and Max-Plus Ordering equations, which should formally be Max-Plus Routing constraints and Max-Plus Ordering constraints. However, as these Max-Plus Routing constraints and Max-Plus Ordering constraints are later combined in a system description to form an equation, they will be denoted as Max-Plus Routing equations and Max-Plus Ordering equations. At last, after having introduced the SMPL systems modelling approach, the next section will first elaborate on the SMPL systems modelling method by deriving the constraints for the UDFR case in great detail.

4-1-2 SMPL systems modelling method

After having introduced important definitions and described the approach used for modelling the SMPL systems, in the previous section, we will continue with showing the method for modelling IWT systems as SMPL systems in this section. Just as in Section 3-5-1, this will be done by deriving the constraints for the UDFR case in great detail. Figure 4-1 shows the schematic top view of the IWT system of the UDFR case. As mentioned, the UDFR case consists of two waterways with a single lock in between, and vessels are only allowed to sail in a single direction.

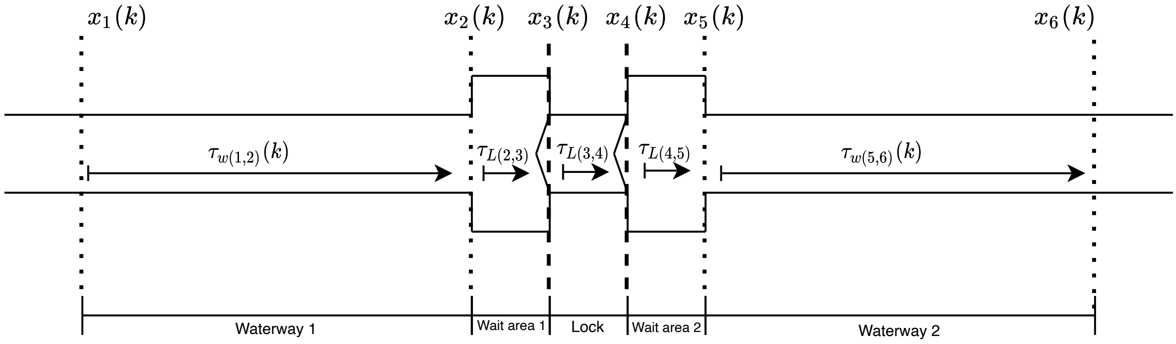


Figure 4-1: Schematic top view of the IWT system of the UDFR case

In Figure 4-1, we have defined $x_i(k)$ as the time instant at which event i happens for the k-th vessel, then we define the following events and descriptions shown in Table 4-1.

Table 4-1: Definitions of state variables $x_i(k)$ of Figure 4-1

State event variable	Description
$\mathrm{x}_1(k)$	Time instant when vessel k enters waterway 1
$\mathrm{x}_2(k)$	Time instant when vessel k enters waiting area 1 and leaves waterway 1
$\mathrm{x}_3(k)$	Time instant when vessel k enters the lock and leaves waiting area 1
$\mathrm{x}_4(k)$	Time instant when vessel k enters waiting area 2 and leaves the lock
$\mathrm{x}_5(k)$	Time instant when vessel k enters waterway 2 and leaves waiting area 2
$\mathrm{x}_6(k)$	Time instant when vessel k leaves waterway 2

In Figure 4-1, we have defined sailing times $\tau_{w(i,j)}(k)$ and lock operation times $\tau_{L(i,j)}$ as described in in Table 4-2.

Table 4-2: Definitions of sailing times $\tau_{w(i,j)}(k)$ and lock operation times $\tau_{L(i,j)}$ of Figure 4-1

Timings	Description
$\tau_{w(1,2)}(k)$	Sailing time of vessel k from the start of waterway 1 to the end of waterway 1
$\tau_{L(2,3)}$	Operation time from the start of waiting area 1 to the end of waiting area 1
$\tau_{L(3,4)}$	Lock operation (i.e. drainage) time to level the water to the other side of the lock
$\tau_{L(4,5)}$	Operation time from the start of waiting area 2 to the end of waiting area 2
$\tau_{w(5,6)}(k)$	Sailing time of vessel k from the start of waterway 2 to the end of waterway 2

At last, $\mathrm{u}_1(\mathrm{k})$ is the time when vessel k departs and enters waterway 1.
Since this is the first time that, to the best knowledge of the author, IWT systems are described as SMPL systems, some simplifications have been made to keep the research manageable and not directly overly complex. Therefore, we summarise the simplifications and made assumptions:

- We assume the locks can only process one vessel at a time
- We assume the locks to start processing directly when a vessel enters the lock
- We assume the lock operation timings $\tau_{L(2,3)}$ $\tau_{L(3,4)}$ $\tau_{L(4,5)}$ to be independent of the vessel and just lock specific (in reality, this is vessel dependent)
- We assume to vessel sailing speed to be constant on the waterways
- We assume that vessels cannot overtake inside locks. This is a logical consequence when locks can only process one vessel at a time.
- We allow for vessel overtaking on all waterways and waiting areas
- We cannot determine the departure time of vessels $u_i(k)$, which are just arbitrary inputs

Next, Figure 4-2 shows six figures with the schematic side view of the IWT system of the UDFR case, with vessel k sailing through the network. Each subfigure shows the next step of vessel k moving through the waterway network. Moreover, the subfigures also show the Max-Plus inequality that describes the dynamics of the vessel movement.
Figure 4-2a, shows vessel k departing at time instant $u_1(k)$ and entering waterway 1 at time instant $x_1(k)$. The departure time $u_1(k)$ should always be greater or equal than the time of entering waterway 1 $x_1(k)$. Thus:

$$x_1(k) \geq u_1(k) \tag{4-1}$$

Which is the same in Max-Plus algebra.
Next, Figure 4-2b, shows vessel k sailing from the start of waterway 1 to the end of waterway 1 and entering waiting area 1. The time of entering waiting area 1, $x_2(k)$, should always be greater or equal than the time of entering waterway 1, $x_1(k)$, plus the sailing time from $x_1(k)$ to $x_2(k)$, which is $\tau_{w(1,2)}(k)$. Thus:

$$x_2(k) \geq x_1(k) + \tau_{w(1,2)}(k) \tag{4-2}$$

Which is in Max-Plus algebra:

$$x_2(k) \geq x_1(k) \otimes \tau_{w(1,2)}(k) \tag{4-3}$$

Next, Figure 4-2c, shows vessel k sailing from the start of waiting area 1 to the end of waiting area 1 and entering the lock. The time of entering the lock $x_3(k)$ should always be greater or equal than the time of entering waiting area 1 $x_2(k)$ plus the operation time from $x_2(k)$ to $x_3(k)$, which is $\tau_{L(2,3)}$. Thus:

$$x_3(k) \geq x_2(k) + \tau_{L(2,3)} \tag{4-4}$$

Which is in Max-Plus algebra:

$$x_3(k) \geq x_2(k) \otimes \tau_{L(2,3)} \tag{4-5}$$

Next, Figure 4-2d, shows vessel k being moved by the lock from the low water level to the high water level, and entering waiting area 2. The time of entering waiting area 2 $x_4(k)$ should always be greater or equal than the time of entering the lock $x_3(k)$ plus the lock operation time from $x_3(k)$ to $x_4(k)$, which is $\tau_{L(3,4)}$. Thus:

$$x_4(k) \geq x_3(k) + \tau_{L(3,4)} \tag{4-6}$$

Which is in Max-Plus algebra:

$$x_4(k) \geq x_3(k) \otimes \tau_{L(3,4)} \tag{4-7}$$

Next, Figure 4-2e, shows vessel k sailing from the start of waiting area 2 to the end of waiting area 2 and entering waterway 2. The time of entering waterway 2 $x_5(k)$ should always be greater or equal than the time of entering waiting area 2 $x_4(k)$ plus the sailing time from $x_4(k)$ to $x_5(k)$, which is $\tau_{L(4,5)}$. Thus:

$$x_5(k) \geq x_4(k) + \tau_{L(4,5)} \tag{4-8}$$

Which is in Max-Plus algebra:

$$x_5(k) \geq x_4(k) \otimes \tau_{L(4,5)} \tag{4-9}$$

At last, Figure 4-2f, shows vessel k sailing from the end of waiting area 2 and the start of waterway 2 to the end of waterway 2. The time of leaving waterway 2 $x_6(k)$ should always be greater or equal than the time of entering waterway 2 $x_5(k)$ plus the sailing time from $x_5(k)$ to $x_6(k)$, which is $\tau_{w(5,6)}(k)$. Thus:

$$x_5(k) \geq x_4(k) + \tau_{w(5,6)}(k) \tag{4-10}$$

Which is in Max-Plus algebra:

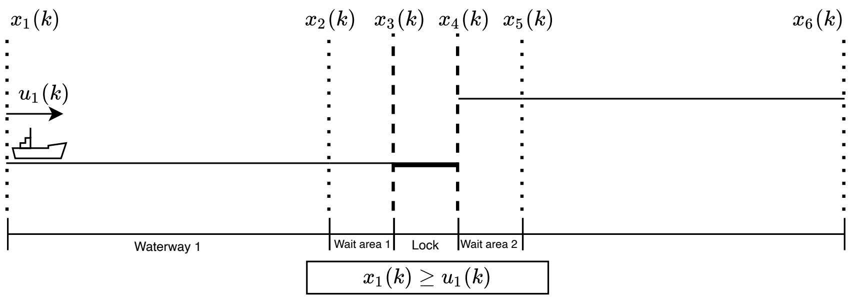
$$x_5(k) \geq x_4(k) \otimes \tau_{w(5,6)}(k) \tag{4-11}$$

Which completes describing the six subfigures of Figure 4-2. For Figure 4-2 to be completely accurate, the vessel icons should have been drawn directly on the dotted and dashed lines. However, as this would make the figure confusing and more difficult to read, it was decided not to do this. This way, the information comes across better. The Max-Plus inequalities mentioned do not yet clearly show the advantage of modelling the IWT system as a SMPL system; this will be done in the next section.

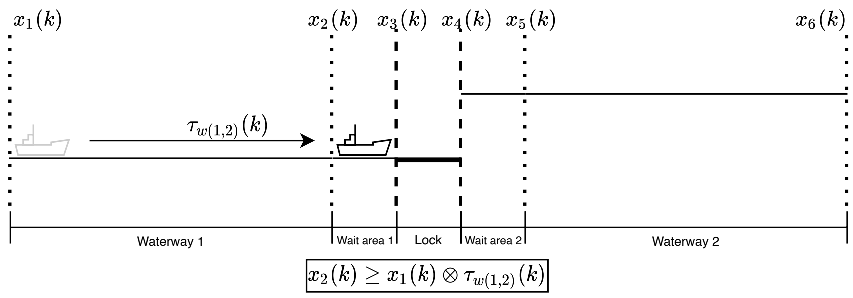
4-1-3 Motivation for using SMPL systems to model IWT systems

The previous sections introduced important definitions, described the approach, and described the method for modelling the SMPL systems. Next, we will continue with showing the actual motivation behind modelling IWT systems as SMPL systems in this section. This order was chosen rather than starting with the motivation, because the motivation will be better understood with the definitions and method mentioned earlier. Just as in Section 3-1 and the example in Appendix A-1, we will show the appearance of the in conventional algebra nonlinear Max-operator when describing the IWT system as Discrete Event System (DES). As shown in Section 4-1-2, the inequalities that describe the routing dynamics of the vessel could just as easily be described in conventional algebra. They do not yet show the advantage of modelling the IWT system as SMPL system.

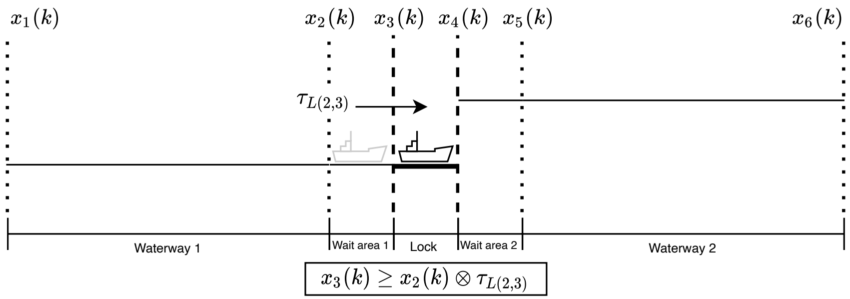
However, let us look at a slightly different example. We consider the same UDFR case of Figure 4-1 with two waterways with a single lock in between, and vessels are only allowed to sail in a single direction. Next, we introduce an additional vessel $k-1$, which is sailing in front of vessel k. This can be seen in Figure 4-3, the blue vessel is $k-1$ and the red vessel is



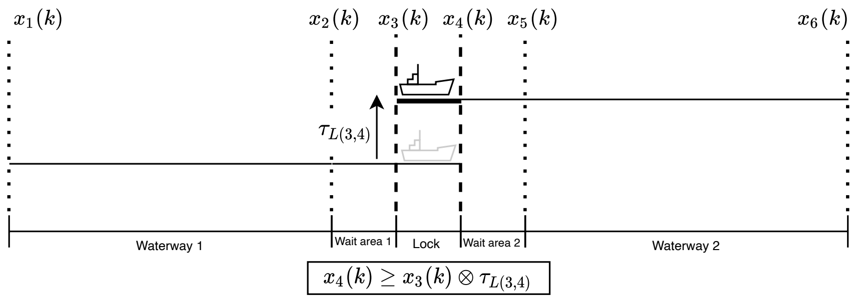
(a) Vessel k departs at $u_1(k)$ and enters the network at x_1



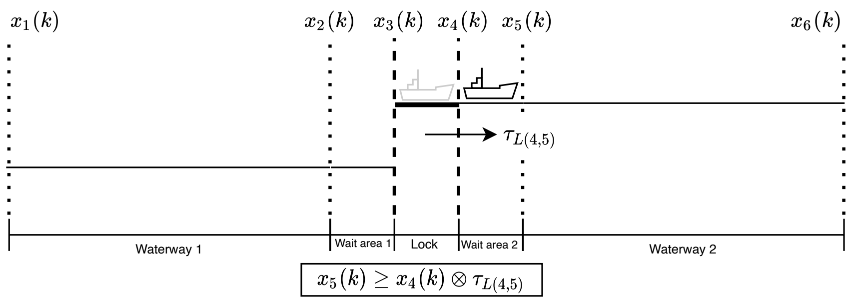
(b) Vessel k going from event $x_1(k)$ to event $x_2(k)$



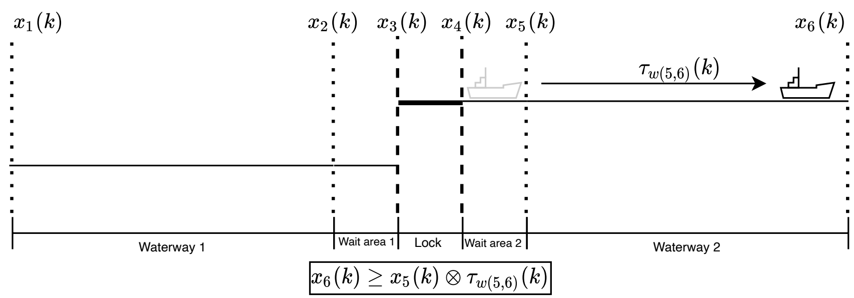
(c) Vessel k going from event $x_2(k)$ to event $x_3(k)$



(d) Vessel k going from $x_4(k)$ to $x_5(k)$



(e) Vessel k going from $x_5(k)$ to $x_6(k)$



(f) Vessel k going from $x_5(k)$ to $x_6(k)$

Figure 4-2: Schematic side view of the IWT system for the UDFR case, with vessel k sailing through the network with a consecutive event step in each sub-figure

k. As vessel $k-1$ sails in front of vessel k, we can see in the figure that it went through the lock first.
Now we have the following situation; vessel $k-1$ is just exiting the lock (recall from Section 4-1-2 that this is described by $x_4(k-1)$) and vessel k wants to enter the lock (recall from Section 4-1-2 that this is described by $x_3(k)$). However, the water level is still high due to the processing of vessel $k-1$, thus vessel k cannot enter yet and has to wait until the water is drained down again. Thus, vessel k can only enter the lock when vessel $k-1$ has left the lock plus the levelling of the lock to the lower water level. This can be described with the following inequality:

$$x_3(k) \geq x_4(k-1) + \tau_{L(3,4)} \tag{4-12}$$

And in Max-Plus algebra:

$$x_3(k) \geq x_4(k-1) \otimes \tau_{L(3,4)} \tag{4-13}$$

We could, of course, also have the situation that vessel $k-1$ arrived way earlier at the lock and that the lock is already drained down again to the lower level, but vessel k has not arrived at the lock yet. It that case, the moment when vessel k can enter the lock is described by the earlier in Section 4-1-2 mentioned Eq (4-4) and Eq (4-5). To recap:

$$x_3(k) \geq x_2(k) + \tau_{L(2,3)} \tag{4-14}$$

Which is in Max-Plus algebra:

$$x_3(k) \geq x_2(k) \otimes \tau_{L(2,3)} \tag{4-15}$$

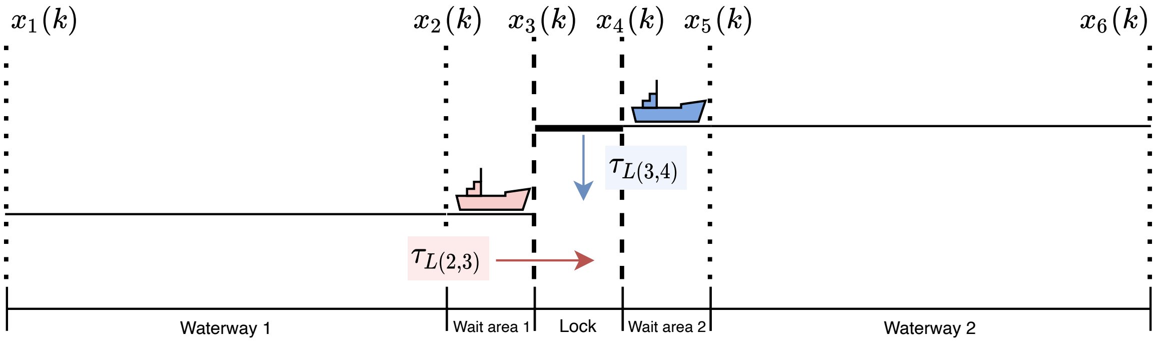
Combining the conventional algebra inequalities for describing the dynamics of vessel k Eq (4-12) and Eq (4-14), we get:

$$x_3(k) = \max\Big(x_2(k) + \tau_{L(2,3)}, x_4(k-1) + \tau_{L(3,4)}\Big) \tag{4-16}$$

In words, the time when vessel k can enter the lock is equal to the maximum of the arrival time of vessel k at $x_3(k)$ or the exiting of vessel $k-1$ at $x_4(k-1)$ plus $\tau_{L(3,4)}$. This is also described in the first equation in Figure 4-3. We can see that describing Eq (4-16) in conventional algebra results in the nonlinear Max-operator, just as explained in Section 3-1 and the example presented in Appendix A-1. If we write Eq (4-16) using Max-Plus algebra, we get:

$$x_3(k) = x_2(k) \otimes \tau_{w(2,3)} \oplus x_4(k-1) \otimes \tau_{L(3,4)} \tag{4-17}$$

This way we create a Max-Plus Linear (MPL) system, as described in Section 3-1 and Section 3-3. This has the advantage that we can analyse the system by using conventional linear system theory tools. Using MPL systems, we can understand the behaviour of the IWT system, which can be used to obtain information about the event evolution. At last, there are system-theoretical analysis methods for MPL systems [22], which can be used to find bottlenecks in the IWT system, for example. Doing these analysis will not be the scope of this research, but will be interesting for future research.



$$(1) \quad x_3(k) = \max(x_2(k) + \tau_{L(2,3)}, x_4(k-1) + \tau_{L(3,4)})$$

$$(2) \quad x_3(k) = x_2(k) \otimes \tau_{w(2,3)} \oplus x_4(k-1) \otimes \tau_{L(3,4)}$$

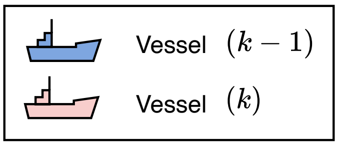


Figure 4-3: Schematic side view of the IWT system for the UDFR case, with red vessel k and blue vessel $k-1$ sailing through the network. Red vessel k has to wait for blue vessel $k-1$ until it can enter the lock; this interaction is denoted in conventional algebra in nonlinear Equation (1) and denoted in Max-Plus algebra in Max-Plus Linear Equation (2).

To summarise, in Figure 4-3, the red part of the equation describes the routing dynamics of vessel k and the blue part of the equation describes interaction and dependency of vessel k on $k-1$, which we will later on call ordering. The Max-operator appears when we combine the routing and ordering dynamics of vessels.
After showing the motivation behind modelling IWT systems as SMPL systems, we continue with the first case, the UDFR case. There will be some overlap with the constraints derived in this section; however, they will be complete and not derived in such great detail. Moreover, this will not be done for the other three cases either, making the thesis more concise.

4-2 Uni-Directional Fixed Routing case

Firstly, this section will present the SMPL system of the IWT system, which will make it possible to order vessels at a single lock with vessels coming from a single direction. This section will answer the research question:

How to model the Uni-Directional Fixed Routing Inland Waterway Transport system as a Switching Max-Plus Linear system?

Section 4-2 is structured as follows: First, the set up and assumptions for the Uni-Directional Fixed Routing case are given in Section 4-2-1. Next, the Max-Plus routing and ordering equations are given in Section 4-2-2 and Section 4-2-3, respectively. At last, the equations are combined and the UDFR IWT SMPL system is completed in Section 4-2-4.

4-2-1 UDFR Setup

For the UDFR case we define the variables shown in Table 4-3, which include most of the variables as presented in Section 4-1 plus the arrival deadline $a_6(k)$.

Table 4-3: Variables for the UDFR case

Variables	Description
$x_i(k)$	Represents the time instant at which event i happens for the k-th vessel
$\tau_{w(i,j)}(k)$	Represents the constant sailing time of vessel k on the waterway from node i to node j
$\tau_{L(i,j)}$	Represents the time of the lock operation from node i to node j
$u_1(k)$	Represents the time instant at which the k-th vessel departs at $x_1(k)$
$a_6(k)$	Represents the time instant at which the k-th vessels should arrive at $x_6(k)$

For example, $\tau_{w(1,2)}(k)$ is the sailing time of vessel k on waterway $(1,2)$. Moreover, $\tau_{L(2,3)}$ is the time for a vessel moving from the lock waiting area into the actual lock and $\tau_{L(3,4)}$ is the lock drainage time. Note that we have assumed that the lock operation timings $\tau_{L(i,j)}$ are not vessel dependent.
Next, we define the topology graph of the UDFR case waterway network in Figure 4-4, consisting of two waterways and a lock. This topology graph is based on the schematic top view of Figure 4-1 and the schematic side view of Figure 4-2. The schematic top view and side view will not be shown for all cases as this would be unnecessarily extensive for some cases. All vessels will start on the left at $x_1(k)$ and will travel over the first waterway through the lock second waterway to $x_6(k)$.

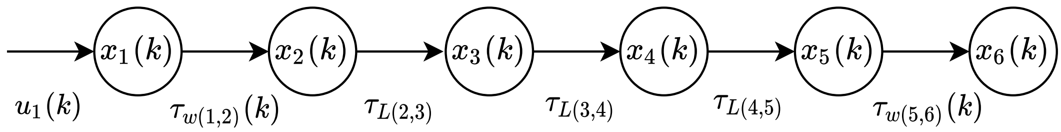


Figure 4-4: Topology graph of the IWT system for the Uni-Directional Fixed Routing case

4-2-2 Routing equations

Following the method described in Section 3-5-1, we can derive the Max-Plus Routing equations. The Max-Plus Routing equations following from the topology graph in Figure 4-4 are described in Eq.(4-18):

$$\begin{aligned}
x_1(k) &\geq u_1(k)\\
x_2(k) &\geq x_1(k) \otimes \tau_{w(1,2)}(k)\\
x_3(k) &\geq x_2(k) \otimes \tau_{L(2,3)}\\
x_4(k) &\geq x_3(k) \otimes \tau_{L(3,4)}\\
x_5(k) &\geq x_4(k) \otimes \tau_{L(4,5)}\\
x_6(k) &\geq x_5(k) \otimes \tau_{w(1,2)}(k)
\end{aligned} \tag{4-18}$$

These can written in the general implicit MPL form of Eq. (3-17):

$$x(k) \geq A_0^{\text{job}}(k) \otimes x(k) \oplus B\,(k) \otimes u(k) \tag{4-19}$$

with

$$x(k) \geq A_0^{\text{job}}\Big(\tau_{i,j}(k)\Big) \otimes x(k) \oplus B\,(k) \otimes u(k) \tag{4-20}$$

Which gives:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \end{bmatrix} \geq \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \tau_{w(1,2)}(k) & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \tau_{L(2,3)} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \tau_{L(3,4)} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \tau_{L(4,5)} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \tau_{w(5,6)}(k) & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \end{bmatrix} \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix} \otimes u(k) \tag{4-21}$$

Important remark, only for this UDFR case the matrix notation will be presented entirely worked out. For larger systems, the matrix-notation becomes disorganised and not uninformative. However, the method remains the same.

4-2-3 Ordering equations

Following the method described in Section 3-5-2, we can derive the Max-Plus ordering equations. To define the Max-Plus ordering constraints, we first define the binary control variable $w_{i,\mu}(k-\mu)$ as follows:

$$w_{i,\mu}(k-\mu) = \begin{cases} e, & \text{if operation } i \text{ for vessel } k-\mu \text{ precedes} \\ & \text{operation } i \text{ for vessel } k \\ \varepsilon, & \text{if operation } i \text{ for vessel } k \text{ precedes} \\ & \text{operation } i \text{ for vessel } k-\mu \end{cases} \tag{4-22}$$

Then the adjoint variable $\bar{w}_{i,\mu}(k-\mu)$ is defined as:

$$\bar{w}_{i,\mu}(k-\mu) = \begin{cases} e, & \text{if } w_{i,\mu}(k-\mu) = \varepsilon \\ \varepsilon, & \text{if } w_{i,\mu}(k-\mu) = e \end{cases} \tag{4-23}$$

For every node in the topology graph where the order of the vessels is fixed two ordering constraints must be introduced. The constraints for node i or state x_i are of the form shown in Eq. (4-24) and Eq. (4-25):

$$x_i(k) \geq x_i(k-\mu) \otimes \tau_{i,j}(k-\mu) \otimes w_{i,\mu}(k-\mu) \tag{4-24}$$

To clarify in words, Eq. (4-24) states the following:

'The time of event x_i for vessel k is greater or equal to the time of event x_i for vessel $k-\mu$ plus some operation time $\tau_{i,j}(k-\mu)$ (e.g. lock water levelling) of the vessel $k-\mu$. Thus event x_i happens first for vessel $k-\mu$ and thus precedes vessel k'

$$x_i(k-\mu) \geq x_i(k) \otimes \tau_{i,j}(k) \otimes \bar{w}_{i,\mu}(k-\mu) \tag{4-25}$$

To clarify in words, Eq. (4-25) states the following:

'The time of event x_i for vessel $k-\mu$ is greater or equal to the time of event x_i for vessel k plus some operation time $\tau_{i,j}(k)$ (e.g. lock water levelling) of the vessel k. Thus event x_i happens first for vessel k and thus precedes vessel $k-\mu$'

The Max-Plus binary control variables $w_{i,\mu}(k-\mu)$ and $\bar{w}_{i,\mu}(k-\mu)$ as defined in Eq. (4-22) and Eq. (4-23) respectively, will ensure that either equation Eq. (4-24) is active or Eq. (4-25) is active.

As introduced in Section 4-1, overtaking on the waterways is allowed and only one vessel can enter a lock at a time. Therefore, we only have to determine the order of the vessels at $x_3(k)$ thus at node 3. Therefore Eq. (4-22) and Eq. (4-23) will become:

$$x_3(k) \geq x_4(k-\mu) \otimes \tau_{L(3,4)} \otimes w_{3,\mu}(k-\mu) \tag{4-26a}$$

$$x_3(k-\mu) \geq x_4(k) \otimes \tau_{L(3,4)} \otimes \bar{w}_{3,\mu}(k-\mu) \tag{4-26b}$$

for $\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$ and $k \in \mathbb{Z}$. To note, as mentioned in Section 3-6-2, the larger the $\mu_{\max}$, the more ordering combinations there are in the system. Thus, the scheduler must determine a larger number of ordering control variables. It will be more computationally demanding to consider all these possible, which will result in the slower finding of a solution by the scheduler. At last, Eq 4-26a and Eq 4-26b can be written in the general SMPL form of Eq. (4-27):

$$x(k) \geq A_{\mu}^{\text{ord}}\Big(\tau_{i,j}(k), w_{3,\mu}(k-\mu)\Big) \otimes x(k-\mu) \tag{4-27}$$

Which gives:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \end{bmatrix} \geq \underbrace{\begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \tau_{L(3,4)} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}}_{A_{\mu}^{\text{ord}}} \otimes \begin{bmatrix} x_1(k-\mu) \\ x_2(k-\mu) \\ x_3(k-\mu) \\ x_4(k-\mu) \\ x_5(k-\mu) \\ x_6(k-\mu) \end{bmatrix} \tag{4-28}$$

4-2-4 Complete SMPL UDFR system

To complete the model the only required additional equations is a constraint on the arrival time. We define $a_i(k)$ as the arrival deadline for vessel k as node i. This constraint will be of the following form:

$$-x_6(k-\mu) \geq -a_6(k) \tag{4-29}$$

Now that conditions have been derived for $x_i(k)$ that belong to the different routing and ordering control variables a complete set of scheduling decision variables can be made, as follows:

- Ordering: $w_{\mu}(k-\mu)$, $\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$

The decision variables can be stacked into one vector:

$$v(k) = \begin{bmatrix} w_{\mu_{\min}}(k-\mu) \\ \vdots \\ w_{\mu_{\max}}(k-\mu) \end{bmatrix} \in (\mathbb{B}_\varepsilon)^{L_{\text{tot}}} \tag{4-30}$$

with L_{tot} the total number of scheduling variables. Now the scheduling model can be written as:

$$x(k) = \bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu(v(k), k) \otimes x(k-\mu) \tag{4-31}$$

where:

$$x(k) = A_0^{\text{job}}(\tau_{ij}(k), k) \otimes x(k) \oplus \left[\bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu^{\text{ord}}(w_\mu(k), k) \otimes x(k-\mu) \right] \oplus B(k) \otimes u(k) \tag{4-32}$$

with $\tau_{ij} = \{\tau_{w(i,j)}(k), \tau_{L(i,j)}\}$. By choosing a specific control vector $v(k)$ the system switches between different modes of operation, it is thereby a SMPL system.

The model can be completed by grouping the Max-Plus

$$x(k) = A_0^{\text{job}}(k) \otimes x(k) \oplus A_{\mu_{min}}(k) \otimes x(k-\mu_{min}) \oplus \cdots \oplus A_{\mu_{\max}}(k) \otimes x(k-\mu_{\max}) \oplus B(k) \otimes u(k) \tag{4-33}$$

4-3 Uni-Directional Variable Routing case

Secondly, this section will present the SMPL system of the IWT system, which will make it possible to allocate vessels to a route. This section will answer the research question:

How to model the Uni-Directional Variable Routing Inland Waterway Transport system as a Switching Max-Plus Linear system?

As mentioned in Section 4-1, the UDVR case deals with determining the order in which vessels can enter locks with only vessels coming from a single direction and with determining the route vessels should take. Thus the additional problem the UDVR case solves, in comparison with the UDFR case, is allocating the vessels to a route such that the cumulative overall vessel arrival times are minimised.
Section 4-3 is structured as follows: First, the set up and assumptions for the Uni-Directional Variable Routing case are given in Section 4-3-1. Next, the Max-Plus routing and ordering equations are given in Section 4-3-2 and Section 4-3-3, respectively. At last, the equations are combined and the UDVR IWT SMPL system is completed in Section 4-3-4.

4-3-1 UDVR Setup

We define the same variables and assumptions as described in Section 4-2-1. Next we define the waterway network shown in Figure 4-5, which consists basically of two parallel systems as the waterway network described in Section 4-2-1 in Figure 4-4. Thus, the network consists of two routes each consisting of three waterways and a lock. All vessels will start at the top at $x_1(k)$ and will travel over the left route 1 or right route 2 over the first waterway through the lock then over the second and third waterway to $x_{12}(k)$.

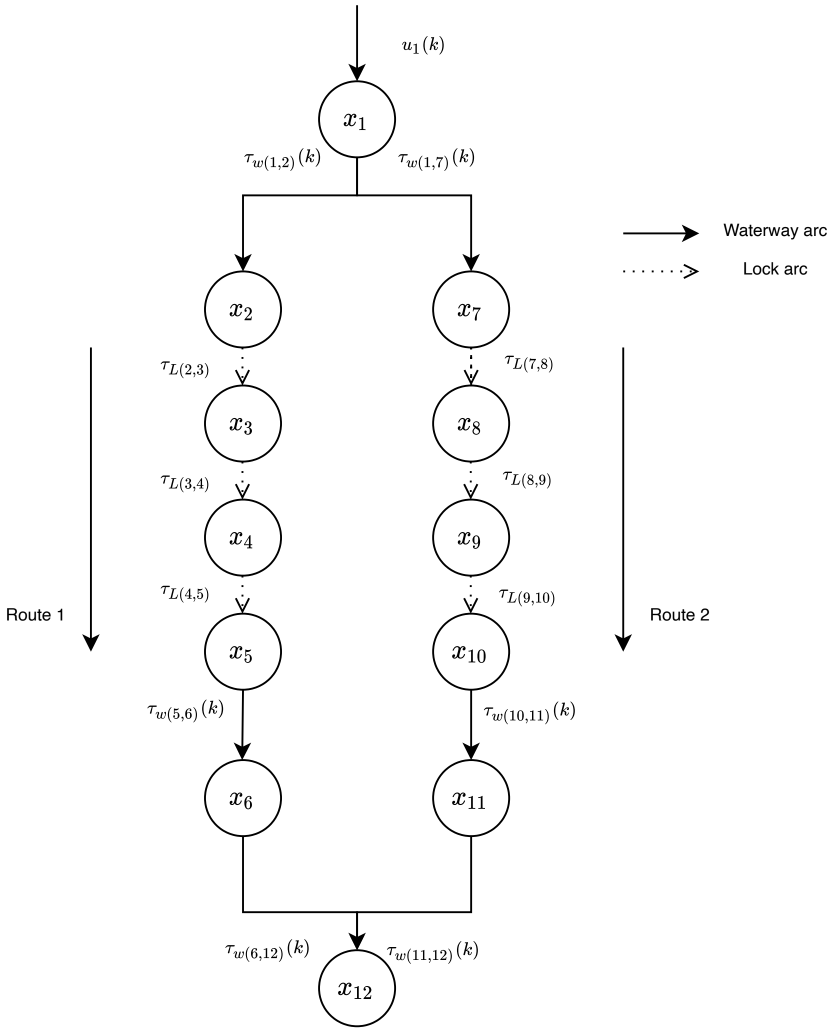


Figure 4-5: Topology graph of the IWT system for the Uni-Directional Variable Routing case

4-3-2 Routing equations

Following the method described in Section 3-5-1, we can derive the Max-Plus Routing equations. We follow the exact same procedure as was done in Section 4-2-2. However, to describe the variable routing of the waterway network presented in Figure 4-5, we will need to introduce an additional variable that will activate either route 1 or route 2 for a particular vessel k. As vessel can obviously only take one route at a time. Therefore we will introduce the Max-Plus binary control variable $s_{ij}(k)$, which has the following definition:

$$s_{ij}(k) = \begin{cases} e, & \text{if arc } (i,j) \text{ is used by vessel } k \\ \varepsilon, & \text{if arc } (i,j) \text{ is not used by vessel } k \end{cases} \tag{4-34}$$

Then to model the routing of Figure 4-5, we require 13 Max-Plus inequality constraints shown in Eq. (4-35):

$$\begin{aligned}
x_1(k) &\geq u_1(k) \otimes s_{u,i}(k) \\
x_2(k) &\geq x_1(k) \otimes \tau_{w(1,2)}(k) \otimes s_{1,2}(k) \\
x_3(k) &\geq x_2(k) \otimes \tau_{L(2,3)} \otimes s_{2,3}(k) \\
x_4(k) &\geq x_3(k) \otimes \tau_{L(3,4)} \otimes s_{3,4}(k) \\
x_5(k) &\geq x_4(k) \otimes \tau_{L(4,5)} \otimes s_{4,5}(k) \\
x_6(k) &\geq x_5(k) \otimes \tau_{w(5,6)}(k) \otimes s_{5,6}(k) \\
x_7(k) &\geq x_1(k) \otimes \tau_{w(1,7)}(k) \otimes s_{1,7}(k) \\
x_8(k) &\geq x_7(k) \otimes \tau_{L(7,8)} \otimes s_{7,8}(k) \\
x_9(k) &\geq x_8(k) \otimes \tau_{L(8,9)} \otimes s_{8,9}(k) \\
x_{10}(k) &\geq x_9(k) \otimes \tau_{L(9,10)} \otimes s_{9,10}(k) \\
x_{11}(k) &\geq x_{10}(k) \otimes \tau_{w(10,11)}(k) \otimes s_{10,11}(k) \\
x_{12}(k) &\geq x_6(k) \otimes \tau_{w(6,12)}(k) \otimes s_{6,12}(k) \\
x_{12}(k) &\geq x_{11}(k) \otimes \tau_{w(11,12)}(k) \otimes s_{11,12}(k)
\end{aligned} \tag{4-35}$$

As there are only two possible routes in the waterway network, we can reduce the number of required Max-Plus binary control variables to two. Then, for the left route we have, $s_{1,2}(k) = s_{2,3}(k) = s_{3,4}(k) = s_{4,5}(k) = s_{5,6}(k) = s_{6,12}(k) = s_1(k)$, and for the right route 2 we have, $s_{1,7}(k) = s_{7,8}(k) = s_{8,9}(k) = s_{9,10}(k) = s_{10,11}(k) = s_{11,12}(k) = s_2(k)$. This simplification is shown in Eq (4-36) for *route 1*:

$$\begin{aligned}
x_1(k) &\geq u(k) \\
x_2(k) &\geq x_1(k) \otimes \tau_{w(1,2)}(k) \otimes s_1(k) \\
x_3(k) &\geq x_2(k) \otimes \tau_{L(2,3)} \otimes s_1(k) \\
x_4(k) &\geq x_3(k) \otimes \tau_{L(3,4)} \otimes s_1(k) \\
x_5(k) &\geq x_4(k) \otimes \tau_{L(4,5)} \otimes s_1(k) \\
x_6(k) &\geq x_5(k) \otimes \tau_{w(5,6)}(k) \otimes s_1(k) \\
x_{12}(k) &\geq x_6(k) \otimes \tau_{w(6,12)}(k) \otimes s_1(k)
\end{aligned} \tag{4-36}$$

And this simplification is shown in Eq (4-37) for *route 2*:

$$\begin{aligned}
x_1(k) &\geq u(k) \\
x_7(k) &\geq x_1(k) \otimes \tau_{w(1,7)}(k) \otimes s_2(k) \\
x_8(k) &\geq x_7(k) \otimes \tau_{L(7,8)} \otimes s_2(k) \\
x_9(k) &\geq x_8(k) \otimes \tau_{L(8,9)} \otimes s_2(k) \\
x_{10}(k) &\geq x_9(k) \otimes \tau_{L(9,10)} \otimes s_2(k) \\
x_{11}(k) &\geq x_{10}(k) \otimes \tau_{w(10,11)}(k) \otimes s_2(k) \\
x_{12}(k) &\geq x_{11}(k) \otimes \tau_{w(11,12)}(k) \otimes s_2(k)
\end{aligned} \tag{4-37}$$

Note that for this case we could say that $s_2(k)$ should be the adjoint of $s_1(k)$, as we only have two routes and a vessel should choose either one. However, this will no longer be the case if

there are more than two routes, which is why it was decided not to model it this way. Not doing this ensures that the the model can be extended more easily to ≥ 2 routes.
At last, the equations of Eq. (4-36) and Eq. (4-37) can written in the general implicit SMPL form of Eq. (3-24):

$$x(k) \geq A_0^{\text{job}}\left(s_{i,j}(k), \tau_{i,j}(k)\right) \otimes x(k) \oplus B(k) \otimes u(k) \tag{4-38}$$

With

$$A_0^{\text{job}}\left(s_{i,j}(k), \tau_{i,j}(k)\right) = S_{i,j}(k) \odot T_{i,j}(k) \tag{4-39}$$

4-3-3 Ordering equations

Following the method described in Section 3-5-2, we can derive the Max-Plus ordering equations. Defining the Max-Plus ordering equations proceeds in the same way as was done in Section 4-2-3 for the uni-directional fixed routing case, however, with two additional points of interest:

- We have ordering in an additional node, not just node 3 but also in node 8, because we have two locks.
- We only have ordering when between vessels when vessels are using the same route, therefore we should use the earlier defined Max-Plus binary routing variable $s_{i,j}(k)$ to ensure that the ordering constraints are only active when this is the case.

Again, first, we define the binary control variable $w_{i,\mu}(k-\mu)$ as follows:

$$w_{i,\mu}(k-\mu) = \begin{cases} e, & \text{if operation } i \text{ for vessel } k-\mu \text{ precedes} \\ & \text{operation } i \text{ for vessel } k \\ \varepsilon, & \text{if operation } i \text{ for vessel } k \text{ precedes} \\ & \text{operation } i \text{ for vessel } k-\mu \end{cases} \tag{4-40}$$

Then the adjoint variable $\bar{w}_{i,\mu}(k-\mu)$ is defined as:

$$\bar{w}_{i,\mu}(k-\mu) = \begin{cases} e, & \text{if } w_{i,\mu}(k-\mu) = \varepsilon \\ \varepsilon, & \text{if } w_{i,\mu}(k-\mu) = e \end{cases} \tag{4-41}$$

For every node in the topology graph where the order of the vessels is fixed two ordering constraints must be introduced. The constraints for node i or state x_i are of the form shown in equations Eq. 4-42 and Eq. 4-43. Note the added Max-Plus binary routing control variables $[s_{i,j}(k-\mu) \otimes s_{i,j}(k)]$ which make sure the constraints are only active when both vessel k and $k-\mu$ take the same route.

$$x_i(k) \geq x_i(k-\mu) \otimes \tau_{i,j}(k-\mu) \otimes w_{i,\mu}(k-\mu) \otimes s_{i,j}(k-\mu) \otimes s_{i,j}(k) \tag{4-42}$$

To clarify in words, Eq. (4-42) states the following:

'The time of event x_i for vessel k is greater or equal to the time of event x_i for vessel $k-\mu$ plus some operation time $\tau_{i,j}(k-\mu)$ of the vessel $k-\mu$, only when $s_{i,j}(k-\mu)$ and $s_{i,j}(k)$ are

both e. Thus, when the vessels are sailing the same route event x_i happens first for vessel $k-\mu$ and thus precedes vessel k'

$$x_i(k-\mu) \geq x_i(k) \otimes \tau_{i,j}(k) \otimes \bar{w}_{i,\mu}(k-\mu) \otimes s_{i,j}(k-\mu) \otimes s_{i,j}(k) \tag{4-43}$$

To clarify in words, Eq. (4-43) states the following:

'The time of event x_i for vessel $k-\mu$ is greater or equal to the time of event x_i for vessel k plus some operation time $\tau_{i,j}(k)$ of the vessel k, only when $s_{i,j}(k-\mu)$ and $s_{i,j}(k)$ are both e. Thus, when the vessels are sailing the same route event x_i happens first for vessel k and thus precedes vessel $k-\mu$'

The Max-Plus binary ordering control variables $w_{i,\mu}(k-\mu)$ and $\bar{w}_{i,\mu}(k-\mu)$ as defined in Eq. (4-40) and Eq. (4-41) respectively, will ensure that either equation Eq. (4-42) is active or Eq. (4-43) is active. The Max-Plus binary routing control variables $s_{i,j}(k)$ and $s_{i,j}(k-\mu)$ as defined in Eq.(4-34), will ensure that the ordering constraint is only active when vessel k and $k-\mu$ take the same route.
As introduced in Section 4-3-1 and shown in Figure 4-5, overtaking on the waterways is allowed, only one vessel can enter a lock at a time and we have two locks. Thus, we have to determine the order of the vessels at $x_3(k)$ thus at node 3 and at $x_8(k)$ thus at node 8. Therefore for every vessel pair we will get for Eq. (4-42) and Eq. (4-43) the following four constraints as defined in Eq. (4-44) and Eq. (4-45):

$$\begin{aligned} x_3(k) &\geq x_4(k-\mu) \otimes \tau_{L(3,4)} \otimes w_{3,\mu}(k-\mu) \otimes s_{3,4}(k-\mu) \otimes s_{3,4}(k) \\ x_3(k-\mu) &\geq x_4(k) \otimes \tau_{L(3,4)} \otimes \bar{w}_{3,\mu}(k-\mu) \otimes s_{3,4}(k-\mu) \otimes s_{3,4}(k) \end{aligned} \tag{4-44}$$

and:

$$\begin{aligned} x_8(k) &\geq x_9(k-\mu) \otimes \tau_{L(8,9)} \otimes w_{8,\mu}(k-\mu) \otimes s_{8,9}(k-\mu) \otimes s_{8,9}(k) \\ x_8(k-\mu) &\geq x_9(k) \otimes \tau_{L(8,9)} \otimes \bar{w}_{8,\mu}(k-\mu) \otimes s_{8,9}(k-\mu) \otimes s_{8,9}(k) \end{aligned} \tag{4-45}$$

Further simplification yields:

$$\begin{aligned} x_3(k) &\geq x_4(k-\mu) \otimes \tau_{L(3,4)} \otimes w_{3,\mu}(k-\mu) \otimes s_1(k-\mu) \otimes s_1(k) \\ x_3(k-\mu) &\geq x_4(k) \otimes \tau_{L(3,4)} \otimes \bar{w}_{3,\mu}(k-\mu) \otimes s_1(k-\mu) \otimes s_1(k) \end{aligned} \tag{4-46}$$

and:

$$\begin{aligned} x_8(k) &\geq x_9(k-\mu) \otimes \tau_{L(8,9)} \otimes w_{8,\mu}(k-\mu) \otimes s_2(k-\mu) \otimes s_2(k) \\ x_8(k-\mu) &\geq x_9(k) \otimes \tau_{L(8,9)} \otimes \bar{w}_{8,\mu}(k-\mu) \otimes s_2(k-\mu) \otimes s_2(k) \end{aligned} \tag{4-47}$$

4-3-4 Complete SMPL UDVR system

To complete the model the only required additional equations is a constraint on the arrival time. This will be of the following form:

$$-x_{12}(k-\mu) \geq -a_{12}(k) \tag{4-48}$$

Now that conditions have been derived for $x_i(k)$ that belong to the different routing and ordering control variables a complete set of scheduling decision variables can be made, as follows:

- Routing: $s_l(k)$, $l \in \{1, 2\}$
- Ordering: $w_\mu(k-\mu)$, $\mu \in \{\mu_{\min}, \dots, \mu_{\max}\}$

The decision variables can be stacked into one vector:

$$v(k) = \begin{bmatrix} s_1(k) \\ s_2(k) \\ w_{\mu_{\min}}(k-\mu) \\ \vdots \\ w_{\mu_{\max}}(k-\mu) \end{bmatrix} \in (\mathbb{B}_\varepsilon)^{L_{\text{tot}}} \tag{4-49}$$

with L_{tot} the total number of scheduling variables. Now the scheduling model can be written as:

$$x(k) = \left(\bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu(v(k), k) \otimes x(k-\mu) \right) \oplus B(v(k)) \otimes u(k) \tag{4-50}$$

where:

$$x(k) = A_0^{\text{job}}(\tau_{ij}(k), s_l(k)) \otimes x(k) \oplus \left[\bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu^{\text{ord}}(w_\mu(k-\mu), \tau_{L(i,j)})) \otimes x(k-\mu) \right] \oplus B(v(k)) \otimes u(k) \tag{4-51}$$

with $\tau_{ij} = \{\tau_{w(i,j)}(k), \tau_{L(i,j)}\}$. By choosing a specific control vector $v(k)$ the system switches between different modes of operation, it is thereby an SMPL system.

4-4 Bi-Directional Fixed Routing case

Thirdly, this section will present the SMPL system of the IWT system, which will make it possible for vessels to sail both downstream and upstream. This section will answer the research question:

How to model the Bi-Directional Fixed Routing Inland Waterway Transport system as a Switching Max-Plus Linear system?

Section 4-4 is structured as follows: First, the set up and assumptions for the Bi-Directional Fixed Routing case are given in Section 4-4-1. Next, the Max-Plus routing and ordering equations are given in Section 4-4-2 and Section 4-4-3, respectively. At last, the equations are combined and the Bi-Directional Fixed Routing IWT SMPL model is completed in Section 4-4-4.

4-4-1 BDFR Setup

We define the same variables and assumptions as described in Section 4-1. Next we define the following waterway network consisting of two waterways and a lock in Figure 4-1. All vessels will start either at the left at $x_1(k)$ or on the right at $x_6(k)$ and will travel over the first waterway through the lock over the second waterway to $x_6(k)$ or $x_1(k)$, respectively. The schematic overview of Figure 4-1 with vessels coming from both directions can be transformed in the topology graph of Figure 4-6.

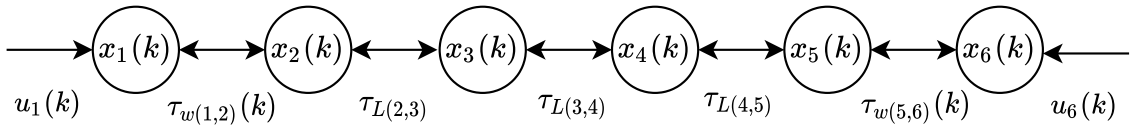


Figure 4-6: Topology graph of the IWT system for the Bi-Directional Variable Routing case

4-4-2 Routing equations

Following the method described in Section 3-5-1, we can derive the Max-Plus routing equations. First, for the *downstream case*, going from state $x_1(k)$ to state $x_6(k)$, we have the Max-Plus routing constraints as defined in Eq. (4-52):

$$\begin{aligned}
x_1(k) &\geq u_1(k) \\
x_2(k) &\geq x_1(k) \otimes \tau_{w(1,2)}(k) \otimes s_{1,2}(k) \\
x_3(k) &\geq x_2(k) \otimes \tau_{L(2,3)} \otimes s_{2,3}(k) \\
x_4(k) &\geq x_3(k) \otimes \tau_{L(3,4)} \otimes s_{3,4}(k) \\
x_5(k) &\geq x_4(k) \otimes \tau_{L(4,5)} \otimes s_{4,5}(k) \\
x_6(k) &\geq x_5(k) \otimes \tau_{w(5,6)}(k) \otimes s_{5,6}(k)
\end{aligned} \tag{4-52}$$

As the routing of an inland vessel going downstream is fixed we have that all routing Max-Plus binary control variables are equal, $s_{1,2}(k) = s_{2,3}(k) = s_{3,4}(k) = s_{4,5}(k) = s_{5,6}(k)$. Thus Eq. (4-52) can be simplified:

$$\begin{aligned}
x_1(k) &\geq u_1(k) \\
x_2(k) &\geq x_1(k) \otimes \tau_{w(1,2)}(k) \otimes s_D(k) \\
x_3(k) &\geq x_2(k) \otimes \tau_{L(2,3)} \otimes s_D(k) \\
x_4(k) &\geq x_3(k) \otimes \tau_{L(3,4)} \otimes s_D(k) \\
x_5(k) &\geq x_4(k) \otimes \tau_{L(4,5)} \otimes s_D(k) \\
x_6(k) &\geq x_5(k) \otimes \tau_{w(5,6)}(k) \otimes s_D(k)
\end{aligned} \tag{4-53}$$

Second, for the *upstream case*, going from state $x_6(k)$ to state $x_1(k)$, we have the routing constraints as defined in Equation (4-54):

$$\begin{aligned}
x_1(k) &\geq x_2(k) \otimes \tau_{w(1,2)}(k) \otimes s_{2,1}(k) \\
x_2(k) &\geq x_3(k) \otimes \tau_{L(2,3)} \otimes s_{3,2}(k) \\
x_3(k) &\geq x_4(k) \otimes \tau_{L(3,4)} \otimes s_{4,3}(k) \\
x_4(k) &\geq x_5(k) \otimes \tau_{L(4,5)} \otimes s_{5,4}(k) \\
x_5(k) &\geq x_6(k) \otimes \tau_{w(5,6)}(k) \otimes s_{6,5}(k) \\
x_6(k) &\geq u_6(k)
\end{aligned} \tag{4-54}$$

As the routing of an inland vessel going upstream is fixed we have that all routing Max-Plus binary control variables are equal, $s_{2,1}(k) = s_{3,2}(k) = s_{4,3}(k) = s_{5,4}(k) = s_{6,5}(k)$. Thus

Equation (4-54) can be simplified:

$$\begin{aligned} x_1(k) &\geq x_2(k) \otimes \tau_{w(1,2)}(k) \otimes s_U(k) \\ x_2(k) &\geq x_3(k) \otimes \tau_{L(2,3)} \otimes s_U(k) \\ x_3(k) &\geq x_4(k) \otimes \tau_{L(3,4)} \otimes s_U(k) \\ x_4(k) &\geq x_5(k) \otimes \tau_{L(4,5)} \otimes s_U(k) \\ x_5(k) &\geq x_6(k) \otimes \tau_{w(5,6)}(k) \otimes s_U(k) \\ x_6(k) &\geq u_6(k) \end{aligned} \tag{4-55}$$

When combining Eq.4-53 and Eq.5-54, we get:

$$\begin{aligned} x_1(k) &\geq u_1(k) \otimes s_D(k) && \oplus x_2(k) \otimes \tau_{w(2,1)}(k) \otimes s_U(k) \\ x_2(k) &\geq x_1(k) \otimes \tau_{w(1,2)}(k) \otimes s_D(k) && \oplus x_3(k) \otimes \tau_{L(3,2)} \otimes s_U(k) \\ x_3(k) &\geq x_2(k) \otimes \tau_{L(2,3)} \otimes s_D(k) && \oplus x_4(k) \otimes \tau_{L(4,3)} \otimes s_U(k) \\ x_4(k) &\geq x_3(k) \otimes \tau_{L(3,4)} \otimes s_D(k) && \oplus x_5(k) \otimes \tau_{L(5,4)} \otimes s_U(k) \\ x_5(k) &\geq x_4(k) \otimes \tau_{L(4,5)} \otimes s_D(k) && \oplus x_6(k) \otimes \tau_{w(6,5)}(k) \otimes s_U(k) \\ x_6(k) &\geq x_5(k) \otimes \tau_{w(5,6)}(k) \otimes s_D(k) && \oplus u_6(k) \otimes s_U(k) \end{aligned} \tag{4-56}$$

Note that the downstream Max-Plus binary control variable $s_{ds}(k)$ and upstream Max-Plus binary control variable $s_{us}(k)$ are each others adjoint, thus the following relationship as defined in Eq. (4-57) applies:

$$s_U(k) = \begin{cases} \varepsilon, & \text{for } s_D(k) = e \\ e, & \text{for } s_D(k) = \varepsilon \end{cases} \tag{4-57}$$

In other words, only one of the two sets of constraints can be active.

4-4-3 Ordering equations

Following the method described in Section 3-5-2, we can derive the Max-Plus ordering equations, which results in 4 pairs of 2 equations. First we will extend the Mintroduce the following additiona

Table 4-4: Definitions of state variables $x_i(k)$ of Figure 4-1

variable	Description of Max-Plus Binary ordering control variables
$w_{DD,\mu}(k-\mu)$	When both vessels are sailing downstream
$w_{UU,\mu}(k-\mu)$	When both vessels are sailing upstream
$w_{DU,\mu}(k-\mu)$	When vessel k is sailing downstream and vessel $k-\mu$ is sailing upstream
$w_{UD,\mu}(k-\mu)$	When vessel $k-\mu$ is sailing downstream and vessel k is sailing upstream

Moreover, we introduce a safety timing τ_{safety}, which is Safety factor time for when a vessel is coming out of the lock and vessel k wants to enter directly. Then for the Max-Plus ordering equations we get:

Vessel k and $k-\mu$ downstream with $k-\mu$ in front:

$$x_3(k) \geq x_4(k-\mu) \otimes w_{DD,\mu}(k-\mu) \otimes \tau_{L(3,4)} \otimes s_{3,4}(k-\mu) \otimes s_{3,4}(k) \quad (4\text{-}58)$$

Vessel k and $k-\mu$ downstream with k in front:

$$x_3(k-\mu) \geq x_4(k) \otimes \bar{w}_{DD,\mu}(k-\mu) \otimes \tau_{L(3,4)} \otimes s_{3,4}(k-\mu) \otimes s_{3,4}(k) \quad (4\text{-}59)$$

Vessel k and $k-\mu$ upstream with $k-\mu$ in front:

$$x_4(k) \geq x_3(k-\mu) \otimes w_{UU,\mu}(k-\mu) \otimes \tau_{L(3,4)} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \quad (4\text{-}60)$$

Vessel k and $k-\mu$ upstream with k in front:

$$x_4(k-\mu) \geq x_3(k) \otimes \bar{w}_{UU,\mu}(k-\mu) \otimes \tau_{L(3,4)} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \quad (4\text{-}61)$$

Vessel k downstream going into the lock and $k-\mu$ upstream coming out of the lock:

$$x_3(k) \geq x_3(k-\mu) \otimes w_{UD,\mu}(k-\mu) \otimes \tau_{safety} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \quad (4\text{-}62)$$

Vessel k downstream coming out of the lock and $k-\mu$ upstream going into the lock:

$$x_4(k-\mu) \geq x_4(k) \otimes \bar{w}_{UD,\mu}(k-\mu) \otimes \tau_{safety} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \quad (4\text{-}63)$$

Vessel k upstream going into the lock and $k-\mu$ downstream coming out of the lock:

$$x_4(k) \geq x_4(k-\mu) \otimes w_{DU,\mu}(k-\mu) \otimes \tau_{safety} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \quad (4\text{-}64)$$

Vessel k upstream coming out of the lock and $k-\mu$ downstream going into the lock:

$$x_3(k-\mu) \geq x_3(k) \otimes \bar{w}_{DU,\mu}(k-\mu) \otimes \tau_{safety} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \quad (4\text{-}65)$$

4-4-4 Complete SMPL BDFR system

To complete the model the only required additional constraints are for the arrival time. This will be of the following form:

$$\begin{aligned} -x_6(k-\mu) &\geq -a_6 \otimes s_D(k) \\ -x_1(k-\mu) &\geq -a_1 \otimes s_U(k) \end{aligned} \quad (4\text{-}66)$$

Now that conditions have been derived for $x_i(k)$ that belong to the different routing and ordering control variables a complete set of scheduling decision variables can be made, as follows:

- Routing: $s_l(k)$, $l \in \{s_D, s_U\}$
- Ordering: $w_\mu(k-\mu)$, $\mu \in \{\mu_{\min}, \dots, \mu_{\max}\}$

The decision variables can be stacked into one vector:

$$v(k) = \begin{bmatrix} s_D(k) \\ s_U(k) \\ \hat{w}_{\mu_{\min}}(k-\mu) \\ \vdots \\ \hat{w}_{\mu_{\max}}(k-\mu) \end{bmatrix} \in (\mathbb{B}_\varepsilon)^{L_{\text{tot}}} \tag{4-67}$$

With

$$\hat{w}_{\mu_{\min}}(k-\mu) = \begin{bmatrix} w_{DD,\mu_{\min}}(k-\mu) & w_{UU,\mu_{\min}}(k-\mu) & w_{DU,\mu_{\min}}(k-\mu) & w_{UD,\mu_{\min}}(k-\mu) \end{bmatrix}^T \tag{4-68}$$

And

$$\hat{w}_{\mu_{\max}}(k-\mu) = \begin{bmatrix} w_{DD,\mu_{\max}}(k-\mu) & w_{UU,\mu_{\max}}(k-\mu) & w_{DU,\mu_{\max}}(k-\mu) & w_{UD,\mu_{\max}}(k-\mu) \end{bmatrix}^T \tag{4-69}$$

And with L_{tot} the total number of scheduling variables. Now the scheduling model can be written as:

$$x(k) = \bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu(v(k), k) \otimes x(k-\mu) \tag{4-70}$$

where:

$$x(k) = A_0^{\text{job}}(\tau_{ij}(k), s_l(k)) \otimes x(k) \oplus \left[\bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu^{\text{ord}}(\hat{w}_\mu(k-\mu), k) \otimes x(k-\mu) \right] \oplus B(v(k)) \otimes u(k) \tag{4-71}$$

By choosing a specific control vector $v(k)$ the system switches between different modes of operation, it is thereby a SMPL system.

4-5 Bi-Directional Variable Routing case

At last, this section will present the SMPL system of the most generic IWT system, which will make it possible for vessels to sail both downstream and upstream and allocate them to a route. This section will answer the research question:

How to model the Bi-Directional Variable Routing Inland Waterway Transport system as a Switching Max-Plus Linear system?

Section 4-5 is structured as follows: First, the set up and assumptions for the Bi-Directional Fixed Routing case are given in Section 4-5-1. Next, the Max-Plus routing and ordering equations are given in Section 4-5-2 and Section 4-5-3, respectively. At last, the equations are combined and the Bi-Directional Fixed Routing IWT SMPL model is completed in Section 4-5-4.

4-5-1 BDVR Setup

We define the same variables and assumptions as described in Section 4-3-1. Next, we define the waterway network shown in Figure 4-7, which consists basically of two parallel systems as the waterway network described in Section 4-4 in Figure 4-6. Thus, the network consists of two corridors each consisting of three waterways and a lock. All vessels will start either at the top at $x_1(k)$ or at the bottom at $x_{12}(k)$ and will travel over the left route or right route to $x_{12}(k)$ or $x_1(k)$, respectively.

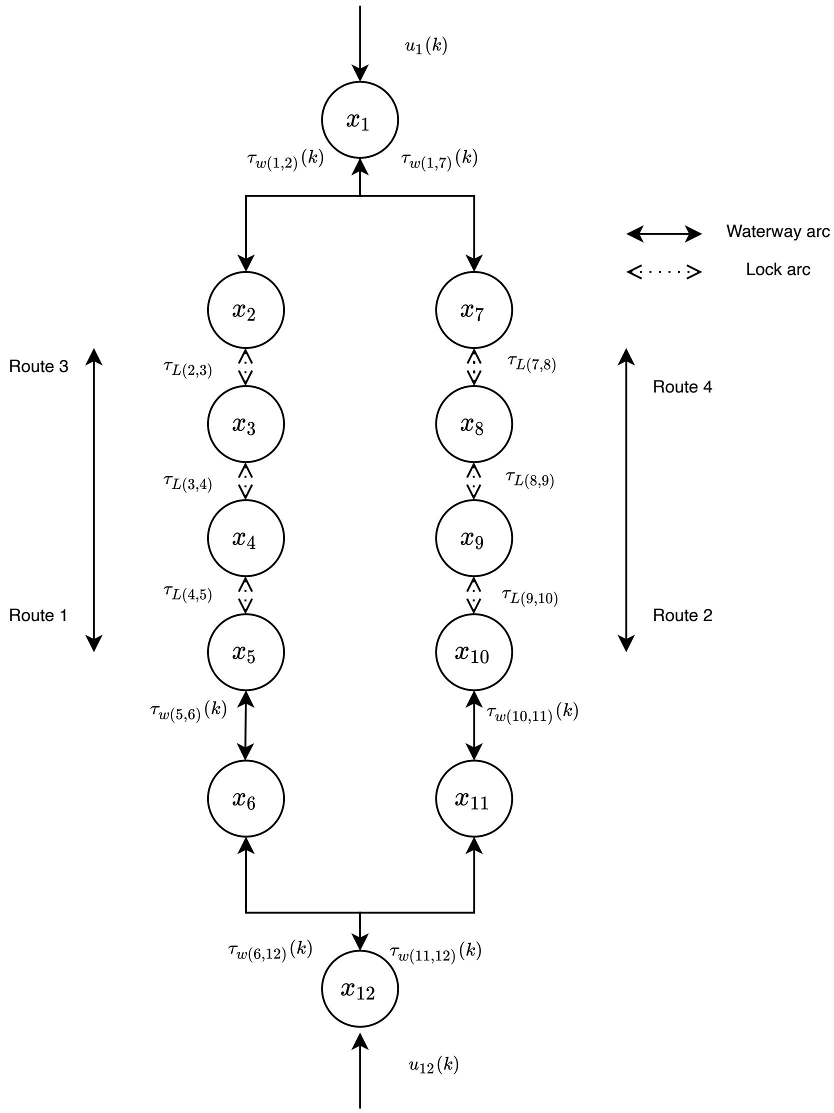


Figure 4-7: Topology graph of the IWT system for the Bi-Directional Variable Routing case

4-5-2 Routing equations

Following the method described in Section 3-5-1, we can derive the Max-Plus routing equations. We immediately apply the simplification as described in Section 4-3-2.
Firstly, for going *downstream* via left *route 1* from state $x_1(k)$ to state $x_{12}(k)$, we have the Max-Plus routing equations as defined in Eq. (4-72):

$$\begin{aligned}
x_1(k) &\geq u(k) \\
x_2(k) &\geq x_1(k) \otimes \tau_{w(1,2)}(k) \otimes s_1(k) \\
x_3(k) &\geq x_2(k) \otimes \tau_{L(2,3)} \otimes s_1(k) \\
x_4(k) &\geq x_3(k) \otimes \tau_{L(3,4)} \otimes s_1(k) \\
x_5(k) &\geq x_4(k) \otimes \tau_{L(4,5)} \otimes s_1(k) \\
x_6(k) &\geq x_5(k) \otimes \tau_{w(5,6)}(k) \otimes s_1(k) \\
x_{12}(k) &\geq x_6(k) \otimes \tau_{w(6,12)}(k) \otimes s_1(k)
\end{aligned} \tag{4-72}$$

Secondly, for going *downstream* via right *route 2* from state $x_1(k)$ to state $x_{12}(k)$, we have the Max-Plus routing equations as defined in Eq. (4-73):

$$\begin{aligned}
x_1(k) &\geq u(k) \\
x_7(k) &\geq x_1(k) \otimes \tau_{w(1,7)}(k) \otimes s_2(k) \\
x_8(k) &\geq x_7(k) \otimes \tau_{L(7,8)} \otimes s_2(k) \\
x_9(k) &\geq x_8(k) \otimes \tau_{L(8,9)} \otimes s_2(k) \\
x_{10}(k) &\geq x_9(k) \otimes \tau_{L(9,10)} \otimes s_2(k) \\
x_{11}(k) &\geq x_{10}(k) \otimes \tau_{w(10,11)}(k) \otimes s_2(k) \\
x_{12}(k) &\geq x_{11}(k) \otimes \tau_{w(11,12)}(k) \otimes s_2(k)
\end{aligned} \tag{4-73}$$

Thirdly, for going *upstream* via left *route 3* from state $x_{12}(k)$ to state $x_1(k)$, we have the Max-Plus routing equations as defined in Eq. (4-74):

$$\begin{aligned}
x_1(k) &\geq x_2(k) \otimes \tau_{w(1,2)}(k) \otimes s_3(k) \\
x_2(k) &\geq x_3(k) \otimes \tau_{L(2,3)} \otimes s_3(k) \\
x_3(k) &\geq x_4(k) \otimes \tau_{L(3,4)} \otimes s_3(k) \\
x_4(k) &\geq x_5(k) \otimes \tau_{L(4,5)} \otimes s_3(k) \\
x_5(k) &\geq x_6(k) \otimes \tau_{w(5,6)}(k) \otimes s_3(k) \\
x_6(k) &\geq x_{12}(k) \otimes \tau_{w(6,12)}(k) \otimes s_3(k) \\
x_{12}(k) &\geq u_{12}(k)
\end{aligned} \tag{4-74}$$

Fourthly, for going *upstream* via right *route 4* from state $x_{12}(k)$ to state $x_1(k)$, we have the

Max-Plus routing equations as defined in Eq. (4-75):

$$\begin{aligned} x_1(k) &\geq x_7(k) \otimes \tau_{w(1,7)}(k) \otimes s_4(k) \\ x_7(k) &\geq x_8(k) \otimes \tau_{L(7,8)} \otimes s_4(k) \\ x_8(k) &\geq x_9(k) \otimes \tau_{L(8,9)} \otimes s_4(k) \\ x_9(k) &\geq x_{10}(k) \otimes \tau_{L(9,10)} \otimes s_4(k) \\ x_{10}(k) &\geq x_{11}(k) \otimes \tau_{w(10,11)}(k) \otimes s_4(k) \\ x_{11}(k) &\geq x_{12}(k) \otimes \tau_{w(11,12)}(k) \otimes s_4(k) \\ x_{12}(k) &\geq u_{12}(k) \end{aligned} \tag{4-75}$$

At last, for the routing we have to make sure that only one route is selected. We can do this as follows:

$$s_1(k) \otimes s_2(k) \otimes s_3(k) \otimes s_4(k) \otimes \leq 1 \tag{4-76}$$

$$-s_1(k) \otimes -s_2(k) \otimes -s_3(k) \otimes -s_4(k) \otimes \leq -1 \tag{4-77}$$

4-5-3 Ordering equations

Following the method described in Section 3-5-2, we can derive the Max-Plus ordering equations, which results in 4 pairs of 2 equations. This is worked out below for lock 1 (between node 3 and 4), but should also be done for lock 2 (between node 8 and 9)

Vessel k and $k-\mu$ downstream with $k-\mu$ in front:

$$x_3(k) \geq x_4(k-\mu) \otimes w_{DD,\mu}(k-\mu) \otimes \tau_{L(3,4)} \otimes s_{3,4}(k-\mu) \otimes s_{3,4}(k) \tag{4-78}$$

Vessel k and $k-\mu$ downstream with k in front:

$$x_3(k-\mu) \geq x_4(k) \otimes \bar{w}_{DD,\mu}(k-\mu) \otimes \tau_{L(3,4)} \otimes s_{3,4}(k-\mu) \otimes s_{3,4}(k) \tag{4-79}$$

Vessel k and $k-\mu$ upstream with $k-\mu$ in front:

$$x_4(k) \geq x_3(k-\mu) \otimes w_{UU,\mu}(k-\mu) \otimes \tau_{L(3,4)} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \tag{4-80}$$

Vessel k and $k-\mu$ upstream with k in front:

$$x_4(k-\mu) \geq x_3(k) \otimes \bar{w}_{UU,\mu}(k-\mu) \otimes \tau_{L(3,4)} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \tag{4-81}$$

Vessel k downstream going into the lock and $k-\mu$ upstream coming out of the lock:

$$x_3(k) \geq x_3(k-\mu) \otimes w_{UD,\mu}(k-\mu) \otimes \tau_{safety} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \tag{4-82}$$

Vessel k downstream coming out of the lock and $k-\mu$ upstream going into the lock:

$$x_4(k-\mu) \geq x_4(k) \otimes \bar{w}_{UD,\mu}(k-\mu) \otimes \tau_{safety} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \tag{4-83}$$

Vessel k upstream going into the lock and $k-\mu$ downstream coming out of the lock:

$$x_4(k) \geq x_4(k-\mu) \otimes w_{DU,\mu}(k-\mu) \otimes \tau_{safety} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \tag{4-84}$$

Vessel k upstream coming out of the lock and $k-\mu$ downstream going into the lock:

$$x_3(k-\mu) \geq x_3(k) \otimes \bar{w}_{DU,\mu}(k-\mu) \otimes \tau_{safety} \otimes s_{4,3}(k-\mu) \otimes s_{4,3}(k) \tag{4-85}$$

4-5-4 Complete SMPL BDVR model

To complete the model the only additional constraint required is for the arrival time. This will be of the following form:

$$\begin{aligned} -x_{12}(k-\mu) &\geq -a_{12} \otimes s_1(k) \\ -x_{12}(k-\mu) &\geq -a_{12} \otimes s_3(k) \\ -x_1(k-\mu) &\geq -a_1 \otimes s_2(k) \\ -x_1(k-\mu) &\geq -a_1 \otimes s_4(k) \end{aligned} \tag{4-86}$$

Now that conditions have been derived for $x_i(k)$ that belong to the different routing and ordering control variables a complete set of Max-Plus binary control variables can be made, as follows:

- Routing: $s_l(k)$, $l \in \{1,2,3,4\}$
- Ordering: $w_\mu(k-\mu)$, $\mu \in \{\mu_{\min}, \dots, \mu_{\max}\}$

The Max-Plus binary control can be stacked into one vector:

$$v(k) = \begin{bmatrix} s_1(k) \\ s_2(k) \\ s_3(k) \\ s_4(k) \\ w_{\mu_{\min}}(k-\mu) \\ \vdots \\ w_{\mu_{\max}}(k-\mu) \end{bmatrix} \in (\mathbb{B}_\varepsilon)^{L_{\text{tot}}} \tag{4-87}$$

with L_{tot} the total number of scheduling variables. Now the scheduling model can be written as:

$$x(k) = \bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu(v(k),k) \otimes x(k-\mu) \oplus B(k) \otimes u(k) \tag{4-88}$$

where:

$$x(k) = A_0^{\text{job}}(\tau_{ij}(k), s_l(k), k) \otimes x(k) \oplus \left[\bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu^{\text{ord}}(w_\mu(k),k) \otimes x(k-\mu) \right] \oplus B(v(k)) \otimes u(k) \tag{4-89}$$

By choosing a specific control vector $v(k)$ the system switches between different modes of operation, it is thereby a SMPL system.
Note, that since we now have > 2 routes it makes sense to parameterise the Max-Plus routing binary control variables as described in Section 3-6-1, and can be seen in Table 3-4. We can reduce the 4 nominal Max-Plus routing binary control variables to 2 parameterised Max-Plus routing binary control variables, as follows:

$$\begin{aligned} s_1(k) &= \bar{\eta}_1(k) \otimes \bar{\eta}_2(k) \\ s_2(k) &= \bar{\eta}_1(k) \otimes \eta_2(k) \\ s_3(k) &= \eta_1(k) \otimes \bar{\eta}_2(k) \\ s_4(k) &= \eta_1(k) \otimes \eta_2(k) \end{aligned} \tag{4-90}$$

Where the values of $[\eta_1(k), \eta_2(k)]$ for each alternative route $l(k)$ can be seen in Table 4-5:

Table 4-5: Parameterisation for $L = 4$

$\ell(k)$	$\eta_1(k)$	$\eta_2(k)$
1	ε	ε
2	ε	e
3	e	ε
4	e	e

Now the complete set of scheduling decision variables can be made, as follows:

- Routing: $\eta_l(k)$, $l \in \{1, 2\}$
- Ordering: $w_\mu(k - \mu)$, $\mu \in \{\mu_{\min}, \dots, \mu_{\max}\}$

The decision variables can be stacked into one vector:

$$v(k) = \begin{bmatrix} \eta_1 \\ \eta_2 \\ w_{\mu_{\min}}(k) \\ \vdots \\ w_{\mu_{\max}}(k) \end{bmatrix} \in (\mathbb{B}_\varepsilon)^{L_{\text{tot}}} \tag{4-91}$$

with L_{tot} the total number of scheduling variables. Now the scheduling model can be written as:

$$x(k) = \bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu(v(k), k) \otimes x(k - \mu) \oplus B(v(k)) \otimes u(k) \tag{4-92}$$

When expanded:

$$x(k) = A_0^{\text{job}}(\tau_{ij}(k), \eta_l(k), k) \otimes x(k) \oplus \left[\bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu^{\text{ord}} \mu(w_\mu(k), k) \otimes x(k - \mu) \right] \oplus B(v(k)) \otimes u(k) \tag{4-93}$$

Due to the reduced number of Max-Plus binary routing control variables Eq (4-93) should be computationally less demanding than Eq (5-3).

4-6 Conclusion

The goal of this Chapter 4 was to model the four IWT cases as SMPL systems such that these could, later on, be used for scheduling. As stated in Chapter 1, this chapter addressed the research question:

How to model IWT systems as SMPL systems?

First of all, it has been shown that a simple IWT system (i.e. the UDFR case), consisting of two waterways with a lock in the middle, can be modelled as a DES consisting of six states. Moreover, it was shown that when describing two vessels sailing this simple IWT DES the in conventional algebra nonlinear maximum operator is required for defining the ordering relationship of these two vessels in the IWT system. As mentioned earlier, DESs with this maximum operator can be written in a 'linear' fashion, in other words, Max-Plus-Linear. This way, the benefit and advantage of modelling IWT systems as SMPL systems were shown.
Next, it was thoroughly described for four IWT cases that we can model them as SMPL systems by describing their routing and ordering dynamics. This is done by defining the Max-Plus routing binary control variables and the Max-Plus ordering binary control variables used within the routing and order equations. Moreover, for the most generic IWT case, the BDVR case, the Max-Plus routing binary control variables were parameterised as described in Section 3-6, to reduce the complexity of the model.
As shown in Section 3-7, the SMPL systems have to be transformed Mixed Integer Linear Programming (MILP) models to use them for scheduling. This will be shown in the next Chapter 5, including the complete MILP IWT model formulation

Chapter 5

MILP models of Inland Waterway Transport systems

After having described how a Switching Max-Plus Linear (SMPL) system can be transformed to a Mixed Integer Linear Programming (MILP) problem, we will do this for the derived SMPL systems in Chapter 4. Therefore, as introduced in Chapter 1, this chapter will answer the research question:

How to transform SMPL IWT systems to MILP models?

Chapter 5 is structured as follows: Section 5-1 will give an introduction to the MILP Inland Waterway Transport (IWT) model formulation. In addition, the objective function which defines the optimisation goals is stated Then we continue with the MILP models of the four IWT cases. Firstly, Section 5-2 will describe the Uni-Directional Fixed Routing case. Secondly, Section 5-3 will state the Uni-Directional Variable Routing case. Thirdly, Section 5-4 will present the Bi-Directional Fixed Routing case. At last, Section 5-5 will describe the Bi-Directional Variable Routing case. At last, Section 5-6 will summarise the findings and answer the corresponding research question.

5-1 Introduction to the MILP IWT model formulation

In the previous chapters, we often discussed 'optimal' scheduling. This section will introduce what this optimal scheduling entails. Moreover, we will show how we will achieve this through MILP for the four SMPL IWT systems, as presented in Chapter 4.
In the context of current challenges in the IWT system as described in Chapter 2, such as long waiting times and suboptimal routing, we want to create a schedule that ensures vessels reach their destinations quickly without delays. This will be done by scheduling them on the most optimal route and scheduling them in the most optimal order through locks. Therefore, the purpose of the scheduler is to generate a schedule that will make sure a set of vessels can go as

fast as possible from their departure locations to their arrival locations while all are meeting their arrival deadlines. In other words, the scheduler will answer the questions; what is the fastest way for a set of vessels to sail from their departure locations to their arrival locations? Note, we allow vessels to arrive before their arrival deadlines and, as will be shown later, this will not be punished either. Of course, it is not always convenient for vessels to arrive earlier than their scheduled arrival deadlines (e.g. availability of berth locations). However, to keep the research manageable, and this being the first time IWT systems are described using SMPL systems, this objective will be the only scope. Further recommendations for objective functions will be given in the final chapter.
The designed scheduler in this research will have an objective function that resembles the mentioned Make-span $C_{\max}$, as discussed in Section 3-7 the most, however, with a small extension. In order to be entirely similar to the Make-span would mean, we want to have all vessels arrive at their deadline locations as soon as possible by minimising the arrival time of the vessel that takes the longest to arrive. In our case, we will extend this to the minimisation of the sum of the arrival times of all vessels. This will guarantee that the scheduler generates a solution that ensures that all vessels in the network reach their arrival destination as soon as possible while meeting the constraints of the IWT system.
We assume that all required information by the scheduler is known on beforehand and does not change. That is why we state that the scheduler operates in an offline open-loop way. We define the optimal scheduling problem as follows:

$$\min_{x\in\mathbb{R}^n, v\in\mathbb{Z}^k} \quad J(k) \tag{5-1}$$

$$\text{subject to} \quad x(k) \geq \bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu(v(k),k) \otimes x(k-\mu) \oplus B(k) \otimes u(k) \tag{5-2}$$

Where Eq (5-1) is the objective function and Eq (5-2) are the SMPL system constraints (i.e. system dynamics), which follow from Eq (4-92) in Section 4-5-4. To clarify, Eq (5-2) is a shorter notation with all Max-Plus binary control variables grouped together in $v(k)$ for:

$$x(k) \geq A_0(\tau_{ij}(k), s_l(k), k) \otimes x(k) \oplus \left[\bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A_\mu(w_\mu(k),k) \otimes x(k-\mu)\right] \oplus B(k) \otimes u(k) \tag{5-3}$$

To recap, thus for Eq (5-2) we have:

$$v(k) = \begin{bmatrix} s_1(k) \\ \vdots \\ s_l(k) \\ w_{\mu_{\min}}(k-\mu) \\ \vdots \\ w_{\mu_{\max}}(k-\mu) \end{bmatrix} \in \mathbb{B}_\varepsilon^{(V_{\text{tot}})\times 1} \tag{5-4}$$

Next we will define these objective function in Eq (5-1) and SMPL system constraints in Eq (5-2). First, we have the objective function $J(k)$, which consists of an objective function for the output $J_{\text{out}}(k)$ and an objective function for the input $J_{\text{in}}(k)$, as follows:

$$J(k) = J_{\text{out}}(k) + J_{\text{in}}(k) \tag{5-5}$$

The output objective function $J_{\text{out}}(k)$ weighs the optimisation of the states $x(k)$, as follows:

$$J_{out}(k) = \sum_{i=1}^{n} \sigma_i(k) x_i(k) \tag{5-6}$$

For a system with n states such that $i \in [1, \ldots, n]$. For the output weight function σ we define the following variables:

- $\sigma_a(k)$: Output weight for arrival location for vessel k
- $\sigma_d(k)$: Output weight for departure and intermediate locations for vessel k

For the Uni-Directional cases we have:

$$\sigma_{\text{UD}}(k) = \begin{bmatrix} \sigma_d(k) & \ldots & \sigma_d(k) & \ldots & \sigma_a(k) \end{bmatrix}, \sigma_{\text{UD}}(k) \in \mathbb{R}^{1\times n} \tag{5-7}$$

For the Bi-Directional cases we have:

$$\sigma_{\text{BD}}(k) = \begin{bmatrix} \sigma_a(k) & \ldots & \sigma_d(k) & \ldots & \sigma_a(k) \end{bmatrix}, \sigma_{\text{BD}}(k) \in \mathbb{R}^{1\times n} \tag{5-8}$$

Then we get the following output weight functions σ for the four different cases with $\sigma_a = 1$ and $\sigma_d = 10^{-4}$:

$$\sigma_{\text{UDFR}}(k) = \begin{bmatrix} 10^{-4} & 10^{-4} & \ldots & 10^{-4} & 1 \end{bmatrix}, \sigma_{\text{UDFR}}(k) \in \mathbb{R}^{1\times 6} \tag{5-9}$$

$$\sigma_{\text{UDVR}}(k) = \begin{bmatrix} 10^{-4} & 10^{-4} & \ldots & 10^{-4} & 1 \end{bmatrix}, \sigma_{\text{UDVR}}(k) \in \mathbb{R}^{1\times 12} \tag{5-10}$$

$$\sigma_{\text{BDFR}}(k) = \begin{bmatrix} 1 & 10^{-4} & \ldots & 10^{-4} & 1 \end{bmatrix}, \sigma_{\text{BDFR}}(k) \in \mathbb{R}^{1\times 6} \tag{5-11}$$

$$\sigma_{\text{BDVR}}(k) = \begin{bmatrix} 1 & 10^{-4} & \ldots & 10^{-4} & 1 \end{bmatrix}, \sigma_{\text{BDVR}}(k) \in \mathbb{R}^{1\times 12} \tag{5-12}$$

In order to be completely accurate, we would actually have to define the output weight functions for Bi-Directional cases as follows:

$$\sigma_{\text{BDFR}}(k) = \begin{bmatrix} \sigma_{a_1}(k) & 10^{-4} & \ldots & 10^{-4} & \sigma_{a_6}(k) \end{bmatrix}, \sigma_{\text{BDFR}}(k) \in \mathbb{R}^{1\times 6} \tag{5-13}$$

$$\sigma_{\text{BDVR}}(k) = \begin{bmatrix} \sigma_{a_1}(k) & 10^{-4} & \ldots & 10^{-4} & \sigma_{a_{12}}(k) \end{bmatrix}, \sigma_{\text{BDVR}}(k) \in \mathbb{R}^{1\times 12} \tag{5-14}$$

With:

$$\sigma_{a_1}(k) = \begin{cases} 10^{-4}, & \text{for } s_D(k) = e \\ 1, & \text{for } s_U(k) = \varepsilon \end{cases} \tag{5-15}$$

and

$$\sigma_{a_6}(k) = \begin{cases} 1, & \text{for } s_D(k) = e \\ 10^{-4}, & \text{for } s_U(k) = \varepsilon \end{cases}, \quad \sigma_{a_{12}}(k) = \begin{cases} 1, & \text{for } s_D(k) = e \\ 10^{-4}, & \text{for } s_U(k) = \varepsilon \end{cases} \tag{5-16}$$

This ensures that only the arrival location is heavily punished and not the input location. However, as the first states are defined as $x_1(k) \geq u_1(k)$ or $x_{12}(k) \geq u_{12}(k)$, and not influenced by anything else, which leads to $x_1(k) = u_1(k)$ and $x_{12}(k) = u_{12}(k)$ the first state will always be equal to the input and the scheduler therefore cannot optimise the values. Thus, in practice

Eq 5-11 and Eq 5-12 will work fine as output weight functions for the objective function. Moreover, in Eq 5-9 to Eq 5-12 we can see that we have a larger output weight associated with the states which correspond to an arrival destination location. For the Uni-Directional cases this is always only the last event state $x_6(k)$ for Fixed Routing and $x_{12}(k)$ for Variable Routing. For the Bi-Directional cases this is both for the first event state $x_1(k)$ or the last event states, again, $x_6(k)$ for Fixed Routing and $x_{12}(k)$ for Variable Routing. Moreover, although the objective is to just minimise the arrival time of the vessels, the other event states in between must also be multiplied by a small constant $\sigma_d > 0$. This is to ensure the vessels achieve the proposed sailing timings on the respective waterways. The input objective function $J_{\text{in}}(k)$ weighs the optimisation of the Max-Plus binary control vector $v(k)$, as follows:

$$J_{in}(k) = \sum_{i=1}^{V_{\text{tot}}} \lambda_i(k) v_i(k) \tag{5-17}$$

Where the input weight function λ is defined for all cases as:

$$\lambda(k) = \begin{bmatrix} 0 & \dots & 0 \end{bmatrix}, \lambda(k) \in \mathbb{R}^{1\times V_{\text{tot}}} \tag{5-18}$$

The weights in the input weight function λ can all be chosen equal to zero, because in the IWT systems there is no punishment on applying a certain control action (i.e. taking a particular route or swapping to a specific order). However, these could be used later if for example certain routes are undesired for environmental reasons. Then completing the objective function we get:

$$J(k) = J_{\text{out}}(k) + J_{\text{in}}(k) \tag{5-19}$$

$$= \sum_{i=1}^{n} \sigma_i(k) x_i(k) + \sum_{i=1}^{V_{\text{tot}}} \lambda_i v_i(k) \tag{5-20}$$

If we rewrite this to MILP formulation as presented in Section 3-7, we get:

$$J(k) = c_x^T x(k) + c_v^T v(k) \tag{5-21}$$

With $c_x^T = \sigma(k)$ and $c_v^T = \lambda(k)$, such that again the following MILP problem is obtained:

$$\begin{aligned} \min_{x\in\mathbb{R}^n, v\in\mathbb{Z}^k} \quad & \begin{bmatrix} c_x^T & c_v^T \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} \\ \text{subject to} \quad & \begin{bmatrix} E_x & E_v \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} \leq \begin{bmatrix} b \end{bmatrix} \end{aligned} \tag{5-22}$$

With $c_x \in \mathbb{R}^{n\times 1}$, $c_v \in \mathbb{B}^{V_{\text{tot}}\times 1}$, $b \in \mathbb{R}^m$, $E_x \in \mathbb{R}^{m\times n}$, and $E_v \in \mathbb{R}^{m\times V_{\text{tot}}}$. The remaining part of this chapter will show how we convert the SMPL systems, used for the inequality constraints in the MILP model in Eq (5-2), to suit the MILP formulation of Eq (5-22) such that they can used for optimisation. Therefore, the next sections will present the transformation of the SMPL systems to MILP models; first for the Uni-Directional Fixed Routing (UDFR) case in Section 5-2, next for the Uni-Directional Variable Routing (UDVR) case in Section 5-3, after which Section 5-4 presents the Bi-Directional Fixed Routing (BDFR) case, and at last Section 5-5 shows the transformation for the Bi-Directional Variable Routing (BDVR) case.

5-2 Uni-Directional Fixed Routing case

This section will define the MILP constraints for the uni-directional fixed routing case. This is first done in Section 5-2-1 for the routing constraints, after which the ordering constraints are derived in Section 5-2-2.

5-2-1 Routing constraints

The Max-Plus routing equations defined in Section 4-2-2 will be transformed to the MILP constraints required for the optimisation, using the method presented in Section 3-8. As we have no Max-Plus binary control variables this is rather straightforward and goes as follows for the second constraint for $x_2(k)$:

$$\begin{aligned} x_2(k) &\geq x_1(k) \otimes \tau_{w(1,2)}(k) \\ x_2(k) &\geq x_1(k) + \tau_{w(1,2)}(k) \\ -x_2(k) + x_1(k) &\leq -\tau_{w(1,2)}(k) \end{aligned} \tag{5-23}$$

The same method is used to transform the other five constraints. This leads to the following set of constraints for vessel k:

$$\begin{aligned} -x_1(k) &\leq -u(k) \\ -x_2(k) + x_1(k) &\leq -\tau_{w(1,2)}(k) \\ -x_3(k) + x_2(k) &\leq -\tau_{L(2,3)} \\ -x_4(k) + x_3(k) &\leq -\tau_{L(3,4)} \\ -x_5(k) + x_4(k) &\leq -\tau_{L(4,5)} \\ -x_6(k) + x_5(k) &\leq -\tau_{w(5,6)}(k) \end{aligned} \tag{5-24}$$

5-2-2 Ordering constraints

The Max-Plus ordering equations defined in Section 4-2-3 will be transformed to the MILP constraints required for the optimisation, using the method presented in Section 3-8.
The Max-Plus ordering equations consist of the scheduling parameters (i.e. Max-Plus binary control variables) that are either zero e (i.e. Max-Plus one) or infinity ε (i.e. Max-Plus zero). For the actual numerical implementation and simulation the infinity value ε cannot be used. Therefore, the Max-Plus binary control variables have to be replaced by conventional binary control variables. To do so, the following approximation is used, where

$$w_{i,\mu} \approx \beta w^c_{i,\mu} \tag{5-25}$$

where the index $(k - \mu)$ is left out of the control variables for convenience. Moreover, $\beta \ll 0$ is a large negative number and $w^c_{i,\mu}$ is the conventional ordering control variable at event i between vessel k and vessel $k - \mu$, which is defined as:

$$w^c_{i,\mu} = \begin{cases} 0, & \text{for } w_{i,\mu} = e \\ 1, & \text{for } w_{i,\mu} = \varepsilon \end{cases} \tag{5-26}$$

Thus $w^c_{i,\mu)} = 0$ will activate (i.e. select) an ordering constraint and $w^c_{i,\mu)} = 1$ will deactivate (i.e. not select) an ordering constraint. Furthermore, the adjoint of $w_{i,\mu}$ which is $\bar{w}_{i,\mu}$ can be approximated by:

$$\bar{w}_{i,\mu} \approx \beta \left(1 - w^c_{i,\mu}\right) \tag{5-27}$$

With the definition of the control variable conversion, we can now transform the Max-Plus ordering equations defined in Section 4-2-3, using Eq. (5-74) and Eq. (5-76). We will completely workout a Max-Plus to MILP transformation for the ordering between vessel k and $k-1$ in Eq. (5-28) and Eq. (5-29), and then list all conventional orderings constraints for 4 vessels in Eq. (5-30) to Eq. (5-34).First for Eq. (4-26a) with $\mu = 1$, we get:

$$\begin{aligned} x_3(k) &\geq x_4(k-\mu) \otimes \tau_{L(3,4)} \otimes w_{3,\mu} \\ x_3(k) &\geq x_4(k-1) \otimes \tau_{L(3,4)} \otimes w_{3,1} \\ x_3(k) &\geq x_4(k-1) + \tau_{L(3,4)} + \beta w^c_{3,1} \\ -x_3(k) + x_4(k-1) + \beta w^c_{3,1} &\leq -\tau_{L(3,4)} \end{aligned} \tag{5-28}$$

Secondly for Eq. (4-26b) with $\mu = 1$, we get:

$$\begin{aligned} x_3(k-\mu) &\geq x_4(k-\mu) \otimes \tau_{L(3,4)} \otimes \bar{w}_{3,\mu} \\ x_3(k-1) &\geq x_4(k) \otimes \tau_{L(3,4)} \otimes \bar{w}_{3,1} \\ x_3(k-1) &\geq x_4(k) + \tau_{L(3,4)} + \beta(1 - w^c_{3,1}) \\ x_3(k-1) &\geq x_4(k) + \tau_{L(3,4)} + \beta - \beta w^c_{3,1} \\ x_3(k-1) - x_4(k) + \beta w^c_{3,1} &\geq \beta + \tau_{L(3,4)} \\ -x_3(k-1) + x_4(k) - \beta w^c_{3,1} &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-29}$$

Next, using the conventional binary control variables defined in Table 5-1 we can define the MILP ordering constraints for all vessel ordering combinations for 4 vessels. Which are 6 pairs of the 2 inequality constraints, as follows:

The MILP constraint for order between vessel 1 (k) and vessel 2 ($k-1$), using conventional control variable $w^c_{3,1}$:

$$\begin{aligned} -x_3(k) + x_4(k-1) + \beta w^c_{3,1} &\leq -\tau_{L(3,4)} \\ -x_3(k-1) + x_4(k) - \beta w^c_{3,1} &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-30}$$

The MILP constraint for ordering between vessel 1 (k) and vessel 3 ($k-2$), using conventional control variable $w^c_{3,2}$:

$$\begin{aligned} -x_3(k) + x_4(k-2) + \beta w^c_{3,2} &\leq -\tau_{L(3,4)} \\ -x_3(k-2) + x_4(k) - \beta w^c_{3,2} &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-31}$$

The MILP constraint for ordering between vessel 1 (k) and vessel 4 ($k-3$), using conventional control variable $w^c_{3,3}$:

$$\begin{aligned} -x_3(k) + x_4(k-3) + \beta w^c_{3,3} &\leq -\tau_{L(3,4)} \\ -x_3(k-3) + x_4(k) - \beta w^c_{3,3} &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-32}$$

The MILP constraint for ordering between vessel 2 $(k-1)$ and vessel 3 $(k-2)$, using conventional control variable $w^c_{3,2}$:

$$\begin{aligned} -x_3(k-1) + x_4(k-2) + \beta w^c_{3,2} &\leq -\tau_{L(3,4)} \\ -x_3(k-2) + x_4(k-1) - \beta w^c_{3,2} &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-33}$$

The MILP constraint for ordering between vessel 2 $(k-1)$ and vessel 4 $(k-3)$, using conventional control variable $w^c_{3,3}$:

$$\begin{aligned} -x_3(k-1) + x_4(k-3) + \beta w^c_{3,3} &\leq -\tau_{L(3,4)} \\ -x_3(k-3) + x_4(k-1) - \beta w^c_{3,3} &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-34}$$

The MILP constraint for ordering between vessel 3 $(k-2)$ and vessel 4 $(k-3)$, using conventional control variable $w^c_{3,3}$:

$$\begin{aligned} -x_3(k-2) + x_4(k-3) + \beta w^c_{3,3} &\leq -\tau_{L(3,4)} \\ -x_3(k-3) + x_4(k-2) - \beta w^c_{3,3} &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-35}$$

Table 5-1: Definitions of the 6 conventional control variables

Ordering between vessels	Event index	Conventional control variable
1 and 2	(k) and $(k-1)$	$w^c_{3,1}$
1 and 3	(k) and $(k-2)$	$w^c_{3,2}$
1 and 4	(k) and $(k-3)$	$w^c_{3,3}$
2 and 3	$(k-1)$ and $(k-2)$	$w^c_{3,2}$
2 and 4	$(k-1)$ and $(k-3)$	$w^c_{3,3}$
3 and 4	$(k-2)$ and $(k-3)$	$w^c_{3,3}$

As can be seen, from Table 5-1, to model 4 vessels we require 6 control variables. As explained in Section 3-6-2, the number of control variables required only depends on the number of vessels and is given by the following equation:

$$\text{number of required control variables} \;=\; \frac{n(n-1)}{2} \tag{5-36}$$

With n number of vessels.

5-3 Uni-Directional Variable Routing case

This section will define the MILP constraints for the Uni-Directional Fixed Routing case. This is first done in Section 5-3-1 for the routing constraints, after which the ordering constraints are derived in Section 5-3-2.

5-3-1 Routing constraints

The Max-Plus routing equations defined in Section 4-3-2 will be transformed to the MILP constraints required for the optimisation, using the method presented in Section 3-8. This will be done in the same way as is done in Section 5-2-1, however, this time we do have the Max-Plus binary control variable $s_{i,j}(k)$ that has to be transformed, as follows:

$$s_{i,j}(k) \approx \beta s^c_{i,j}(k) \tag{5-37}$$

Where $\beta \ll 0$ is a very large negative number and $s^c_{i,j}(k)$ is the conventional routing control variable on the arc from node i to node j, which is defined as:

$$s^c_{i,j}(k) = \begin{cases} 0, & \text{for } s_{i,j}(k) = e \\ 1, & \text{for } s_{i,j}(k) = \varepsilon \end{cases} \tag{5-38}$$

Thus $s^c_{i,j}(k)$ will activate (i.e. select) a routing constraint and $s^c_{i,j}(k) = 1$ will deactivate (i.e. not select) an a routing constraint. As explained in Section 3-5-1, the Max-Plus binary control variable $s_{ij}(k)$ does not have an adjoint. Therefore the conventional binary control variable $s^c_{i,j}(k)$ does not have one either.
Using the relationships defined in Eq. 5-37 and Eq. 5-38, we can transform the Max-Plus routing equations defined in Section 4-3-2 to MILP constraints. We then get the following set:

$$\begin{aligned}
-x_1(k) &\leq -u(k) \\
\\
-x_2(k) + x_1(k) + \beta s^c_1(k) &\leq -\tau_{w(1,2)}(k) \\
-x_3(k) + x_2(k) + \beta s^c_1(k) &\leq -\tau_{L(2,3)} \\
-x_4(k) + x_3(k) + \beta s^c_1(k) &\leq -\tau_{L(3,4)} \\
-x_5(k) + x_4(k) + \beta s^c_1(k) &\leq -\tau_{L(4,5)} \\
-x_6(k) + x_5(k) + \beta s^c_1(k) &\leq -\tau_{w(5,6)}(k) \\
\\
-x_7(k) + x_1(k) + \beta s^c_2(k) &\leq -\tau_{w(1,7)}(k) \\
-x_8(k) + x_7(k) + \beta s^c_2(k) &\leq -\tau_{L(7,8)} \\
-x_9(k) + x_8(k) + \beta s^c_2(k) &\leq -\tau_{L(8,9)} \\
-x_{10}(k) + x_9(k) + \beta s^c_2(k) &\leq -\tau_{L(10,11)} \\
-x_{11}(k) + x_{10}(k) + \beta s^c_2(k) &\leq -\tau_{w(12,13)}(k) \\
\\
-x_{12}(k) + x_6(k) + \beta s^c_1(k) &\leq -\tau_{w(6,12)}(k) \\
-x_{12}(k) + x_{11}(k) + \beta s^c_2(k) &\leq -\tau_{w(11,12)}(k)
\end{aligned} \tag{5-39}$$

At last, to make the variable routing complete we have to add 2 more constraints to make sure the MILP only selects one route. Obviously, a vessel cannot sail both routes at the same time. This will be done as in Eq 5-40, which will make sure that either $s^c_1(k)$ is 0 or 1 and

$s_2^c(k)$ the other way around. This can be extended to multiple routes to make sure only 1 $s_i^c(k)$ is selected.

$$\begin{aligned} s_1^c(k) + s_2^c(k) &\leq 1 \\ -s_1^c(k) - s_2^c(k) &\leq -1 \end{aligned} \tag{5-40}$$

5-3-2 Ordering constraints

The Max-Plus ordering equations defined in Section 4-3-3 will be transformed to the MILP constraints required for the optimisation, using the method presented in Section 3-8.
The Max-Plus routing control variables $s_{ij}(k)$ are transformed as earlier described in Eq. (5-37) and Eq. (5-38). The Max-Plus binary ordering control variables are transformed in the same way as is done Section 5-2-2:

$$w_{i,\mu} \approx \beta w_{i,\mu}^c \tag{5-41}$$

Where:

$$w_{i,\mu}^c = \begin{cases} 0, & \text{for } w_{i,\mu} = e \\ 1, & \text{for } w_{i,\mu} = \varepsilon \end{cases} \tag{5-42}$$

And, the adjoint can be approximated by:

$$\bar{w}_{i,\mu} \approx \beta \left(1 - w_{i,\mu}^c\right) \tag{5-43}$$

Each inequality constraint now consists of 3 decision variables. Doing this for 3 vessels we get the following inequality constraints.

MILP ordering constraint in node 3 for vessel 1 and 2:

$$\begin{aligned} -x_3(k) + x_4(k-1) + \beta w_{3,(1,2)}^c + \beta s_{1,1}(k) + \beta s_{2,1}(k-1) &\leq -\tau_{L(3,4)} \\ -x_3(k-1) + x_4(k) - \beta w_{3,(1,2)}^c + \beta s_{1,1}(k) + \beta s_{2,1}(k-1) &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-44}$$

MILP ordering constraint in node 8 for vessel 1 and 2:

$$\begin{aligned} -x_8(k) + x_9(k-1) + \beta w_{8,(1,2)}^c + \beta s_{1,2}(k) + \beta s_{2,2}(k-1) &\leq -\tau_{L(8,9)} \\ -x_8(k-1) + x_9(k) - \beta w_{8,(1,2)}^c + \beta s_{1,2}(k) + \beta s_{2,2}(k-1) &\leq -\beta - \tau_{L(8,9)} \end{aligned} \tag{5-45}$$

MILP ordering constraint in node 3 for vessel 1 and 3:

$$\begin{aligned} -x_3(k) + x_4(k-2) + \beta w_{3,(1,3)}^c + \beta s_{1,1}(k) + \beta s_{3,1}(k-2) &\leq -\tau_{l1,2} \\ -x_3(k-2) + x_4(k) - \beta w_{3,(1,3)}^c + \beta s_{1,1}(k) + \beta s_{3,1}(k-2) &\leq -\beta - \tau_{l1,2} \end{aligned} \tag{5-46}$$

MILP ordering constraint in node 8 for vessel 1 and 3:

$$\begin{aligned} -x_8(k) + x_9(k-2) + \beta w_{8,(1,3)}^c + \beta s_{1,2}(k) + \beta s_{3,2}(k-2) &\leq -\tau_{L(8,9)} \\ -x_8(k-2) + x_9(k) - \beta w_{8,(1,3)}^c + \beta s_{1,2}(k) + \beta s_{3,2}(k-2) &\leq -\beta - \tau_{L(8,9)} \end{aligned} \tag{5-47}$$

MILP ordering constraint in node 3 for vessel 2 and 3:

$$\begin{aligned} -x_3(k-1) + x_4(k-2) + \beta w_{3,(2,3)}^c + \beta s_{2,1}(k) + \beta s_{3,1}(k-2) &\leq -\tau_{L(3,4)} \\ -x_3(k-2) + x_4(k-1) - \beta w_{3,(2,3)}^c + \beta s_{2,1}(k) + \beta s_{3,1}(k-2) &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-48}$$

MILP ordering constraint in node 8 for vessel 2 and 3:

$$\begin{aligned} -x_8(k-1) + x_9(k-2) + \beta w^c_{8,(2,3)} + \beta s_{2,2}(k) + \beta s_{3,2}(k-2) &\leq -\tau_{L(8,9)} \\ -x_8(k-2) + x_9(k-1) - \beta w^c_{8,(2,3)} + \beta s_{2,2}(k) + \beta s_{3,2}(k-2) &\leq -\beta - \tau_{L(8,9)} \end{aligned} \quad (5\text{-}49)$$

Table 5-2: Definitions of the 6 conventional routing control variables for the example case

Vessel number	Route	Conventional routing control variable
Vessel 1 (k)	route 1	$s^c_{1,1}(k)$
Vessel 1 (k)	route 2	$s^c_{1,2}(k)$
Vessel 2 $(k-1)$	route 1	$s^c_{2,1}(k-1)$
Vessel 2 $(k-1)$	route 2	$s^c_{2,2}(k-1)$
Vessel 3 $(k-2)$	route 1	$s^c_{3,1}(k-2)$
Vessel 3 $(k-2)$	route 2	$s^c_{3,2}(k-2)$

Table 5-3: Definitions of the 6 conventional ordering control variables for the example case

Ordering between vessels	Ordering at node	Conventional ordering control variable
Vessel 1 (k) and 2 $(k-1)$	3	$w^c_{3,(1,2)}$
Vessel 1 (k) and 2 $(k-1)$	8	$w^c_{8,(1,2)}$
Vessel 1 (k) and 3 $(k-2)$	3	$w^c_{3,(1,3)}$
Vessel 1 (k) and 3 $(k-2)$	8	$w^c_{8,(1,3)}$
Vessel 2 $(k-1)$ and 3 $(k-2)$	3	$w^c_{3,(2,3)}$
Vessel 2 $(k-1)$ and 3 $(k-2)$	8	$w^c_{8,(2,3)}$

As can be seen, from Table 5-3, to model 3 vessels we require 6 control variables, 3 for each ordering location. As explained in Section 3-6-2, the number of control variables not only depends on the number of vessels but also on the number of locks. Thus we get: We require :

$$L\frac{n(n-1)}{2} \quad (5\text{-}50)$$

With n number of vessels and L number of locks

5-4 Bi-Directional Fixed Routing case

This section will define the MILP constraints for the Bi-Directional Fixed Routing case. This is first done in Section 5-4-1 for the routing constraints, after which the ordering constraints are derived in Section 5-4-2.

5-4-1 Routing constraints

The Max-Plus routing equations defined in Section 4-4-2 will be transformed to the MILP constraints required for the optimisation, using the method presented in Section 3-8. This will be done in the same way as is done in Section 5-3-1, because this time we do have the Max-Plus binary control variable $s_{i,j}(k)$ in the routing equations again for the upstream and downstream case. Thus we use the following transformation, with $\beta \ll 0$:

$$s^c_{D,U} = \begin{cases} 0, & \text{for } s_{D,U} = e \\ 1, & \text{for } s_{D,U} = \varepsilon \end{cases} \quad \text{and} \quad s^c_{D,U} = \begin{cases} 0, & \text{for } s_{D,U} = e \\ 1, & \text{for } s_{D,U} = \varepsilon \end{cases} \tag{5-51}$$

$$s_D \approx \beta s^c_D \quad \text{and} \quad s_U \approx \beta s^c_U \tag{5-52}$$

Firstly, for going *downstream* from state $x_1(k)$ to state $x_6(k)$, we have the MILP routing inequality constraints for vessel k as defined in Eq. (5-53):

$$\begin{aligned} -x_1(k) + s^c_D(k)\beta &\leq -u_1(k) \\ -x_2(k) + x_1(k) + s^c_D(k)\beta &\leq -\tau_{w(1,2)}(k) \\ -x_3(k) + x_2(k) + s^c_D(k)\beta &\leq -\tau_{L(2,3)} \\ -x_4(k) + x_3(k) + s^c_D(k)\beta &\leq -\tau_{L(3,4)} \\ -x_5(k) + x_4(k) + s^c_D(k)\beta &\leq -\tau_{L(4,5)} \\ -x_6(k) + x_5(k) + s^c_D(k)\beta &\leq -\tau_{w(5,6)}(k) \end{aligned} \tag{5-53}$$

Secondly, for going *upstream* from state $x_6(k)$ to state $x_1(k)$, we have the MILP routing inequality constraints for vessel k as defined in Eq. (5-54):

$$\begin{aligned} x_2(k) - x_1(k) + s^c_U(k)\beta &\leq -\tau_{w(2,1)}(k) \\ x_3(k) - x_2(k) + s^c_U(k)\beta &\leq -\tau_{L(3,2)} \\ x_4(k) - x_3(k) + s^c_U(k)\beta &\leq -\tau_{L(4,3)} \\ x_5(k) - x_4(k) + s^c_U(k)\beta &\leq -\tau_{L(5,4)} \\ x_6(k) - x_5(k) + s^c_U(k)\beta &\leq -\tau_{w(6,5)}(k) \\ -x_6(k) + s_U(k)\beta &\leq -u_6(k) \end{aligned} \tag{5-54}$$

At last, to make the routing complete we have to add 2 more constraints to make sure the MILP only selects one direction. Obviously, a vessel cannot sail both upstream and downstream at the same time. This will be done as in Eq 5-55, which will make sure that either $s^c_U(k)$ is 0 or 1 and $s^c_D(k)$ the other way around. This can be extended to multiple routes to make sure only 1 $s^c_i(k)$ is selected.

$$\begin{aligned} s^c_U(k) + s^c_D(k) &\leq 1 \\ -s^c_U(k) - s^c_D(k) &\leq -1 \end{aligned} \tag{5-55}$$

5-4-2 Ordering constraints

The Max-Plus ordering equations defined in Section 4-4-3 will be transformed to the MILP constraints required for the optimisation, using the method presented in Section 3-8. Again, the Max-Plus routing control variables $s_{D,U}(k)$ are transformed as follows:

$$s^c_{D,U} = \begin{cases} 0, & \text{for } s_{D,U} = e \\ 1, & \text{for } s_{D,U} = \varepsilon \end{cases} \quad \text{and} \quad s^c_{D,U} = \begin{cases} 0, & \text{for } s_{D,U} = e \\ 1, & \text{for } s_{D,U} = \varepsilon \end{cases} \tag{5-56}$$

$$s_D \approx \beta s^c_D \quad \text{and} \quad s_U \approx \beta s^c_U \tag{5-57}$$

And, the Max-Plus binary ordering control variables, for $w_{i,DD,\mu}$, $w_{i,UU,\mu}$, $w_{i,DU,\mu}$, and $w_{i,UD,\mu}$, are transformed as follows:

$$w_{i,\mu} = \beta w^c_{i,\mu} \tag{5-58}$$

Where:

$$w^c_{i,\mu} = \begin{cases} 0, & \text{for } w_{i,\mu} = e \\ 1, & \text{for } w_{i,\mu} = \varepsilon \end{cases} \tag{5-59}$$

And, the adjoint can be approximated by:

$$\bar{w}_{i,\mu} = \beta \left(1 - w^c_{i,\mu}\right) \tag{5-60}$$

with $\beta \ll 0$. Then transforming the Max-Plus ordering equations defined in Section 4-4-3 will result in 4 pairs of 2 constraints for each vessel combination, a pair for both vessels going downstream, a pair for both vessels going upstream, a pair for vessel k going downstream and $k-1$ going upstream, and lastly, a pair for vessel k going upstream and $k-1$ going downstream. These 8 constraints will be presented for vessel 1 and 2 in node 3.
For both vessels downstream:

$$\begin{aligned} -x_3(k) + x_4(k-1) + \beta w^c_{3,DD,1} + \beta s_D(k) + \beta s_D(k-1) &\leq -\tau_{L(3,4)} \\ -x_3(k-1) + x_4(k) - \beta w^c_{3,DD,1} + \beta s_D(k) + \beta s_D(k-1) &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-61}$$

For both vessels upstream:

$$\begin{aligned} -x_4(k) + x_3(k-1) + \beta w^c_{3,UU,1} + \beta s_U(k) + \beta s_U(k-1) &\leq -\tau_{L(3,4)} \\ -x_4(k-1) + x_3(k) - \beta w^c_{3,UU,1} + \beta s_U(k) + \beta s_U(k-1) &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-62}$$

For vessel k downstream and vessel $k-1$ upstream:

$$\begin{aligned} -x_3(k) + x_3(k-1) + \beta w^c_{3,DU,1} + \beta s_D(k) + \beta s_U(k-1) &\leq -\tau_{safety} \\ -x_4(k-1) + x_4(k) - \beta w^c_{3,DU,1} + \beta s_D(k) + \beta s_U(k-1) &\leq -\beta - \tau_{safety} \end{aligned} \tag{5-63}$$

For vessel k upstream and vessel $k-1$ downstream:

$$\begin{aligned} -x_4(k) + x_4(k-1) + \beta w^c_{3,UD,1} + \beta s_U(k) + \beta s_D(k-1) &\leq -\tau_{safety} \\ -x_3(k-1) + x_3(k) - \beta w^c_{3,UD,1} + \beta s_U(k) + \beta s_D(k-1) &\leq -\beta - \tau_{safety} \end{aligned} \tag{5-64}$$

5-5 Bi-Directional Variable Routing case

This section will define the MILP constraints for the Bi-Directional Fixed Routing case. This is first done in Section 5-5-1 for the routing constraints, after which the ordering constraints are derived in Section 5-5-2.

5-5-1 Routing constraints

The Max-Plus routing equations defined in Section 4-5-2 will be transformed to the MILP constraints required for the optimisation, using the method presented in Section 3-8. This will be done in the same way as is done in Section 5-3-1, because this time we do have the Max-Plus binary control variable $s_{i,j}(k)$ in the routing equations again for the upstream and downstream case and also for selecting the route. We use the following transformation:

$$s_i^c = \begin{cases} 0, & \text{for } s_i = e \\ 1, & \text{for } s_i = \varepsilon \end{cases} \tag{5-65}$$

$$s_i \approx \beta s_i^c \tag{5-66}$$

With $\beta \ll 0$ and $i \in [1, 2, 3, 4]$

Firstly, for going *downstream* via left *route 1* from state $x_1(k)$ to state $x_{12}(k)$, we have the MILP routing inequality constraints for vessel k as defined in Eq. (5-67):

$$\begin{aligned}
-x_1(k) + \beta s_1^c &\leq -u_1(k) \\
-x_2(k) + x_1(k) + \beta s_1^c(k) &\leq -\tau_{w(1,2)}(k) \\
-x_3(k) + x_2(k) + \beta s_1^c(k) &\leq -\tau_{L(2,3)} \\
-x_4(k) + x_3(k) + \beta s_1^c(k) &\leq -\tau_{L(3,4)} \\
-x_5(k) + x_4(k) + \beta s_1^c(k) &\leq -\tau_{L(4,5)} \\
-x_6(k) + x_5(k) + \beta s_1^c(k) &\leq -\tau_{w(5,6)}(k) \\
-x_{12}(k) + x_6(k) + \beta s_1^c(k) &\leq -\tau_{w(6,12)}(k)
\end{aligned} \tag{5-67}$$

Secondly, for going *downstream* via right *route 2* from state $x_1(k)$ to state $x_{12}(k)$, we have the MILP routing inequality constraints for vessel k as defined in Eq. (5-68):

$$\begin{aligned}
-x_1(k) + \beta s_2^c &\leq -u_1(k) \\
-x_7(k) + x_1(k) + \beta s_2^c(k) &\leq -\tau_{w(1,7)}(k) \\
-x_8(k) + x_7(k) + \beta s_2^c(k) &\leq -\tau_{L(7,8)} \\
-x_9(k) + x_8(k) + \beta s_2^c(k) &\leq -\tau_{L(8,9)} \\
-x_{10}(k) + x_9(k) + \beta s_2^c(k) &\leq -\tau_{L(10,11)} \\
-x_{11}(k) + x_{10}(k) + \beta s_2^c(k) &\leq -\tau_{w(12,13)}(k) \\
-x_{12}(k) + x_{11}(k) + \beta s_2^c(k) &\leq -\tau_{w(11,12)}(k)
\end{aligned} \tag{5-68}$$

Thirdly, for going *upstream* via left *route 3* from state $x_{12}(k)$ to state $x_1(k)$, we have the MILP routing inequality constraints for vessel k as defined in Eq. (5-69):

$$\begin{aligned}
x_2(k) - x_1(k) + s_3^c(k)\beta &\leq -\tau_{w(2,1)}(k) \\
x_3(k) - x_2(k) + s_3^c(k)\beta &\leq -\tau_{L(3,2)} \\
x_4(k) - x_3(k) + s_3^c(k)\beta &\leq -\tau_{L(4,3)} \\
x_5(k) - x_4(k) + s_3^c(k)\beta &\leq -\tau_{L(5,4)} \\
x_6(k) - x_5(k) + s_3^c(k)\beta &\leq -\tau_{w(6,5)}(k) \\
x_{12}(k) - x_6(k) + s_3^c(k)\beta &\leq -\tau_{w(12,6)}(k) \\
-x_{12}(k) + s_3^c(k)\beta &\leq -u_{12}(k)
\end{aligned} \tag{5-69}$$

Fourthly, for going *upstream* via right *route 4* from state $x_{12}(k)$ to state $x_1(k)$, we have the MILP routing inequality constraints for vessel k as defined in Eq. (5-70):

$$\begin{aligned} x_7(k) - x_1(k) + \beta s_4^c(k) &\leq -\tau_{w(7,1)}(k) \\ x_8(k) - x_7(k) + \beta s_4^c(k) &\leq -\tau_{L(8,7)} \\ x_9(k) - x_8(k) + \beta s_4^c(k) &\leq -\tau_{L(9,8)} \\ x_{10}(k) - x_9(k) + \beta s_4^c(k) &\leq -\tau_{L(10,9)} \\ x_{11}(k) - x_{10}(k) + \beta s_4^c(k) &\leq -\tau_{w(11,10)}(k) \\ x_{12}(k) - x_{11}(k) + \beta s_4^c(k) &\leq -\tau_{w(12,11)}(k) \\ -x_{12}(k) + s_4^c(k)\beta &\leq -u_{12}(k) \end{aligned} \tag{5-70}$$

At last, to make the routing complete we have to add 2 more constraints to make sure the MILP only selects one direction. Obviously, a vessel cannot sail both upstream and downstream at the same time. This will be done as in Eq. 5-71, which will make sure that either $s_i^c(k)$ is 0 or 1 and $s_i^c(k)$ the other way around. This can be extended to multiple routes to make sure only 1 $s_i^c(k)$ is selected.

$$\begin{aligned} s_1^c(k) + s_2^c(k) + s_3^c(k) + s_4^c(k) &\leq 1 \\ -s_1^c(k) - s_2^c(k) - s_3^c(k) - s_4^c(k) &\leq -1 \end{aligned} \tag{5-71}$$

5-5-2 Ordering constraints

The Max-Plus ordering equations defined in Section 4-5-3 will be transformed to the MILP constraints required for the optimisation, using the method presented in Section 3-8. Again, the Max-Plus routing control variables $s_{ij}(k)$ are transformed as follows:

$$s_i^c = \begin{cases} 0, & \text{for } s_i = e \\ 1, & \text{for } s_i = \varepsilon \end{cases} \tag{5-72}$$

$$s_i \approx \beta s_i^c \tag{5-73}$$

With $\beta \ll 0$ and $i \in [1, 2, 3, 4]$ And, the Max-Plus binary ordering control variables, for $w_{i,DD,\mu}$, $w_{i,UU,\mu}$, $w_{i,DU,\mu}$, and $w_{i,UD,\mu}$, are transformed as follows:

$$w_{i,\mu} = \beta w_{i,\mu}^c \tag{5-74}$$

Where:

$$w_{i,\mu}^c = \begin{cases} 0, & \text{for } w_{i,\mu} = e \\ 1, & \text{for } w_{i,\mu} = \varepsilon \end{cases} \tag{5-75}$$

And, the adjoint can be approximated by:

$$\bar{w}_{i,\mu} = \beta \left(1 - w_{i,\mu}^c\right) \tag{5-76}$$

with $\beta \ll 0$. Then transforming the Max-Plus ordering equations defined in Section 4-5-3 will result in 4 pairs of 2 constraints for each vessel combination and every ordering location. We get a pair for both vessels going downstream, a pair for both vessels going upstream, a

pair for vessel k going downstream and $k-1$ going upstream, and lastly, a pair for vessel k going upstream and $k-1$ going downstream. These 8 constraints will be presented for vessel 1 and 2 in node 3.
For both vessels downstream:

$$\begin{aligned} -x_3(k) + x_4(k-1) + \beta w^c_{3,DD,1} + \beta s_{3,4}(k) + \beta s_{3,4}(k-1) &\leq -\tau_{L(3,4)} \\ -x_3(k-1) + x_4(k) - \beta w^c_{3,DD,1} + \beta s_{3,4}(k) + \beta s_{3,4}(k-1) &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-77}$$

For both vessels upstream:

$$\begin{aligned} -x_4(k) + x_3(k-1) + \beta w^c_{3,UU,1} + \beta s_{4,3}(k) + \beta s_{4,3}(k-1) &\leq -\tau_{L(3,4)} \\ -x_4(k-1) + x_3(k) - \beta w^c_{3,UU,1} + \beta s_{4,3}(k) + \beta s_{4,3}(k-1) &\leq -\beta - \tau_{L(3,4)} \end{aligned} \tag{5-78}$$

For vessel k downstream and vessel $k-1$ upstream:

$$\begin{aligned} -x_3(k) + x_3(k-1) + \beta w^c_{3,DU,1} + \beta s_{3,4}(k) + \beta s_{4,3}(k-1) &\leq -\tau_{safety} \\ -x_4(k-1) + x_4(k) - \beta w^c_{3,DU,1} + \beta s_{3,4}(k) + \beta s_{4,3}(k-1) &\leq -\beta - \tau_{safety} \end{aligned} \tag{5-79}$$

For vessel k upstream and vessel $k-1$ downstream:

$$\begin{aligned} -x_4(k) + x_4(k-1) + \beta w^c_{3,UD,1} + \beta s_{4,3}(k) + \beta s_{3,4}(k-1) &\leq -\tau_{safety} \\ -x_3(k-1) + x_3(k) - \beta w^c_{3,UD,1} + \beta s_{4,3}(k) + \beta s_{3,4}(k-1) &\leq -\beta - \tau_{safety} \end{aligned} \tag{5-80}$$

Next, these 8 constraints will be presented for vessel 1 and 2 in node 8.
For both vessels downstream:

$$\begin{aligned} -x_8(k) + x_9(k-1) + \beta w^c_{8,DD,1} + \beta s_{8,9}(k) + \beta s_{8,9}(k-1) &\leq -\tau_{L(8,9)} \\ -x_8(k-1) + x_9(k) - \beta w^c_{8,DD,1} + \beta s_{8,9}(k) + \beta s_{8,9}(k-1) &\leq -\beta - \tau_{L(8,9)} \end{aligned} \tag{5-81}$$

For both vessels upstream:

$$\begin{aligned} -x_9(k) + x_8(k-1) + \beta w^c_{8,UU,1} + \beta s_{9,8}(k) + \beta s_{9,8}(k-1) &\leq -\tau_{L(8,9)} \\ -x_9(k-1) + x_8(k) - \beta w^c_{8,UU,1} + \beta s_{9,8}(k) + \beta s_{9,8}(k-1) &\leq -\beta - \tau_{L(8,9)} \end{aligned} \tag{5-82}$$

For vessel k downstream and vessel $k-1$ upstream:

$$\begin{aligned} -x_8(k) + x_8(k-1) + \beta w^c_{8,DU,1} + \beta s_{8,9}(k) + \beta s_{9,8}(k-1) &\leq -\tau_{safety} \\ -x_9(k-1) + x_9(k) - \beta w^c_{8,DU,1} + \beta s_{8,9}(k) + \beta s_{9,8}(k-1) &\leq -\beta - \tau_{safety} \end{aligned} \tag{5-83}$$

For vessel k upstream and vessel $k-1$ downstream:

$$\begin{aligned} -x_9(k) + x_9(k-1) + \beta w^c_{8,UD,1} + \beta s_{9,8}(k) + \beta s_{8,9}(k-1) &\leq -\tau_{safety} \\ -x_8(k-1) + x_8(k) - \beta w^c_{8,UD,1} + \beta s_{9,8}(k) + \beta s_{8,9}(k-1) &\leq -\beta - \tau_{safety} \end{aligned} \tag{5-84}$$

5-6 Conclusion

The goal of this Chapter 5 was to transform the SMPL systems of the four IWT cases to MILP models such that, when these MILP models are solved, an optimal schedule is generated. As stated in Chapter 1, this chapter addressed the research question:

How to transform SMPL IWT systems to MILP models?

First of all, this chapter has shown what optimal scheduling entails in the context of this research, minimising the cumulative arrival time of all vessels. Moreover, the objective function which ensures this has been defined. Next, it was thoroughly described for the four SMPL IWT cases that we can transform them to MILP models by transforming the Max-Plus routing equations and Max-Plus ordering equations to, respectively, MILP routing inequality constraints and MILP ordering inequality constraints. This is done by transforming the Max-Plus routing binary control variables and the Max-Plus ordering binary control variables to their conventional control variable counterparts by using an approximation. This results in a large set of linear inequality constraints that can be solved with fast solvers.
Before the MILP models can be used for scheduling, certain vessel information is required. The overall scheduler optimisation architecture, which includes inputs and outputs, is described in the next Chapter 6. The chapter will then also continue with the results of the four IWT cases plus some additional scenarios.

Chapter 6

Optimal Inland Waterway Transport Scheduling results

This chapter will continue with the Mixed Integer Linear Programming (MILP) Inland Waterway Transport (IWT) models derived in the previous chapter, and present the schedules they generate. This is done such that we can answer the research question, as introduced in Chapter 1:

How can we verify the designed scheduling strategy for IWT systems?

Chapter 6 is structured as follows: Firstly, Section 6-1 will introduce the architecture of the scheduler. Next, Section 6-2 will describe the scheduling results for the Uni-Directional Fixed Routing (UDFR) case. Thirdly, the scheduling results for the Uni-Directional Variable Routing (UDVR) case will be shown in Section 6-3. Furthermore, Section 6-4 will continue with the scheduling results for the Bi-Directional Fixed Routing (BDFR) case. To finalise the four cases Section 6-5 will present the results for the Bi-Directional Variable Routing (BDVR) case. At last, Section 6-6 will describe the results for the BDVR case, which is extended with locks in series to show that other waterway networks can be generated from these primary cases as well. Moreover, Section 6-7 will show some additional scenarios for this last case to show the versatility of the scheduler. Finally, the results will be summarised and concluded in Section 6-8.

6-1 Introduction to scheduler architecture and results

This section will introduce the architecture of the scheduler, which describes the implementation of the MILP optimisation model. This architecture is used to generate all the results and schedules, which are presented in the remaining sections of this chapter. The overall architecture of the MILP optimisation model can be seen in Figure 6-1, which consist of three parts the input, the scheduler and the output. These three will be elaborated on in Section 6-1-1, Section 6-1-2 and Section 6-1-3, respectively. Moreover, this section will present the method used for sharing the results and schedules. An important remark to make is that all data used in this chapter is arbitrary and chosen, such that interesting cases to verify the scheduling strategy can be generated. The data does not resemble an actual waterway network or actual vessel sailing times. This will not be a problem, as the scheduling strategy will not be validated on real-world data. These arbitrarily chosen values are sufficient to show the working principles of the scheduling strategy. However, logical values were chosen; for instance, the sailing time of a vessel on a waterway will always be larger than the sailing time of a vessel within a waiting area if the vessel does not have to wait in the waiting area. At last, as the values in this chapter are not based upon real-world data, there are also no time units defined for all the timings. Therefore, this chapter will refer to 'time units' as the time unit. This unit of time could be replaced with minutes or hours; however, it was chosen not to do this since it could be illogical and cause confusion when compared to real-world data. Note regarding the result sharing method; each new result case will start on a new page, so that the waterway graph and input data table are on a single page, and the schedule figure and output data table are on a single page as well. This will improve readability.

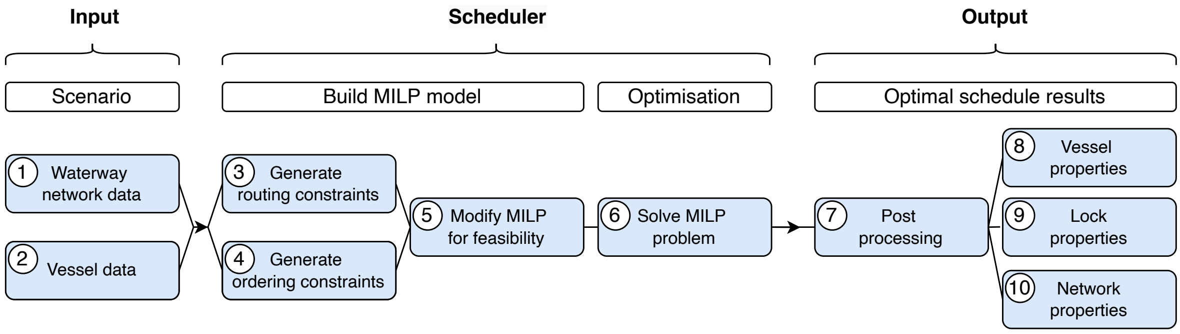


Figure 6-1: Schematic overview of the scheduler optimisation architecture

6-1-1 Scheduler input

In Table 6-1 is shown what is required to be fed into the scheduler to generate results and optimal schedules. This input can be through a general MATLAB function, .mat file or excel file. The scheduler input can be split into waterway network data and vessel data; both types will be explained in further detail.

Table 6-1: Scheduler input

1) Waterway network data	**2) Vessel data**
1) Topology matrix T	1) Number of vessels sailing the network $k_{\max}$
2) Object locations in network	2) Vessel sailing times on all waterways $\tau_{w(i,j)}(k)$
3) Object processing times $\tau_{L(i,j)}$	3) Departure location $d(k)$
4) Possible departure locations	4) Departure time $u_i(k)$
5) Possible arrival locations	5) Arrival deadlines $a_i(k)$
	6) Vessel priority $\sigma(k)$

1) Waterway network data

Before generating vessel scheduling plans, we must first define where we are scheduling the vessels on. Therefore, we must create the waterway network the vessels are sailing. To generate the waterway network on which the vessel will sail, we need to establish the network graph. This can be done by defining the topology matrix T of the graph as was presented in Section 3-4. Next, we need to define the object locations in the waterway network by defining which arcs and nodes belong to locks and waterways. As was shown in Section 4-1, every waterway consist of two nodes connected by one arc and every lock consists of four nodes with three arcs connecting them in series. After having defined the object locations, we have to specify the object processing times $\tau_{L(i,j)}$. The chosen values will be presented in the results through a table as shown in Table 6-25, where the cells marked with an 'x' can be any arbitrary timing in 'time units'. At last, we should define the set of nodes in the waterway network from which vessels can depart and where the vessels can arrive. The examples presented in the remaining part of this chapter will include just a single departure location and a single arrival location.

Table 6-2: Example table of waterway network data required by the scheduler

Operations	**Vessel** 1	...	**Vessel** $k_{\max}$
$\tau_{L(i,j)}$	x	...	x

2) Vessel data

Subsequently, we will elaborate on the vessel input data. We first have to define the number of vessels sailing the waterway network and the individual vessel sailing times $\tau_{w(i,j)}(k)$ on all waterways. These vessel sailing times are the times that it takes the vessel to sail those particular waterways which they will provide the scheduler at the start of the optimisation. As mentioned in Section 4-1, we assume these vessel sailing times to be known and constant. Thirdly, we must define the departure location $d(k)$. By specifying the departure location, the arrival location (the other side of the network as we do not have multiple arrivals and departure locations) will also be fixed. Moreover, by specifying the departure location, we will also immediately determine the direction in which a vessel will sail, either upstream or downstream. Next, to the departure location, the departure time $u_i(k)$, must be defined. As mentioned earlier in Section 4-1, we cannot control the departure time of the vessel, which is something that is often possible in Switching Max-Plus Linear (SMPL) systems. Fifthly, we have to specify the arrival deadlines of all the vessels, which will be a hard constraint for the scheduler. The chosen values will be presented in the results through a table as shown in Table 6-3, where the cells marked with an 'x' can be any arbitrary timing in 'time units'. Again, as mentioned in the introduction of this section, the values used later on in this chapter are chosen arbitrarily and do not resemble actual sailing times. At last, the vessel priority can be specified by defining the output weight function $\sigma(k)$ for the objective function. For instance, consider a Uni-Directional case, where we have vessel 2 which has a special type of cargo (e.g. fresh products). This cargo makes it for vessel 2 more critical to arrive at the arrival location faster than vessel 1. In that case, we can define $\sigma(k)$ as follows:

$$\sigma_{\mathrm{UD}}(1) = \begin{bmatrix} 10^{-4} & \dots & 10^{-4} & \dots & 10^{-4} & \sigma_a(1) \end{bmatrix}, \sigma_{\mathrm{UD}}(k) \in \mathbb{R}^{1\times n} \tag{6-1}$$

and

$$\sigma_{\text{UD}}(2) = \left[\begin{array}{ccccccc} 10^{-4} & \dots & 10^{-4} & \dots & 10^{-4} & \sigma_a(2) \end{array}\right], \sigma_{\text{UD}}(k) \in \mathbb{R}^{1\times n} \tag{6-2}$$

With $\sigma_a(2) > \sigma_a(1)$.

Table 6-3: Example table of vessel data required by the scheduler

Operations	**Vessel** 1	...	**Vessel** $k_{\max}$
$\tau_{w(i,j)}(k)$	x	...	x
$d(k)$	x	...	x
$u_i(k)$	x	...	
$\vdots$			
$u_{16}(k)$	x	...	x
$a_i(k)$	x	...	x
$\vdots$			
$a_{16}(k)$	x	...	x

In the remaining part of this chapter where the results and schedules are shown, the waterway network data table and vessel data table will be combined to a single input table and will be presented for every case.

6-1-2 MILP model generation

This section will describe what is being computed by the scheduler in order to generate the result and optimal schedules. Since this is actually described throughout the whole Chapter 5, this section will be relatively short. Firstly, the routing constraints are computed as is explained in Section 3-8 and are presented in the routing sections of Chapter 5. These are generated based on just the topology matrix T for all possible routes in the network and for all vessels. Secondly, the ordering constraints are also computed as is explained in Section 3-8 and are presented in the ordering sections of Chapter 5. These are generated for all ordering combinations of all possible vessel pairs at every ordering location (i.e. at every lock in the waterway network). We get for the Uni-Directional cases 2 constraints per vessel permutation for every ordering location, and for the Bi-Directional case 8 constraints per vessel permutation for every location. Thirdly, the MILP model is modified for feasibility and additional constraints are added, such as; parameterisation of the routing control variables as described in Section 3-6-1, ensuring only one route is chosen per vessel, as shown in Eq (5-71) and ensuring vessels can only go either upstream or downstream. Moreover, a constraint that allows for picking the starting location of individual vessels is added. At last, the MILP model is solved, and outputs are computed. For all cases discussed in this chapter, the commercial solver GUROBI [21] with an academic license was used. GUROBI primarily uses a Branch-And-Bound and Cutting-Planes algorithm to find a feasible solution. However, to solve such significant size problems within a reasonable computation time, the solver also uses a few additional heuristic techniques as well.

6-1-3 Scheduler output

In Table 6-4 is shown what the scheduler produces after the MILP model is solved. The scheduler output can be split into; the vessel schedule and properties, the object schedule and properties, and the global network properties. All three will be explained in further detail. However, first it is good to mention that the scheduler will only return a schedule if the MILP model is feasible for the given input data. Naturally, if an arrival deadline $a_i(k)$ is chosen to strict, and vessel k cannot make it regardless of what other vessel are doing, it will result in a infeasible schedule.
As can be seen in Section 5-1, in Eq (5-22) the MILP model will return a single vector $\begin{bmatrix} x(k) & v(k) \end{bmatrix}^T$ with all the optimised states $x(k)$ and binary control variables vector $v(k)$. As this is a single vector, some post-processing is required. First, all the states and binary control variables are collected from the vector. Next, the vessel state evolutions and lock state evolutions are created. By using the binary control variables, the route for each vessel can be determined, and the overall vessel ordering can be derived. At last, the vessel delays are calculated with respect to a baseline scenario, in which the optimal schedule is created for the case the vessels would be sailing the network alone, and therefore there are no delays caused by other vessels.

Table 6-4: Scheduler output

8) Vessel schedule and properties	**9) Object schedule and properties**	**10) Global network properties**
1) Optimal routes	1) Optimal operation times for lock i	1) Cumulative arrival time $\mathcal{A}_{\text{net}}$
2) Destination arrival time $x_a(k)$	2) Number of lock levellings $\mathcal{L}_{\text{i}}$	2) Cumulative delays $\mathcal{D}_{\text{net}}$
3) Intermediate node arrival times $x_i(k)$	3) Number of empty lock levellings $\mathcal{L}_{\text{e,i}}$*	3) Make-span $C_{\max}$
4) Delays at lock i for vessel k is $\Delta_{\text{lock,i}}(k)$	4) Occupancy at time unit $\mathcal{O}_{\text{lock,i}}(t)$*	5) Bottleneck $\mathcal{B}_{\text{net}}$
5) Intermediate node departure times*	5) Queue order at time unit $\mathcal{Q}_{\text{lock,i}}(t)$*	

* Inferred information from model output

Note some outputs follow directly from the scheduler, and other information can be inferred from the scheduler output. This inferred information is denoted with an asterisk.

8) Vessel schedule and properties

The main outputs from the scheduler for the vessels are the; destination arrival time, intermediate node arrival times and the optimal route in the case of the Variable Routing cases. These are presented in a table of the form of Table 6-5 and visualised in a figure of the form Figure 6-2. Moreover, we compare the generated optimal vessel schedule (with the influence of other vessels) with the schedule for when a vessel would be sailing the waterway network alone (without the influence of other vessels) to determine the delays $\Delta_{\text{lock},i}(k)$, which are presented in a table of the form of Table 6-6. At last, when delays occur, we can also infer the intermediate node departure times from Figure 6-2. For example, we can see in the figure that the arrival time at node 3 is $x_3(1) = 30$, the delay at lock 1 is $\Delta_{\text{lock},1}(1) = 9$ (as the vessel is not moving and waiting in the waiting area), then the departure time at node 3 follows $x_3(1) = 39$.

Table 6-5: Example table of vessel schedule produced by the scheduler, in case vessel k is sailing downstream and vessel $k_{\max}$ is sailing upstream

States	Vessel 1	...	Vessel $k_{\max}$
Route	$\mathrm{x_{route}}$	...	$\mathrm{x_{route}}$
$x_{i,\text{first}}(k)$	$x_i(k)$	...	$x_a(k_{\max})$
$\vdots$	$\vdots$		$\vdots$
$x_{i,\text{last}}(k)$	$x_a(k)$	...	$x_i(k_{\max})$

Table 6-6: Example table of vessel delays produced by the scheduler

Objects	Vessel 1	...	Vessel $k_{\max}$
Lock 1	$\Delta_{\text{lock},1}(1)$	...	$\Delta_{\text{lock},1}(k_{\max})$

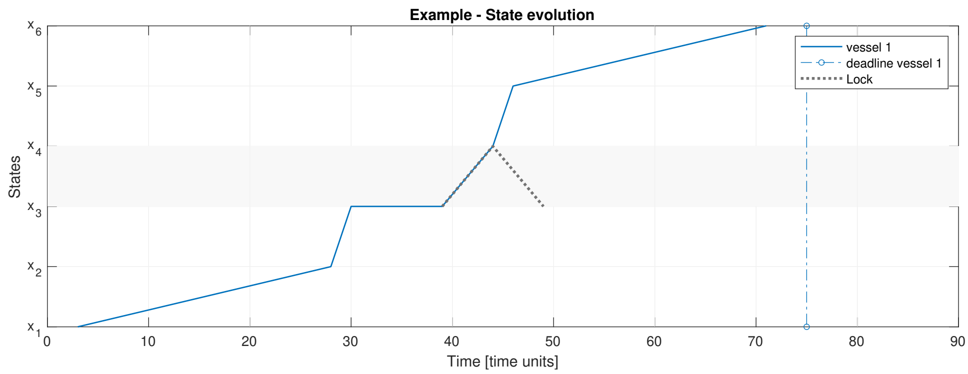


Figure 6-2: Example visualisation of vessel schedule produced by the scheduler

9) Object schedule and properties

The main outputs from the scheduler for the objects are; the optimal operation times and the number of lock levellings $\mathcal{L}_\text{i}$. These are presented in a table of the form of Table 6-7 and visualised in a figure of the form Figure 6-3. This table and figure show in time units when the lock should move upstream and when the lock should move downstream. For this example the number of lock levelling is $\mathcal{L}_1 = 11$. This lock schedule, as visualised in Figure 6-3 is also shown in Figure 6-2. Since in this research, the focus is on the IWT infrastructure perspective, it is convenient to show the lock schedule solely as well. This could, for example, be shared with a lock operator, so he\she knows the schedule for a certain length of time. Moreover, from Figure 6-2 we can infer the number of empty lock levellings $\mathcal{L}_{\text{e},1}$, which is defined as the number of lock operations when no vessel occupies the lock. Thus, the case when the lock has to go to the other water level to pick up the vessel on the other side, while no vessel is in the lock. Although, not very informative, as we only show the schedule of a single vessel, but for Figure 6-2, we would get $\mathcal{L}_{\text{e},1} = 1$. Additionally, we can infer the occupancy for lock i at time unit t with $\mathcal{O}_{\text{lock,i}}(t)$. For instance, $\mathcal{O}_{\text{lock},1}(35) =$ vessel 1. At last, we can see the queue order and queue length for lock i at time unit t with $\mathcal{Q}_{\text{lock,i}}(t)$. For example, $\mathcal{Q}_{\text{lock},1}(40) =$ vessel 1. Actually, we could also determine the throughput time for each lock, but as we have assumed in this research that the lock operation times are all the same, this is not an interesting insight and the throughput of all locks would be the same.

Table 6-7: Example table of lock schedule produced by the scheduler

Objects	Direction	Timings
Lock 1	Upstream	39
	Downstream	44

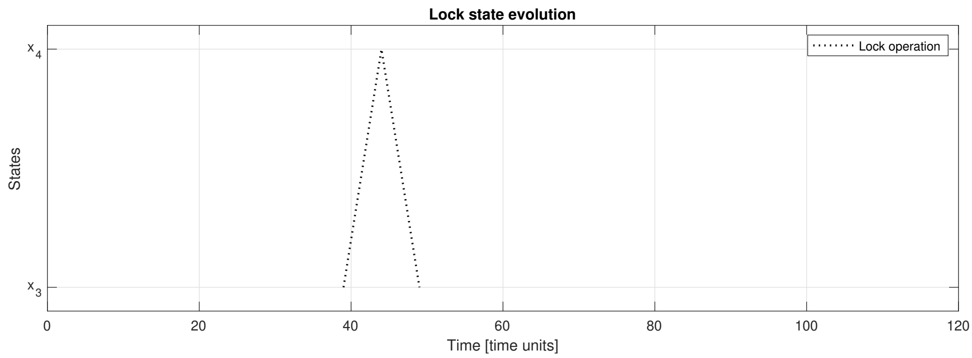


Figure 6-3: Example visualisation of vessel schedule produced by the scheduler

10) Global network properties

At last, we can derive several global waterway network properties from the scheduler, which are summarised below. With these, we can analyse the network for efficiency and possibly compare different scenarios in future research.

The cumulative arrival times $\mathcal{A}_{\text{net}}$, is defined as the sum of all individual vessel destination arrival times $x_a(k)$:

$$\mathcal{A}_{\text{net}} = \sum_{k=1}^{k_{\max}} x_a(k) \tag{6-3}$$

The cumulative vessel delays $\mathcal{D}_{\text{net}}$, is defined as the sum of all individual vessel delays at locks $\Delta(k)$:

$$\mathcal{D}_{\text{net}} = \sum_{k=1}^{k_{\max}} \Delta(k) \tag{6-4}$$

The make-span $C_{\max}$ uses the definition as described in Section 3-7 and as is often used in other scheduling literature. The maximum of all vessel arrival times:

$$\mathcal{C}_{\max} = \max\left(\{x_a\left(k\right), \ldots, x_a(k_{\max})\}\right) \tag{6-5}$$

The bottleneck $\mathcal{B}_{\text{net}}$ is defined as the lock where most vessel delays occur. This is defined as the maximum of the sum of all vessel delays for each lock:

$$\mathcal{B}_{\text{net}} = \max\left(\left\{\sum_{k=1}^{k_{\max}} \Delta_{\text{lock,i}}(k), \ldots, \sum_{k=1}^{k_{\max}} \Delta_{\text{lock,i}_{\max}}(k)\right\}\right) \tag{6-6}$$

6-2 Uni-Directional Fixed Routing case

This section will present the scheduler results for the UDFR case. The scheduler consist of the MILP model formulation as presented in 5-1, the SMPL equations as described in Section 4-2, which are transformed to MILP model constraints, as described in Section 5-2. This section will present the results when six vessels are sailing through the network.
As described in Section 6-1, first the inputs; the waterway network data and the vessel data, have to be defined. The waterway network graph results from the following topology matrix:

$$T_{\text{UDFR}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6-7}$$

This produces the waterway network of Figure 6-4, where the blue arcs represent the waterways and the red arcs represent the lock area, with the solid line illustrating the waiting areas and the dotted line illustrating the lock itself. The arc weights (i.e. sailing timings $\tau_{w(i,j)}(k)$ and operation timings $\tau_{L(i,j)}$) and the nodes (i.e. states $x_i(k)$) are also shown in the figure. The practical definitions of the states follow the description presented in Section 4-1. To recap, the definitions of all states are given in Table 6-8 and of all timings in Table 6-9. Note, to keep the size of the thesis manageable these definitions tables will not be shown for the cases in the remainder of this chapter. It can be seen that the waterway network consists of one lock between node 3 and 4 (also denoted with the squared markers). To repeat, overtaking within the locks is not possible. All vessels will start at $x_1(k)$ and will travel through the network to either $x_6(k)$. As we have vessels sailing only downstream and no routing decision has to be made by the scheduler, we have the UDFR case. Next, we define the vessel specific waterway sailing timings $\tau_{w(i,j)}(k)$ and the lock specific levelling timings $\tau_{L(i,j)}$, which are shown in Table 6-10. Moreover, the vessel specific departure location $d(k)$, the departure time $u_i(k)$ and arrival deadline $a_i(k)$ are given as well.
It can be seen in Table 6-10 that we simulate a scenario in which vessels 2, 4, and 6 are faster than vessels 1, 3 and 5, because the sailing times $\tau_{w(i,j)}(k)$ of vessel 2, 4 and 5 are less. The vessel priorities $\sigma(k)$ are equal for all six vessel as well. At last, the vessels depart one time unit after the preceding vessel, with vessel 1 departing at time unit 1.

Table 6-8: Definitions of state variables $x_i(k)$ of Figure 6-4

State event variable	Description
$x_1(k)$	Time instant when vessel k enters waterway 1
$x_2(k)$	Time instant when vessel k enters waiting area 1 and leaves waterway 1
$x_3(k)$	Time instant when vessel k enters the lock and leaves waiting area 1
$x_4(k)$	Time instant when vessel k enters waiting area 2 and leaves the lock
$x_5(k)$	Time instant when vessel k enters waterway 2 and leaves waiting area 2
$x_6(k)$	Time instant when vessel k leaves waterway 2

Table 6-9: Definitions of sailing times $\tau_{w(i,j)}(k)$ and lock operation times $\tau_{L(i,j)}$ of Figure 4-1

Timings	Description
$\tau_{w(1,2)}(k)$	Sailing time of vessel k from the start of waterway 1 to the end of waterway 1
$\tau_{L(2,3)}$	Operation time from the start of waiting area 1 to the end of waiting area 1
$\tau_{L(3,4)}$	Lock operation (i.e. drainage) time to level the water to the other side of the lock
$\tau_{L(4,5)}$	Operation time from the start of waiting area 2 to the end of waiting area 2
$\tau_{w(5,6)}(k)$	Sailing time of vessel k from the start of waterway 2 to the end of waterway 2

Table 6-10: The vessel data and object data for the UDFR case with locks in series in time units, which includes; sailing times $\tau_{w(i,j)}(k)$, lock operation times $\tau_{L(i,j)}$, departure location $d(k)$, departure time $u_i(k)$ and arrival deadlines $a_i(k)$ for 6 vessels sailing the waterway network

Variables	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
$\tau_{w(1,2)}(k)$	25	15	25	15	25	15
$\tau_{L(2,3)}$	2	2	2	2	2	2
$\tau_{L(3,4)}$	5	5	5	5	5	5
$\tau_{L(4,5)}$	2	2	2	2	2	2
$\tau_{w(5,6)}(k)$	25	15	25	15	25	15
$d(k)$	1	1	1	1	1	1
$u_1(k)$	1	2	3	4	5	6
$a_6(k)$	65	75	85	95	105	115

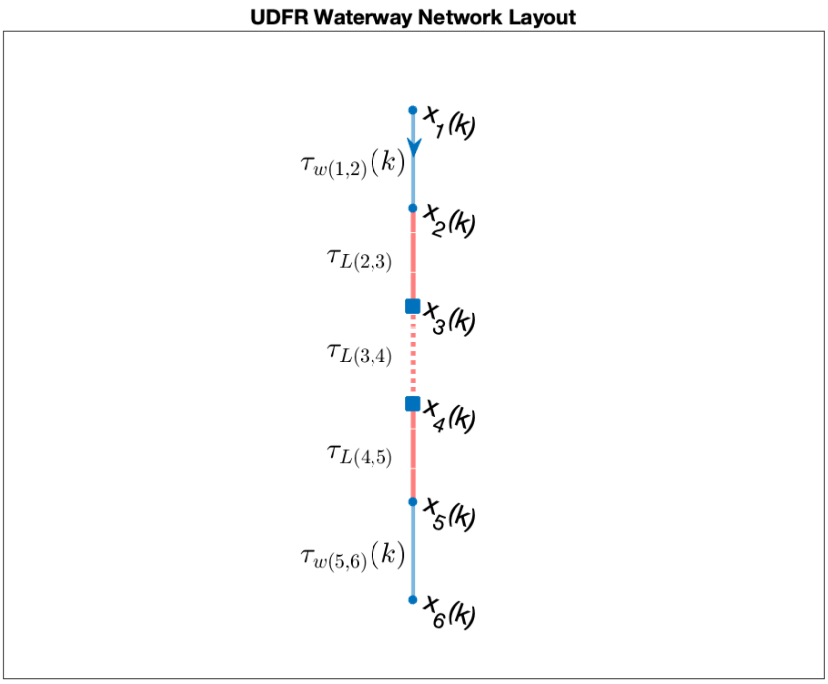


Figure 6-4: The waterway network layout for the UDFR case; with the blue arcs representing the waterways and the red arcs representing the lock area, where the solid line illustrates the waiting areas and the dotted line illustrates the lock itself

After having defined the waterway network input data and the vessel input data, the scheduler for the UDFR case was executed, and a feasible solution was found. We continue this section with the scheduler output. As described in Section 6-1-3, the scheduler output can be split into; the vessel schedule and properties, the object schedule and properties, and the global network properties. All three will be presented in the remainder of this section.

Firstly, this section will present the vessel schedule and vessel properties. The optimal schedule and subsequent state evolution of the six vessels can be seen in Table 6-11 and is visualised in Figure 6-5. Moreover, the destination and intermediate node arrival times are given. As can be seen the order of lock passing is for the lock (node 3 and 4):

$$\text{Vessel 2} \rightarrow \text{Vessel 1} \rightarrow \text{Vessel 4} \rightarrow \text{Vessel 3} \rightarrow \text{Vessel 6} \rightarrow \text{Vessel 5}$$

We would expect the faster vessels 2, 4, and 6 to overtake the other vessels, be processed by the lock earlier and arrive at their destination the earliest. However, we can see from the ordering that this is not the case. We can see that the lock does not operate purple vessel 4 and cyan vessel 6 even though they arrive before blue vessel 1. This ordering is because, otherwise, blue vessel 1 will not make its arrival deadline. Therefore, the scheduler decides to keep purple vessel 4 and cyan vessel 6 in the waiting area. Then when blue vessel 1 has passed, the lock returns to x_3 to pick up the purple vessel 4. By then, all vessels have arrived at the lock and are waiting. Again, when the lock returns to x_3 from dropping off purple vessel 4, it is expected to take cyan vessel 6 next as it is waiting for the longest. However, the scheduler decides that yellow vessel 3 should go first; otherwise, it will not make its arrival deadline. Thus, cyan vessel 6 has to wait again. We can see from the results that this new scheduling strategy will make sure the vessels reach their arrival deadline. If the scheduling strategy was based on the First-come, First-serve principle, blue vessel 1 and yellow vessel 3 would have missed their arrival deadlines. Thus we can verify from the simulations that the scheduling strategy works.

Table 6-11: The optimal IWT schedule for the UDFR case in time units, which includes; the intermediate event timings $x_i(k)$ and the arrival time at $x_6(k)$ for 6 vessels sailing the waterway network

States	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
$x_1(k)$	1	2	3	4	5	6
$x_2(k)$	26	17	28	19	30	21
$x_3(k)$	29	19	49	39	69	59
$x_4(k)$	34	24	54	44	74	64
$x_5(k)$	36	26	56	46	76	66
$x_6(k)$	61	41	81	61	101	81

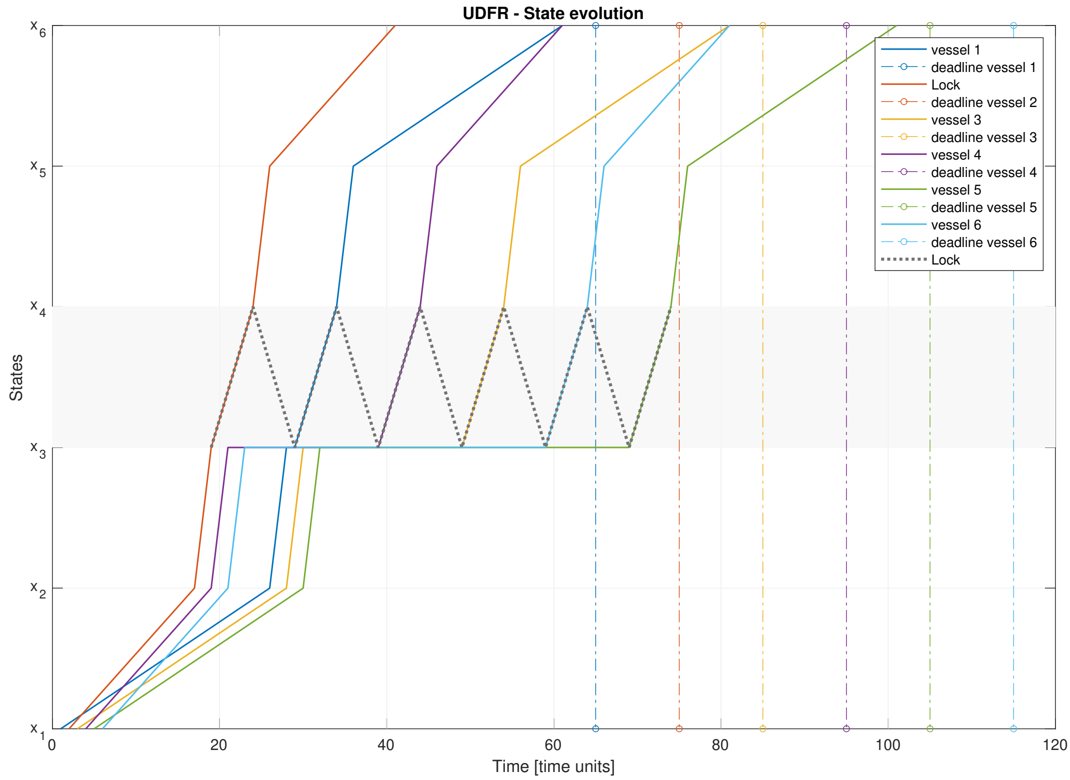


Figure 6-5: Visualisation of the optimal IWT schedule for the UDFR case shown in Table 6-11, with the grey shaded area representing the locks

Moreover, the optimal schedule results in delays for some vessels compared to when the vessel would be sailing the waterway network alone (without the influence of other vessels). The delays in time units for each vessel per lock can be seen in Table 6-12. These vessel delays are visualised in Appendix B-1 in Figure B-1.

Table 6-12: The individual vessel delays per lock for the UDFR case in time units

Objects	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Lock 1	1	0	19	18	37	36

Secondly, this section will continue with presenting the object schedule and object properties. The optimal schedule and subsequent state evolution of the lock can be seen in Table 6-13 and is visualised in Figure 6-6. We can see the number of lock levellings for the lock, with; $\mathcal{L}_1 = 11$. Moreover, we can infer the number of empty lock levellings from Figure 6-5, with; $\mathcal{L}_{\text{e},1} = 5$, which makes sense as vessels are all coming from the same direction, thus the lock has to go back empty each time. At last, we can see the occupancy for lock i at time unit t. For instance, $\mathcal{O}_{\text{lock},1}(40) =$ vessel 4. The queue order and queue length for lock i at time unit t can also be derived, for example, $\mathcal{Q}_{\text{lock},1}(40) = 3^{\text{rd}}$ vessel 5 $\rightarrow 2^{\text{nd}}$ vessel 6 $\rightarrow 1^{\text{st}}$ vessel 3.

Table 6-13: The optimal lock schedule for the UDFR case in time units, which includes for the lock the time when it should move upstream and when it should move downstream

Objects	Direction	Timings					
Lock 1	Upstream	19	29	39	49	59	69
	Downstream	24	34	44	54	64	74

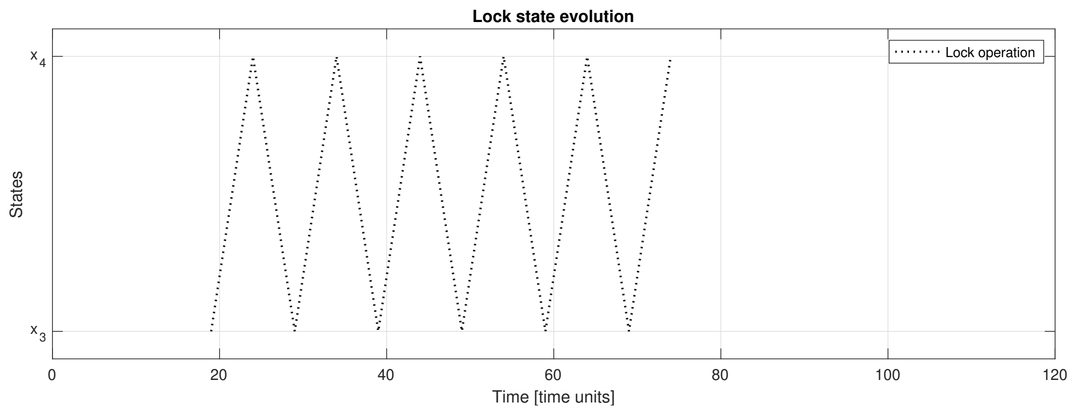


Figure 6-6: Visualisation of the optimal lock schedule for the UDFR case shown in Table 6-13

Thirdly, to conclude the results for the UDFR case, Table 6-14 shows the global network properties in time units as defined in Section 6-1. We have one route with one lock, therefore the bottleneck is rather obvious.

Table 6-14: The global network properties in time units for the UDFR case

Variable	**Value** [time units]
$\mathcal{A}_{\text{net}}$	426
$\mathcal{D}_{\text{net}}$	111
$\mathcal{C}_{\text{net}}$	101
$\mathcal{B}_{\text{net}}$	Lock 1

6-3 Uni-Directional Variable Routing case

This section will present the scheduler results for the UDVR case. The scheduler consist of the MILP model formulation as presented in 5-1, the SMPL equations as described in Section 4-3, which are transformed to MILP model constraints, as described in Section 5-3. This section will present the results when six vessels are sailing through the network.
As described in Section 6-1, first the inputs; the waterway network data and the vessel data, have to be defined. The waterway network graph results from the following topology matrix:

$$T_{\text{UDVR}} = \begin{bmatrix} 0&1&0&0&0&0&1&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \tag{6-8}$$

This produces the waterway network of Figure 6-7, where the blue arcs represent the waterways and the red arcs represent the lock area, with the solid line illustrating the waiting areas and the dotted line illustrating the lock itself. The arc weights (i.e. sailing timings $\tau_{w(i,j)}(k)$ and operation timings $\tau_{L(i,j)}$) and the nodes (i.e. states $x_i(k)$) are also shown in the figure. The practical definitions of the states follow the description presented in Section 4-1. To be complete, the definitions of all states are also given in Appendix B-2. It can be seen that the waterway network consists of two locks; lock 1 between node 3 and 4 and lock 2 between node 8 and 9 (also denoted with the squared markers). To repeat, overtaking within the locks is not possible. All vessels will start at $x_1(k)$ and will travel through the network to $x_{12}(k)$. As the scheduler decides if a vessel should take the left or the right route, and we have vessels only sailing downstream, we have the UDVR case. Next, we define the vessel specific waterway sailing timings $\tau_{w(i,j)}(k)$ and the lock specific levelling timings $\tau_{L(i,j)}$, which are shown in Table 6-15. Moreover, the vessel specific departure location $d(k)$, the departure time $u_i(k)$ and arrival deadline $a_i(k)$ are given as well.
It can be seen in Table 6-15 that we simulate a scenario in which the lock operation times $\tau_{L(i,j)}$ are equal for both locks. Moreover, the vessel sailing timings $\tau_{w(i,j)}(k)$ and the vessel priorities $\sigma(k)$ are equal for all six vessel as well. At last, the vessels depart 5 time unit after the preceding vessel, with vessel 1 departing at time unit 0.

Table 6-15: The vessel data and object data for the UDVR case in time units, which includes; sailing times $\tau_{w(i,j)}(k)$, lock operation times $\tau_{L(i,j)}$, departure location $d(k)$, departure time $u_i(k)$ and arrival deadlines $a_i(k)$ for 6 vessels sailing the waterway network

Operations	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
$\tau_{w(1,2)}(k)$	25	25	25	25	25	25
$\tau_{L(2,3)}$	2	2	2	2	2	2
$\tau_{L(3,4)}$	5	5	5	5	5	5
$\tau_{L(4,5)}$	2	2	2	2	2	2
$\tau_{w(5,6)}(k)$	25	25	25	25	25	25
$\tau_{w(6,12)}(k)$	25	25	25	25	25	25
$\tau_{w(1,7)}(k)$	25	25	25	25	25	25
$\tau_{L(7,8)}$	2	2	2	2	2	2
$\tau_{L(8,9)}$	5	5	5	5	5	5
$\tau_{L(9,10)}$	2	2	2	2	2	2
$\tau_{w(10,11)}(k)$	25	25	25	25	25	25
$\tau_{w(11,12)}(k)$	25	25	25	25	25	25
$d(k)$	1	1	1	1	1	1
$u_1(k)$	0	5	10	15	20	25
$a_{12}(k)$	90	100	110	120	130	140

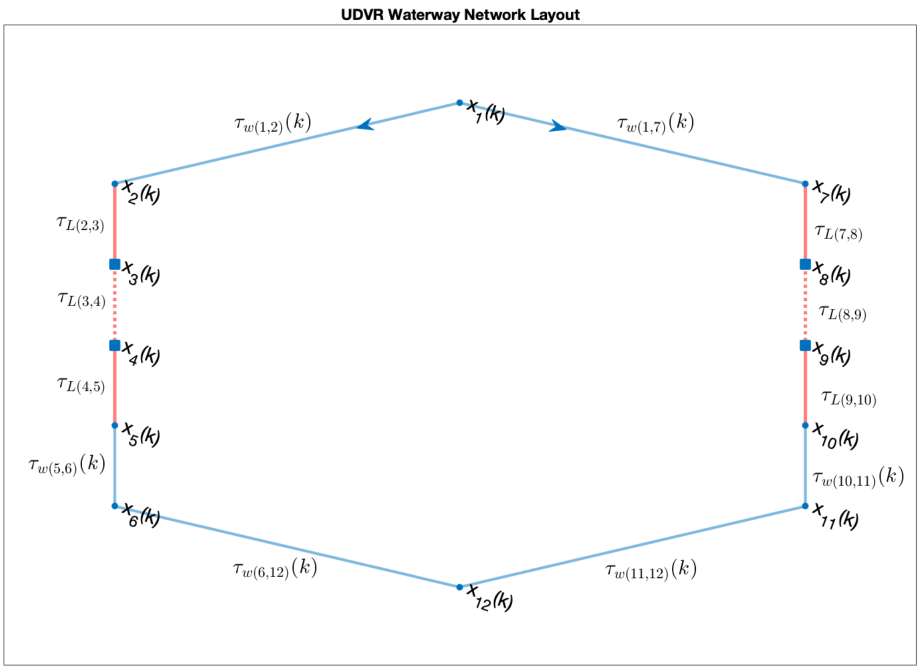


Figure 6-7: The waterway network layout for the UDVR case; with the blue arcs representing the waterways and the red arcs representing the lock area, where the solid line illustrates the waiting areas and the dotted line illustrates the lock itself

After having defined the waterway network input data and the vessel input data, the scheduler for the UDVR case was executed, and a feasible solution was found. We continue this section with the scheduler output. As described in Section 6-1-3, the scheduler output can be split into; the vessel schedule and properties, the object schedule and properties, and the global network properties. All three will be presented in the remainder of this section.

Firstly, this section will present the vessel schedule and vessel properties. The optimal schedule and subsequent state evolution of the six vessels can be seen in Table 6-16 and is visualised in Figure 6-8. Moreover, the destination and intermediate node arrival times are given. As can be seen the order of lock passing is for the first lock (node 3 and 4):

$$\text{Vessel 1} \rightarrow \text{Vessel 3} \rightarrow \text{Vessel 5}$$

For the second lock (node 8 and 9):

$$\text{Vessel 2} \rightarrow \text{Vessel 4} \rightarrow \text{Vessel 6}$$

As all vessels have equal sailing times $\tau_{w(i,j)}(k)$, lock operation times $\tau_{L(i,j)}$ and no vessel has a priority over other vessels, we can expect the vessels to arrive in their departure order. Moreover, this is an interesting case because all vessels depart 5 time units after the preceding vessel. As the operating time of the locks is also exactly 5 see, $\tau_{L(3,4)} = 5$ and $\tau_{L(8,9)} = 5$, the vessels should be able to sail the network without any delays when they would be scheduled alternately on left route 1 and right route 2. This is confirmed by the simulation. As shown, the odd-numbered vessels are scheduled on left route 1 and the even-numbered vessels are all scheduled on right route 2.

Table 6-16: The optimal IWT schedule for the UDVR case in time units, which includes; the route the vessels are scheduled on, the intermediate event timings $x_i(k)$ and the arrival time at $x_{12}(k)$ for 6 vessels sailing the waterway network

States	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Route	Left	Right	Left	Right	Left	Right
$x_1(k)$	0	5	10	15	20	25
$x_2(k)$	25	-	35	-	45	-
$x_3(k)$	27	-	37	-	47	-
$x_4(k)$	32	-	42	-	52	-
$x_5(k)$	34	-	44	-	54	-
$x_6(k)$	59	-	69	-	79	-
$x_7(k)$	-	30	-	40	-	50
$x_8(k)$	-	32	-	42	-	52
$x_9(k)$	-	37	-	47	-	57
$x_{10}(k)$	-	39	-	49	-	59
$x_{11}(k)$	-	64	-	74	-	84
$x_{12}(k)$	84	89	94	99	104	109

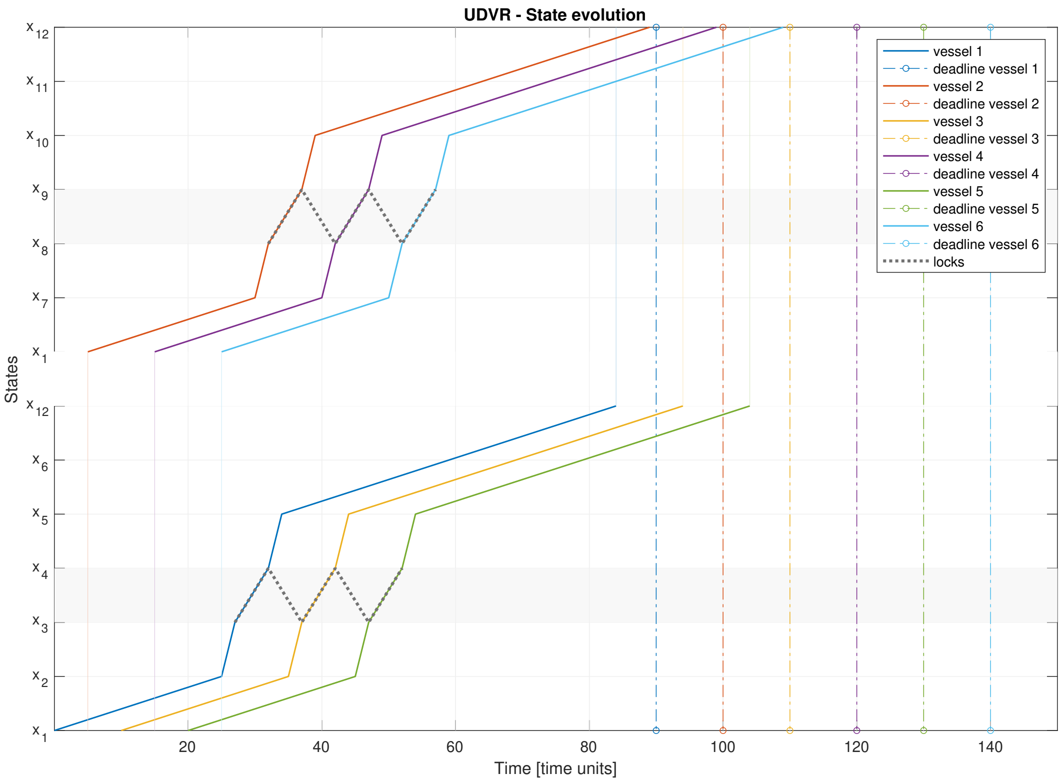


Figure 6-8: Visualisation of the optimal IWT schedule for the UDVR case shown in Table 6-16, with the grey shaded area representing the locks (note, the numbering on the state axis)

As mentioned, due to the late departures and the optimal alternation of vessels on left route 1 and right route 2, there are no delays in the schedule. However, for completeness, the delay table is shown in Table 6-22.

Table 6-17: The individual vessel delays per lock for the UDVR case in time units

Objects	Direction	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Lock 1	Downstream	0	-	0	-	0	-
Lock 2	Downstream	-	0	-	0	-	0

Secondly, this section will continue with presenting the object schedule and properties. The optimal schedule and subsequent state evolution of the lock can be seen in Table 6-18 and is visualised in Figure 6-9. We can see the number of lock levellings for the two locks, with; $\mathcal{L}_1 = 5$ and $\mathcal{L}_2 = 5$. Moreover, we can infer the number of empty lock levellings from Figure 6-14, with; $\mathcal{L}_{\text{e},1} = 2$ and $\mathcal{L}_{\text{e},2} = 2$. At last, we can see the occupancy for lock i at time unit t. For instance, $\mathcal{O}_{\text{lock},1}(40) = \text{vessel 3}$. As there are no delays in the schedule there are also no queues to determine, for example, $\mathcal{Q}_{\text{lock},1}(35) = \text{none}$.

Table 6-18: The optimal lock schedule for the UDFR case in time units, which includes for each lock the time when it should move upstream and when it should move downstream

Objects	Direction	Timings		
Lock 1	Upstream	27	37	47
	Downstream	32	42	52
Lock 2	Upstream	32	42	52
	Downstream	37	47	57

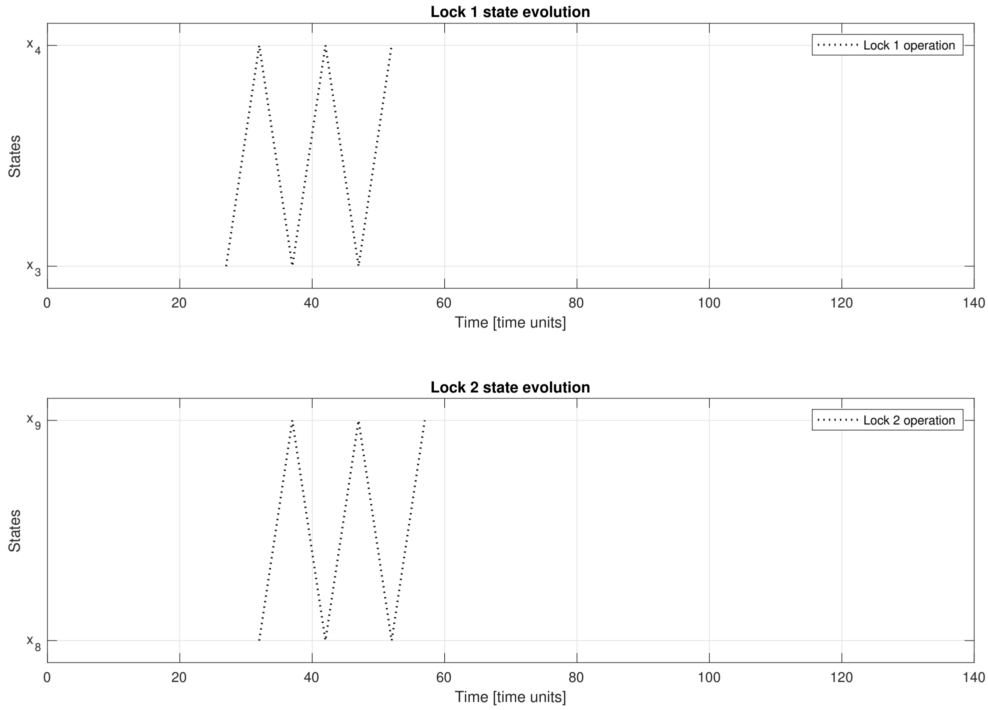


Figure 6-9: Visualisation of the optimal lock schedule for the UDVR case shown in Table 6-18

Thirdly, to conclude the results for the UDVR case, Table 6-19 shows the global network properties in time units as defined in Section 6-1. We have two routes with no bottlenecks, because no delays occur in the schedule.

Table 6-19: The global network properties in time units for the UDVR case

Variable	**Value** [time units]
$\mathcal{A}_{\text{net}}$	470
$\mathcal{D}_{\text{net}}$	0
$\mathcal{C}_{\text{net}}$	109
$\mathcal{B}_{\text{net}}$	none

6-4 Bi-Directional Fixed Routing case

This section will present the scheduler results for the BDFR case. The scheduler consist of the MILP model formulation as presented in 5-1, the SMPL equations as described in Section 4-4, which are transformed to MILP model constraints, as described in Section 5-4. This section will present the results when six vessels are sailing through the network.
As described in Section 6-1, first the inputs; the waterway network data and the vessel data, have to be defined. The waterway network graph results from the following topology matrix:

$$T_{\text{BDFR}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{6-9}$$

This produces the waterway network of Figure 6-10, where the blue arcs represent the waterways and the red arcs represent the lock area, with the solid line illustrating the waiting areas and the dotted line illustrating the lock itself. The arc weights (i.e. sailing timings $\tau_{w(i,j)}(k)$ and operation timings $\tau_{L(i,j)}$) and the nodes (i.e. states $x_i(k)$) are also shown in the figure. The practical definitions of the states follow the description presented in Section 4-1. To be complete, the definitions of all states are also given in Appendix B-3. It can be seen that the waterway network consists of one lock between node 3 and 4 (also denoted with the squared markers). To repeat, overtaking within the locks is not possible. All vessels will start either at $x_1(k)$ or at $x_6(k)$ and will travel through the network to either $x_6(k)$ or $x_1(k)$, respectively. As we have vessels sailing upstream and downstream and no routing decision has to be made by the scheduler, we have the BDFR case. Next, we define the vessel specific waterway sailing timings $\tau_{w(i,j)}(k)$ and the lock specific levelling timings $\tau_{L(i,j)}$, which are shown in Table 6-20. Moreover, the vessel specific departure location $d(k)$, the departure time $u_i(k)$ and arrival deadline $a_i(k)$ are given as well.
It can be seen in Table 6-20 that we simulate a scenario in which vessels 1, 3, and 5 sail downstream, and vessels 2, 4 and 6 sail upstream. Moreover, we can see that vessels 3 and 4 are faster than vessels 1, 2, 5 and 6, because the sailing times $\tau_{w(i,j)}(k)$ of vessel 3 and 4 are less. The vessel priorities $\sigma(k)$ are equal for all six vessel as well. At last, the vessels depart one time unit after the preceding vessel, with vessel 1 departing at time unit 1.

Table 6-20: The vessel data and object data for the BDFR case in time units, which includes; sailing times $\tau_{w(i,j)}(k)$, lock operation times $\tau_{L(i,j)}$, departure location $d(k)$, departure time $u_i(k)$ and arrival deadlines $a_i(k)$ for 6 vessels sailing the waterway network

Operations	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
$\tau_{w(1,2)}(k)$	25	25	15	15	25	25
$\tau_{L(2,3)}$	2	2	2	2	2	2
$\tau_{L(3,4)}$	5	5	5	5	5	5
$\tau_{L(4,5)}$	2	2	2	2	2	2
$\tau_{w(5,6)}(k)$	25	25	15	15	25	25
$d(k)$	1	6	1	6	1	6
$u_1(k)$	1	-	3	-	5	-
$u_6(k)$	-	2	-	4	-	6
$a_1(k)$	-	75	-	95	-	115
$a_6(k)$	65	-	85	-	105	-

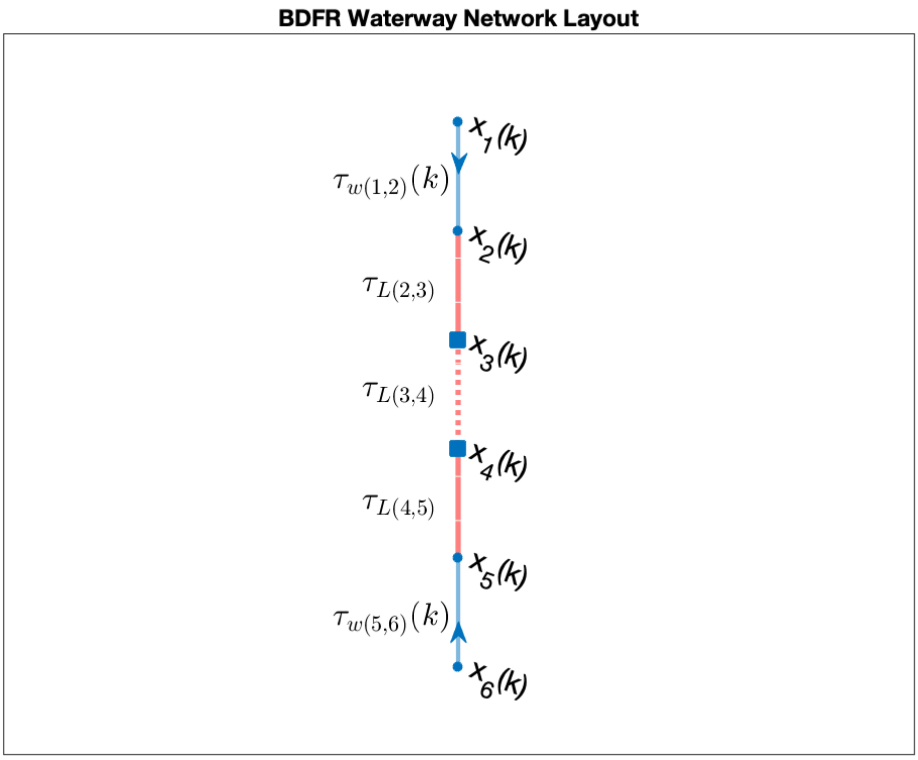


Figure 6-10: The waterway network layout for the BDFR case; with the blue arcs representing the waterways and the red arcs representing the lock area, where the solid line illustrates the waiting areas and the dotted line illustrates the lock itself

After having defined the waterway network input data and the vessel input data, the scheduler for the BDFR case was executed, and a feasible solution was found. We continue this section with the scheduler output. As described in Section 6-1-3, the scheduler output can be split into; the vessel schedule and properties, the object schedule and properties, and the global network properties. All three will be presented in the remainder of this section.

Firstly, this section will present the vessel schedule and vessel properties. The optimal schedule and subsequent state evolution of the six vessels can be seen in Table 6-21 and is visualised in Figure 6-11. Moreover, the destination and intermediate node arrival times are given. As can be seen the order of lock passing is for the lock (node 3 and 4):

$$\text{Vessel 3} \rightarrow \text{Vessel 4} \rightarrow \text{Vessel 1} \rightarrow \text{Vessel 2} \rightarrow \text{Vessel 5} \rightarrow \text{Vessel 6}$$

We would expect the faster vessels 3 and 4 to overtake the other vessels, be processed by the lock earlier and arrive at their destination the earliest. Moreover, as vessels 1, 2, 5, and 6 have equal sailing times $\tau_{w(i,j)}(k)$, lock operation times $\tau_{L(i,j)}$ and no vessel has a priority over other vessels, we can expect them to arrive in their departure order. Both statements are confirmed by the results.

Table 6-21: The optimal IWT schedule for the BDFR case in time units, which includes; the intermediate event timings $x_i(k)$ and the arrival time at $x_1(k)$ or $x_6(k)$ for 6 vessels sailing the waterway network

States	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
$x_1(k)$	1	70	3	48	5	82
$x_2(k)$	26	45	18	33	30	57
$x_3(k)$	32	43	20	31	44	55
$x_4(k)$	37	38	25	26	49	50
$x_5(k)$	39	27	27	19	51	31
$x_6(k)$	64	2	42	4	76	6

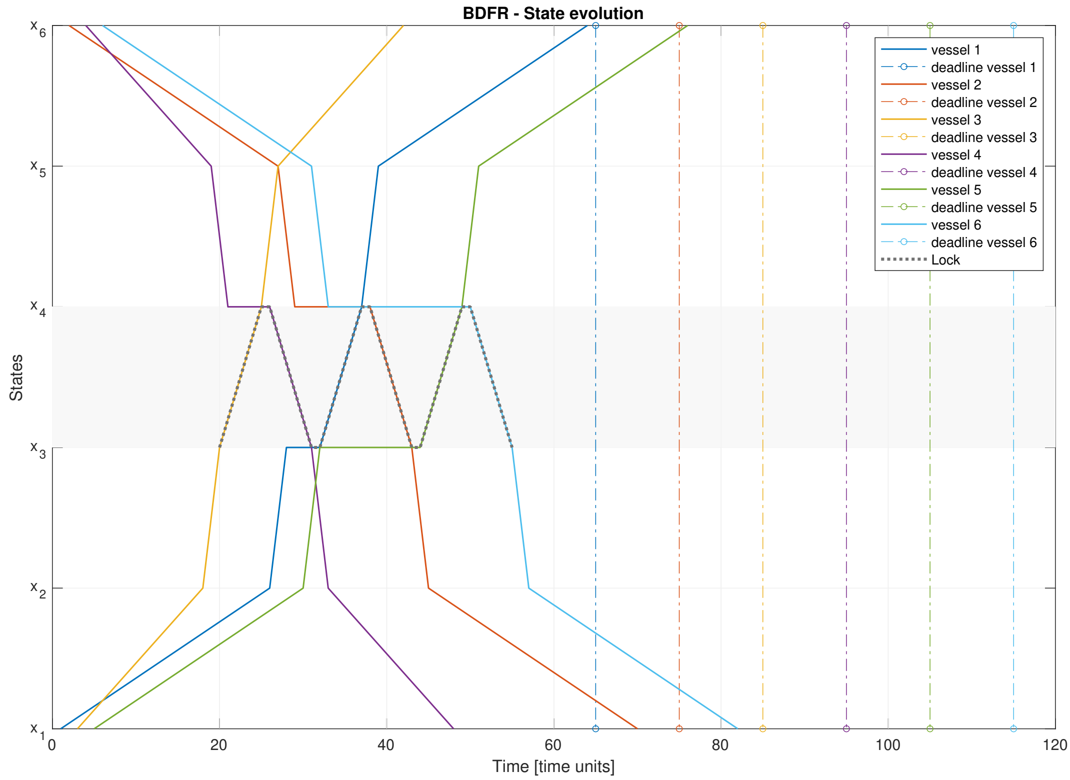


Figure 6-11: Visualisation of the optimal IWT schedule for the BDFR case shown in Table 6-21, with the grey shaded area representing the locks

Moreover, the optimal schedule results in delays for some vessels compared to when the vessel would be sailing the waterway network alone (without the influence of other vessels). The delays in time units for each vessel per lock can be seen in Table 6-22. These vessel delays are visualised in Appendix B-3 in Figure B-2.

Table 6-22: The individual vessel delays at the lock for the BDFR case in time units

Objects	Direction	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Lock 1	Downstream	4	-	0	-	12	-
	Upstream	-	9	-	5	-	17

Secondly, this section will continue with presenting the object schedule and object properties. The optimal schedule and subsequent state evolution of the lock can be seen in Table 6-23 and is visualised in Figure 6-12. We can see the number of lock levellings for the lock, with; $\mathcal{L}_1 = 6$. Moreover, we can infer the number of empty lock levellings from Figure 6-11, with; $\mathcal{L}_{e,1} = 0$, which makes sense as vessels can immediately be picked up from the other direction. At last, we can see the occupancy for lock i at time unit t. For instance, $\mathcal{O}_{\text{lock},1}(40) =$ vessel 2. The queue order and queue length for lock i at time unit t can also be derived, for example, $\mathcal{Q}_{\text{lock},1}(40) = 2^{\text{nd}}$ vessel 6 $\rightarrow 1^{\text{st}}$ vessel 5.

Table 6-23: The optimal lock schedule for the BDFR case in time units, which includes for the lock the time when it should move upstream and when it should move downstream

Objects	Direction	Timings		
Lock 1	Upstream	20	32	44
	Downstream	26	38	50

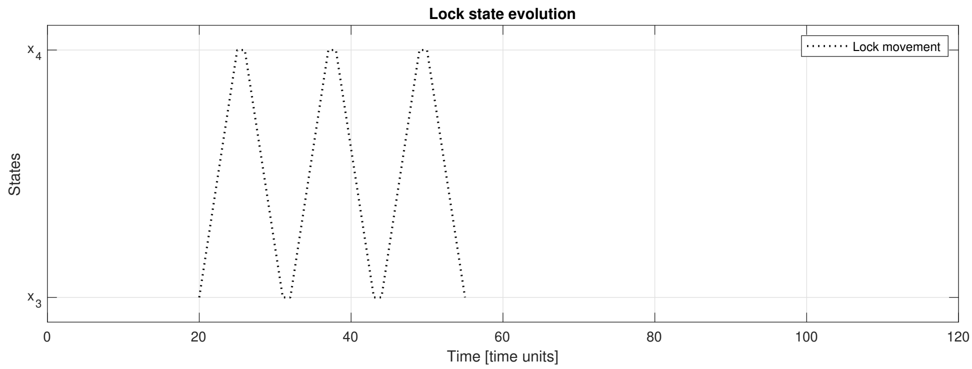


Figure 6-12: Visualisation of the optimal lock schedule for the BDFR case shown in Table 6-23

Thirdly, to conclude the results for the BDFR case, Table 6-24 shows the global network properties in time units as defined in Section 6-1. We have one route with one lock, therefore the bottleneck is rather obvious.

Table 6-24: The global network properties in time units for the BDFR case

Variable	**Value** [time units]
$\mathcal{A}_{\text{net}}$	382
$\mathcal{D}_{\text{net}}$	47
$\mathcal{C}_{\text{net}}$	82
$\mathcal{B}_{\text{net}}$	Lock 1

6-5 Bi-Directional Variable Routing case

This section will present the scheduler results for the BDVR case. The scheduler consist of the MILP model formulation as presented in 5-1, the SMPL equations as described in Section 4-5, which are transformed to MILP model constraints, as described in Section 5-5. This section will present the results when six vessels are sailing through the network.
As described in Section 6-1, first the inputs; the waterway network data and the vessel data, have to be defined. The waterway network graph results from the following topology matrix:

$$T_{\text{BIVR}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{6-10}$$

This produces the waterway network of Figure 6-13, where the blue arcs represent the waterways and the red arcs represent the lock area, with the solid line illustrating the waiting areas and the dotted line illustrating the lock itself. The arc weights (i.e. sailing timings $\tau_{w(i,j)}(k)$ and operation timings $\tau_{L(i,j)}$) and the nodes (i.e. states $x_i(k)$) are also shown in the figure. The practical definitions of the states follow the description presented in Section 4-1. To be complete, the definitions of all states are also given in Appendix B-4. It can be seen that the waterway network consists of two locks; lock 1 between node 3 and 4 and lock 2 between node 8 and 9 (also denoted with the squared markers). To repeat, overtaking within the locks is not possible. All vessels will start either at $x_1(k)$ or at $x_{12}(k)$ and will travel through the network to either $x_{12}(k)$ or $x_1(k)$, respectively. As the scheduler decides if a vessel has to take the left or the right route, and we have vessels sailing upstream and downstream, we have the BDVR case. Next, we define the vessel specific waterway sailing timings $\tau_{w(i,j)}(k)$ and the lock specific levelling timings $\tau_{L(i,j)}$, which are shown in Table 6-25. Moreover, the vessel specific departure location $d(k)$, the departure time $u_i(k)$ and arrival deadline $a_i(k)$ are given as well.
It can be seen in Table 6-25 that we simulate a scenario in which vessels 1, 3, and 5 sail downstream, and vessels 2, 4 and 6 sail upstream. The lock operation times $\tau_{L(i,j)}$ are equal for both locks. The vessel sailing timings $\tau_{w(i,j)}(k)$ and the vessel priorities $\sigma(k)$ are equal for all six vessel as well. At last, the vessels depart one time unit after the preceding vessel, with vessel 1 departing at time unit 1.

Table 6-25: The vessel data and object data for the BDVR case in time units, which includes; sailing times $\tau_{w(i,j)}(k)$, lock operation times $\tau_{L(i,j)}$, departure location $d(k)$, departure time $u_i(k)$ and arrival deadlines $a_i(k)$ for 6 vessels sailing the waterway network

Operations	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
$\tau_{w(1,2)}(k)$	25	25	25	25	25	25
$\tau_{L(2,3)}$	2	2	2	2	2	2
$\tau_{L(3,4)}$	5	5	5	5	5	5
$\tau_{L(4,5)}$	2	2	2	2	2	2
$\tau_{w(5,6)}(k)$	25	25	25	25	25	25
$\tau_{w(6,12)}(k)$	25	25	25	25	25	25
$\tau_{w(1,7)}(k)$	25	25	25	25	25	25
$\tau_{L(7,8)}$	2	2	2	2	2	2
$\tau_{L(8,9)}$	5	5	5	5	5	5
$\tau_{L(9,10)}$	2	2	2	2	2	2
$\tau_{w(10,11)}(k)$	25	25	25	25	25	25
$\tau_{w(11,12)}(k)$	25	25	25	25	25	25
$d(k)$	1	12	1	12	1	12
$u_1(k)$	1	-	3	-	5	-
$u_{12}(k)$	-	2	-	4	-	6
$a_1(k)$	-	100	-	120	-	140
$a_{12}(k)$	90	-	110	-	130	-

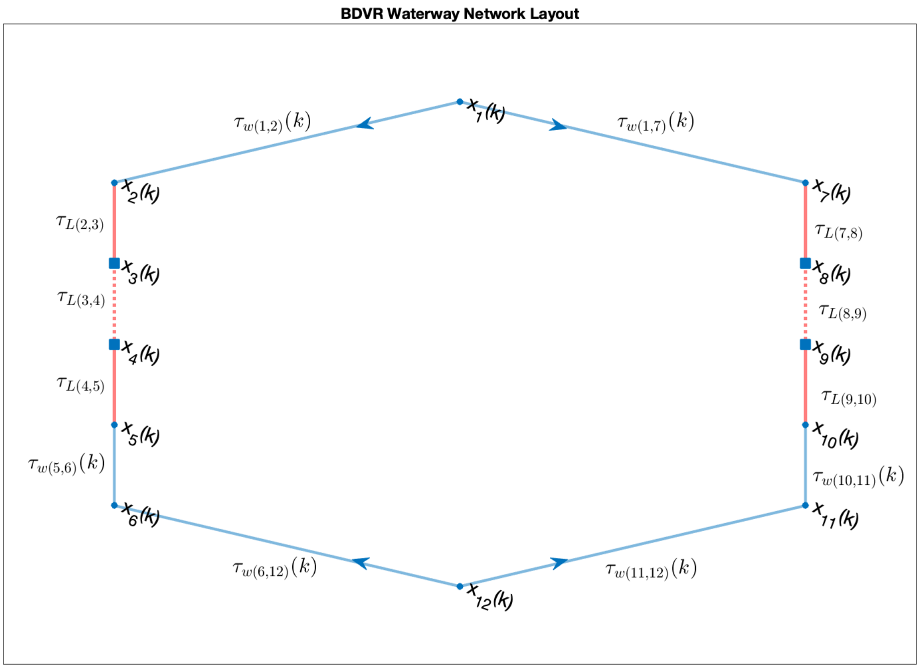


Figure 6-13: The waterway network layout for the BDVR case; with the blue arcs representing the waterways and the red arcs representing the lock area, where the solid line illustrates the waiting areas and the dotted line illustrates the lock itself

After having defined the waterway network input data and the vessel input data, the scheduler for the BDVR case was executed, and a feasible solution was found. We continue this section with the scheduler output. As described in Section 6-1-3, the scheduler output can be split into; the vessel schedule and properties, the object schedule and properties, and the global network properties. All three will be presented in the remainder of this section.

Firstly, this section will present the vessel schedule and vessel properties. The optimal schedule and subsequent state evolution of the six vessels can be seen in Table 6-26 and is visualised in Figure 6-14. Moreover, the destination and intermediate node arrival times are given. As can be seen the order of lock passing is for the first lock (node 3 and 4):

$$\text{Vessel 1} \rightarrow \text{Vessel 5} \rightarrow \text{Vessel 4}$$

For the second lock (node 8 and 9):

$$\text{Vessel 3} \rightarrow \text{Vessel 2} \rightarrow \text{Vessel 6}$$

As all vessels have equal sailing times $\tau_{w(i,j)}(k)$, lock operation times $\tau_{L(i,j)}$ and no vessel has a priority over other vessels, we can expect vessels which are scheduled on the same route and sail the same direction (i.e. upstream or downstream) to arrive in their departure order. This is confirmed by the results.

Table 6-26: The optimal IWT schedule for the BDVR case in time units, which includes; the route the vessels are scheduled on, the intermediate event timings $x_i(k)$ and the arrival time at $x_1(k)$ or $x_{12}(k)$ for 6 vessels sailing the waterway network

States	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Route	Left	Right	Right	Left	Left	Right
$x_1(k)$	1	86	3	88	5	96
$x_2(k)$	26	-	-	63	30	-
$x_3(k)$	28	-	-	61	38	-
$x_4(k)$	33	-	-	56	43	-
$x_5(k)$	35	-	-	54	45	-
$x_6(k)$	60	-	-	29	70	-
$x_7(k)$	-	61	28	-	-	71
$x_8(k)$	-	59	30	-	-	69
$x_9(k)$	-	54	35	-	-	64
$x_{10}(k)$	-	52	37	-	-	56
$x_{11}(k)$	-	27	62	-	-	31
$x_{12}(k)$	85	2	87	4	95	6

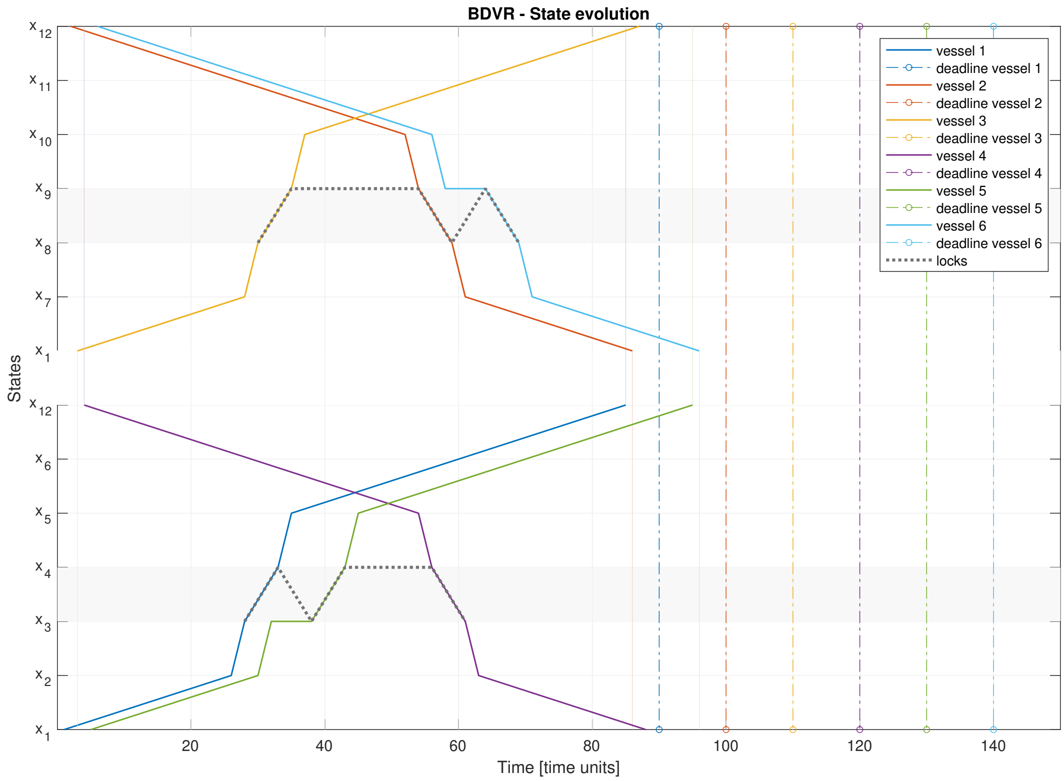


Figure 6-14: Visualisation of the optimal IWT schedule for the BDVR case shown in Table 6-26, with the grey shaded area representing the locks (note, the numbering on the state axis)

Moreover, the optimal schedule results in delays for some vessels compared to when the vessel would be sailing the waterway network alone (without the influence of other vessels). The delays in time units for each vessel per lock can be seen in Table 6-27. These vessel delays are visualised in Appendix B-4 in Figure B-3.

Table 6-27: The individual vessel delays per lock for the BDVR case in time units

Objects	Direction	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Lock 1	Downstream	0	-	-	0	6	-
	Upstream	-	-	-	-	-	-
Lock 2	Downstream	-	0	0	-	-	-
	Upstream	-	-	-	-	-	6

Secondly, this section will continue with presenting the object schedule and properties. The optimal schedule and subsequent state evolution of the lock can be seen in Table 6-28 and is visualised in Figure 6-15. We can see the number of lock levellings for the two locks, with; $\mathcal{L}_1 = 4$ and $\mathcal{L}_2 = 4$. Moreover, we can infer the number of empty lock levellings from Figure 6-14, with; $\mathcal{L}_{e,1} = 1$ and $\mathcal{L}_{e,2} = 1$. At last, we can see the occupancy for lock i at time unit t. For instance, $\mathcal{O}_{\text{lock},1}(40) =$ vessel 5. The queue order and queue length for lock i at time unit t can also be derived, for example, $\mathcal{Q}_{\text{lock},1}(35) = 1^{\text{st}}$ vessel 5.

Table 6-28: The optimal lock schedule for the BDVR case in time units, which includes for each lock the time when it should move upstream and when it should move downstream

Objects	Direction	Timings		
Lock 1	Upstream	28	38	56
	Downstream	33	43	-
Lock 2	Upstream	30	54	64
	Downstream	35	59	-

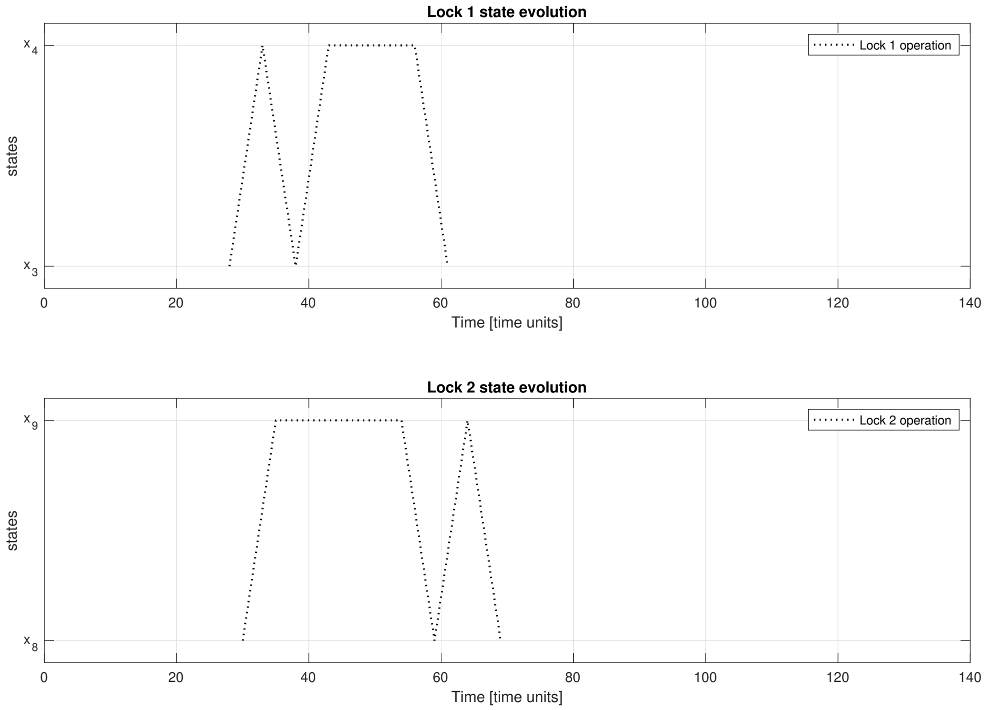


Figure 6-15: Visualisation of the optimal lock schedule for the BDVR case shown in Table 6-28

Thirdly, to conclude the results for the BDVR case, Table 6-29 shows the global network properties in time units as defined in Section 6-1. We have two routes, with the same number of vessels sailing the waterways with equal sailing times. Therefore, we would expect to see the same amount of delays on both routes and no obvious bottleneck in the IWT system. As can be seen, we have a delay of 6 time units on both routes, and therefore, the results confirm this.

Table 6-29: The global network properties in time units for the BDVR case

Variable	**Value** [time units]
$\mathcal{A}_{\text{net}}$	537
$\mathcal{D}_{\text{net}}$	12
$\mathcal{C}_{\text{net}}$	96
$\mathcal{B}_{\text{net}}$	none

6-6 Bi-Directional Variable Routing case with locks in series

In order to show that we can generate different waterway networks using these four basic cases, an additional case was created by stacking the BDFR case on top of the BDVR case, which results in a waterway network with two locks in a row. This is done to show that every conceivable waterway network can be scheduled. The scheduler consist of the MILP model formulation as presented in 5-1, the SMPL equations as described in Section 4-4 and Section 4-5, which are transformed to MILP model constraints, as described in Section 5-4 and Section 5-5. This section will present the results when six vessels are sailing through the network.

As described in Section 6-1, first the inputs; the waterway network data and the vessel data, have to be defined. The waterway network graph results from the following topology matrix:

$$T_{\text{BIFRC}} = \begin{bmatrix} 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0 \end{bmatrix} \tag{6-11}$$

This produces the waterway network of Figure 6-16, where the blue arcs represent the waterways and the red arcs represent the lock area, with the solid line illustrating the waiting areas and the dotted line illustrating the lock itself. The arc weights (i.e. sailing timings $\tau_{w(i,j)}(k)$ and operation timings $\tau_{L(i,j)}$) and the nodes (i.e. states $x_i(k)$) are also shown in the figure. The practical definitions of the states follow the description presented in Section 4-1. To be complete, the definitions of all states are also given in Appendix B-5. It can be seen that the waterway network consists of three locks; lock 1 between node 3 and 4, lock 2 between node 7 and 8, and lock 3 between node 12 and 13 (also denoted with the squared markers). To repeat, overtaking within the locks is not possible. All vessels will start either at $x_1(k)$ or at $x_{16}(k)$ and will travel through the network to either $x_{16}(k)$ or $x_1(k)$, respectively. As the scheduler decides if a vessel has to take the left or the right route, and we have vessels sailing upstream and downstream, and we have two locks in a row, we have the BDVR case with locks in series. Next, we define the vessel specific waterway sailing timings $\tau_{w(i,j)}(k)$ and the lock specific levelling timings $\tau_{L(i,j)}$, which are shown in Table 6-30. Moreover, the vessel specific departure location $d(k)$, the departure time $u_i(k)$ and arrival deadline $a_i(k)$ are given as well.

It can be seen in Table 6-30 that we simulate a scenario in which vessels 1, 2, and 5 sail downstream, and vessels 3, 4 and 6 sail upstream. The lock operation times $\tau_{L(i,j)}$ are equal for all three locks. The vessel sailing timings $\tau_{w(i,j)}(k)$ and the vessel priorities $\sigma(k)$ are equal for all six vessel as well. At last, the vessels depart one time unit after the preceding vessel, with vessel 1 departing at time unit 1.

Table 6-30: The vessel data and object data for the BDVR case with locks in series in time units, which includes; sailing times $\tau_{w(i,j)}(k)$, lock operation times $\tau_{L(i,j)}$, departure location $d(k)$, departure time $u_i(k)$ and arrival deadlines $a_i(k)$ for 6 vessels sailing the waterway network

Operations	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
$\tau_{w(1,2)}(k)$	25	25	25	25	25	25
$\tau_{L(2,3)}$	2	2	2	2	2	2
$\tau_{L(3,4)}$	5	5	5	5	5	5
$\tau_{L(4,5)}$	2	2	2	2	2	2
$\tau_{w(5,6)}(k)$	25	25	25	25	25	25
$\tau_{L(6,7)}$	2	2	2	2	2	2
$\tau_{L(7,8)}$	5	5	5	5	5	5
$\tau_{L(8,9)}$	2	2	2	2	2	2
$\tau_{w(9,10)}(k)$	25	25	25	25	25	25
$\tau_{w(10,16)}(k)$	25	25	25	25	25	25
$\tau_{w(5,11)}(k)$	25	25	25	25	25	25
$\tau_{L(11,12)}$	2	2	2	2	2	2
$\tau_{L(12,13)}$	5	5	5	5	5	5
$\tau_{L(13,14)}$	2	2	2	2	2	2
$\tau_{w(14,15)}(k)$	25	25	25	25	25	25
$\tau_{w(15,16)}(k)$	25	25	25	25	25	25
$d(k)$	1	1	16	16	1	16
$u_1(k)$	1	2	-	-	5	-
$u_{16}(k)$	-	-	3	4	-	6
$a_1(k)$	-	-	140	150	-	170
$a_{16}(k)$	120	130	-	-	160	-

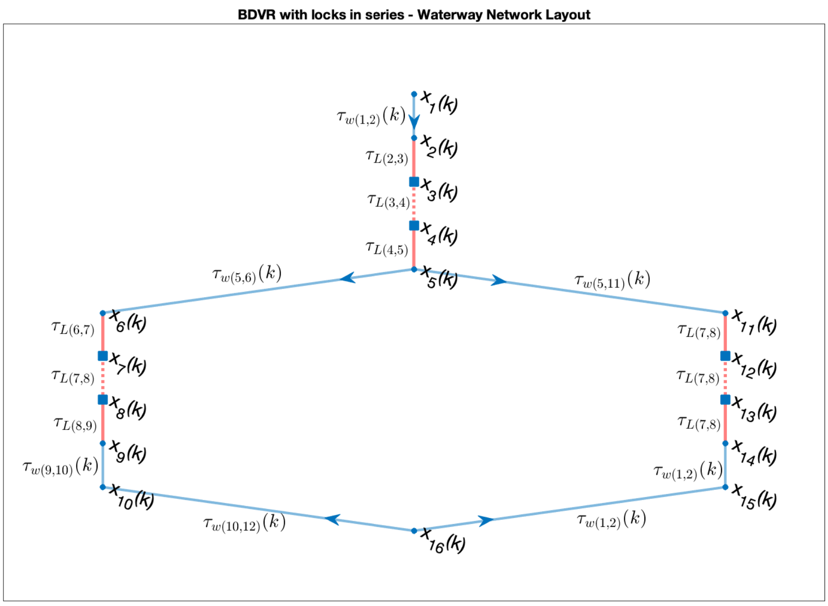


Figure 6-16: The waterway network layout for the BDVR case with locks in series; with the blue arcs representing the waterways and the red arcs representing the lock area, where the solid line illustrates the waiting areas and the dotted line illustrates the lock itself

After having defined the waterway network input data and the vessel input data, the scheduler for the BDVR case with lock in series was executed, and a feasible solution was found. We continue this section with the scheduler output. As described in Section 6-1-3, the scheduler output can be split into; the vessel schedule and properties, the object schedule and properties, and the global network properties. All three will be presented in the remainder of this section.

Firstly, this section will present the vessel schedule and vessel properties. The optimal schedule and subsequent state evolution of the six vessels can be seen in Table 6-31 and is visualised in Figure 6-17. Moreover, the destination and intermediate node arrival times are given. As can be seen the order of lock passing is for the first lock (node 3 and 4):

$$\text{Vessel 1} \rightarrow \text{Vessel 2} \rightarrow \text{Vessel 5} \rightarrow \text{Vessel 3} \rightarrow \text{Vessel 4} \rightarrow \text{Vessel 6}$$

For the second lock (node 7 and 8):

$$\text{Vessel 4} \rightarrow \text{Vessel 1} \rightarrow \text{Vessel 5}$$

For the third lock (node 12 and 13):

$$\text{Vessel 3} \rightarrow \text{Vessel 6} \rightarrow \text{Vessel 2}$$

As all vessels have equal sailing times $\tau_{w(i,j)}(k)$, lock operation times $\tau_{L(i,j)}$ and no vessel has a priority over other vessels, we can expect vessels which are scheduled on the same route and sail the same direction (i.e. upstream or downstream) to arrive in their departure order. This is confirmed by the results.

Table 6-31: The optimal IWT schedule for the BDVR case with locks in series in time units, which includes; the route the vessels are scheduled on, the intermediate event timings $x_i(k)$ and the arrival time at $x_1(k)$ or $x_{16}(k)$ for 6 vessels sailing the waterway network

States	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Route	Left	Right	Right	Left	Left	Right
$x_1(k)$	1	2	121	131	5	141
$x_2(k)$	26	27	96	106	30	116
$x_3(k)$	28	38	94	104	48	114
$x_4(k)$	33	43	89	99	53	109
$x_5(k)$	35	45	87	88	55	97
$x_6(k)$	60	-	-	63	80	-
$x_7(k)$	62	-	-	61	82	-
$x_8(k)$	67	-	-	56	87	-
$x_9(k)$	69	-	-	54	89	-
$x_{10}(k)$	94	-	-	29	114	-
$x_{11}(k)$	-	70	62	-	-	72
$x_{12}(k)$	-	72	60	-	-	70
$x_{13}(k)$	-	77	55	-	-	65
$x_{14}(k)$	-	79	53	-	-	56
$x_{15}(k)$	-	104	28	-	-	31
$x_{16}(k)$	119	129	3	4	139	6

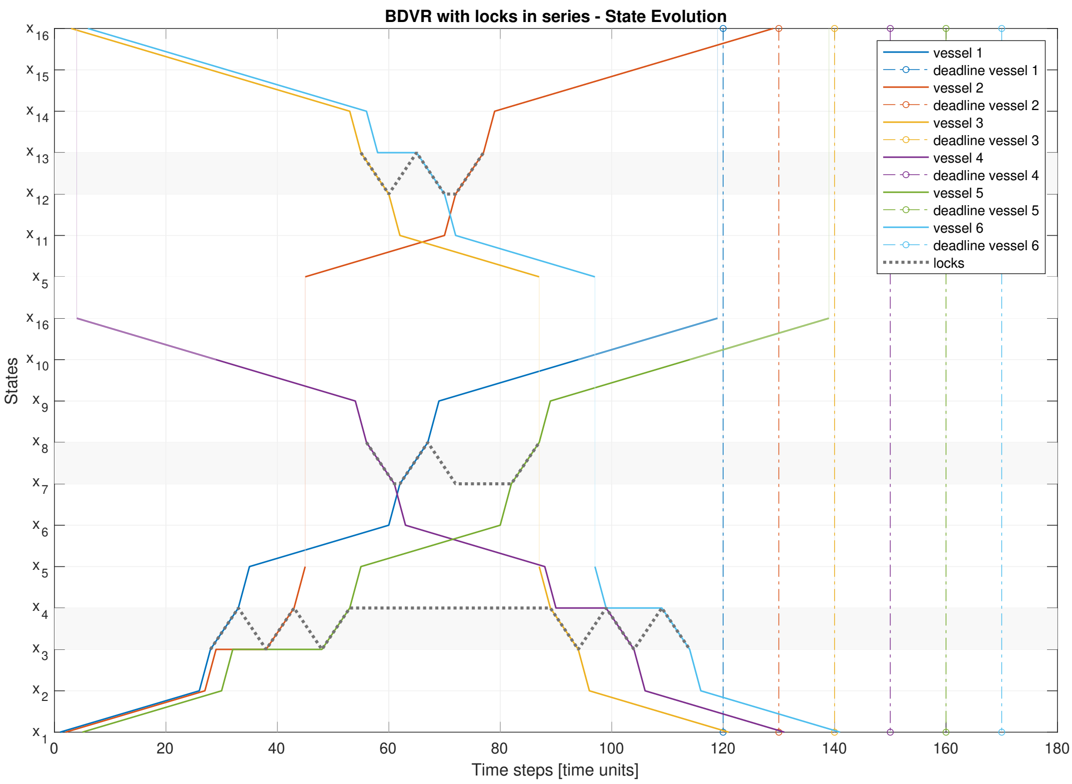


Figure 6-17: Visualisation of the optimal IWT schedule for the BDVR case with locks in series shown in Table 6-31, with the grey shaded area representing the locks (note, the numbering on the state axis)

Moreover, the optimal schedule results in delays for some vessels compared to when the vessel would be sailing the waterway network alone (without the influence of other vessels). The delays in time units for each vessel per lock can be seen in Table 6-32. These vessel delays are visualised in Appendix B-5 in Figure B-4.

Table 6-32: The individual vessel delays per lock for the BDVR case with locks in series in time units

Objects	Direction	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Lock 1	Downstream	0	9	-	-	16	-
	Upstream	-	-	0	9	-	7
Lock 2	Downstream	0	-	-	-	0	-
	Upstream	-	-	-	0	-	-
Lock 3	Downstream	-	0	-	-	-	-
	Upstream	-	-	0	-	-	10

Secondly, this section will continue with presenting the object schedule and properties. The optimal schedule and subsequent state evolution of the lock can be seen in Table 6-33 and is visualised in Figure 6-18. We can see the number of lock levellings for the 3 locks, with; $\mathcal{L}_1 = 10$, $\mathcal{L}_2 = 4$, and $\mathcal{L}_3 = 4$. Moreover, we can infer the number of empty lock levellings from Figure 6-17, with; $\mathcal{L}_{e,1} = 4$, $\mathcal{L}_{e,2} = 1$, and $\mathcal{L}_{e,3} = 1$. At last, we can see the occupancy for lock i at time unit t. For instance, $\mathcal{O}_{\text{lock},1}(50) =$ vessel 1. The queue order and queue length for lock i at time unit t can also be derived, for example, $\mathcal{Q}_{\text{lock},1}(35) = 2^{\text{nd}}$ vessel $5 \rightarrow 1^{\text{st}}$ vessel 2.

Table 6-33: The optimal lock schedule for the BDVR case with locks in series, which includes for each lock the time when it should move upstream and when it should move downstream in time units

Objects	Direction	Timings				
Lock 1	Upstream	28	38	48	94	104
	Downstream	33	43	89	99	109
Lock 2	Upstream	-	62	82		
	Downstream	56	67	-		
Lock 3	Upstream	-	60	72		
	Downstream	55	65	-		

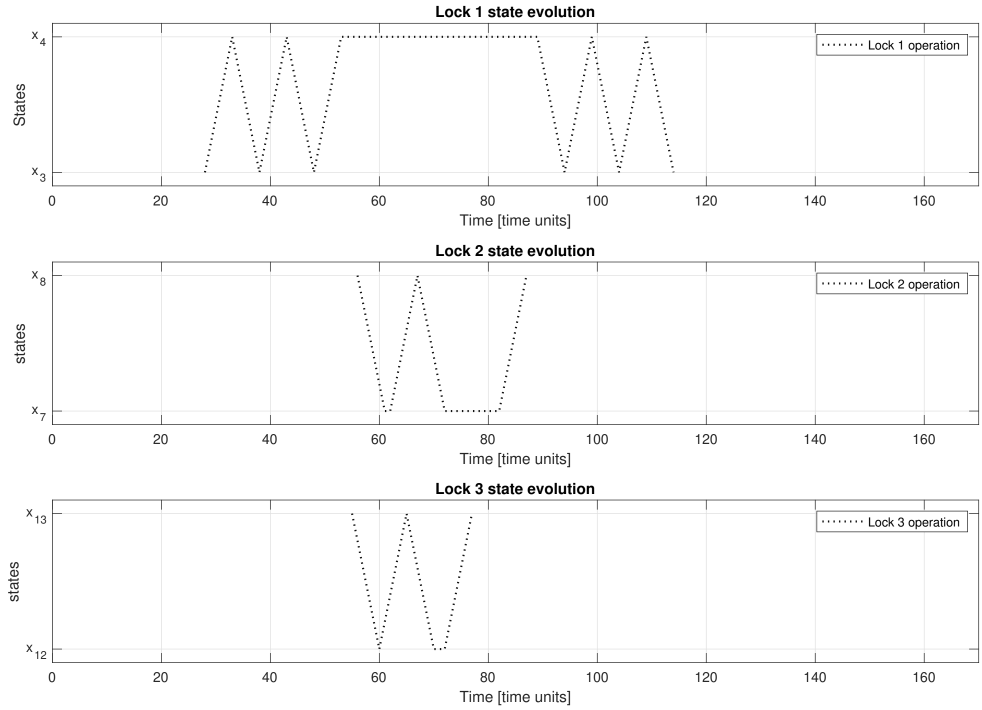


Figure 6-18: Visualisation of the optimal lock schedule for the BDVR case with locks in series shown in Table 6-33

Thirdly, to conclude the results for the BDVR case with locks in series, Table 6-34 shows the global network properties in time units as defined in Section 6-1. This is actually the first case where the bottleneck $\mathcal{B}_{\text{net}}$ becomes clear and valuable. As indicated in Table 6-34, the bottleneck of the IWT system is at lock 1. This is also expected and quite natural as lock 1 is present on two different routes; thus, vessels from multiple routes interfere.

Table 6-34: The global network properties in time units for the BDVR case with locks in series

Variable	**Value** [time units]
$\mathcal{A}_{\text{net}}$	780
$\mathcal{D}_{\text{net}}$	53
$\mathcal{C}_{\text{net}}$	141
$\mathcal{B}_{\text{net}}$	Lock 1

6-7 Additional scenarios

In order to prove that the scheduling strategy works for other vessel input data as well, three additional example scenarios of the BDVR case with locks in series are shown. We will only show the input data table, the output data, and the output schedule figure for these scenarios. The output data tables will be shown in Appendix C, such that the results can be verified. The scenarios will not be described as extensively as was done for the cases earlier in this chapter. This is to try to keep it as compact as possible. Scenario A will consider three types of vessels with various sailing times. Scenario B will build upon the data of scenario A, but the objective function will be modified. At last, scenario C will build upon scenario B, with an added strict arrival deadline.

6-7-1 Scenario A: BDVR with locks in series with 3 vessel types

For scenario A, we consider three vessel types; slow, normal and fast vessels. Vessel 1 and 2 are slow vessels and have a sailing time of $\tau_{w(i,j)}(k) = 25$. Vessel 3 and 4 are normal vessels and have a sailing time of $\tau_{w(i,j)}(k) = 20$. Vessel 5 and 6 are fast vessels and have a sailing time of $\tau_{w(i,j)}(k) = 15$. This vessel data can be seen in Table 6-35, including lock operation times $\tau_{L(i,j)}$, departure location $d(k)$, departure time $u_i(k)$ and arrival deadlines $a_i(k)$ for the six vessels.

Table 6-35: The vessel data and object data for the **scenario A** BDVR case with locks in series in time units, which includes; sailing times $\tau_{w(i,j)}(k)$, lock operation times $\tau_{L(i,j)}$, departure location $d(k)$, departure time $u_i(k)$ and arrival deadlines $a_i(k)$ for 6 vessels sailing the waterway network

Operations	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
$\tau_{w(1,2)}(k)$	25	**20**	25	**20**	**15**	**15**
$\tau_{L(2,3)}$	2	2	2	2	2	2
$\tau_{L(3,4)}$	5	5	5	5	5	5
$\tau_{L(4,5)}$	2	2	2	2	2	2
$\tau_{w(5,6)}(k)$	25	**20**	25	**20**	**15**	**15**
$\tau_{L(6,7)}$	2	2	2	2	2	2
$\tau_{L(7,8)}$	5	5	5	5	5	5
$\tau_{L(8,9)}$	2	2	2	2	2	2
$\tau_{w(9,10)}(k)$	25	**20**	25	**20**	**15**	**15**
$\tau_{w(10,16)}(k)$	25	**20**	25	**20**	**15**	**15**
$\tau_{w(5,11)}(k)$	25	**20**	25	20	**15**	**15**
$\tau_{L(11,12)}$	2	2	2	2	2	2
$\tau_{L(12,13)}$	5	5	5	5	5	5
$\tau_{L(13,14)}$	2	2	2	2	2	2
$\tau_{w(14,15)}(k)$	25	**20**	25	20	**15**	**15**
$\tau_{w(15,16)}(k)$	25	**20**	25	20	**15**	**15**
$d(k)$	1	1	16	16	1	16
$u_1(k)$	1	2	-	-	5	-
$u_{16}(k)$	-	-	3	4	-	6
$a_1(k)$	-	-	140	150	-	170
$a_{16}(k)$	120	130	-	-	160	-

For scenario A, the optimal schedule and subsequent state evolution of the six vessels can be seen in Table C-1 and is visualised in Figure 6-19. We can see that the faster vessels 2 and 5 have to wait at lock 1, even though they arrive earlier then vessel 1. This ordering ensures vessel 1 reaches its arrival deadline. If the scheduling strategy was based on the First-come, First-serve principle, vessel 1 would have missed its arrival deadline. Thus we can verify from the simulations that the scheduling strategy works. Moreover, note how vessel 2 and 5 are scheduled on different routes after coming out of lock 1, this is convenient so they do not interfere.

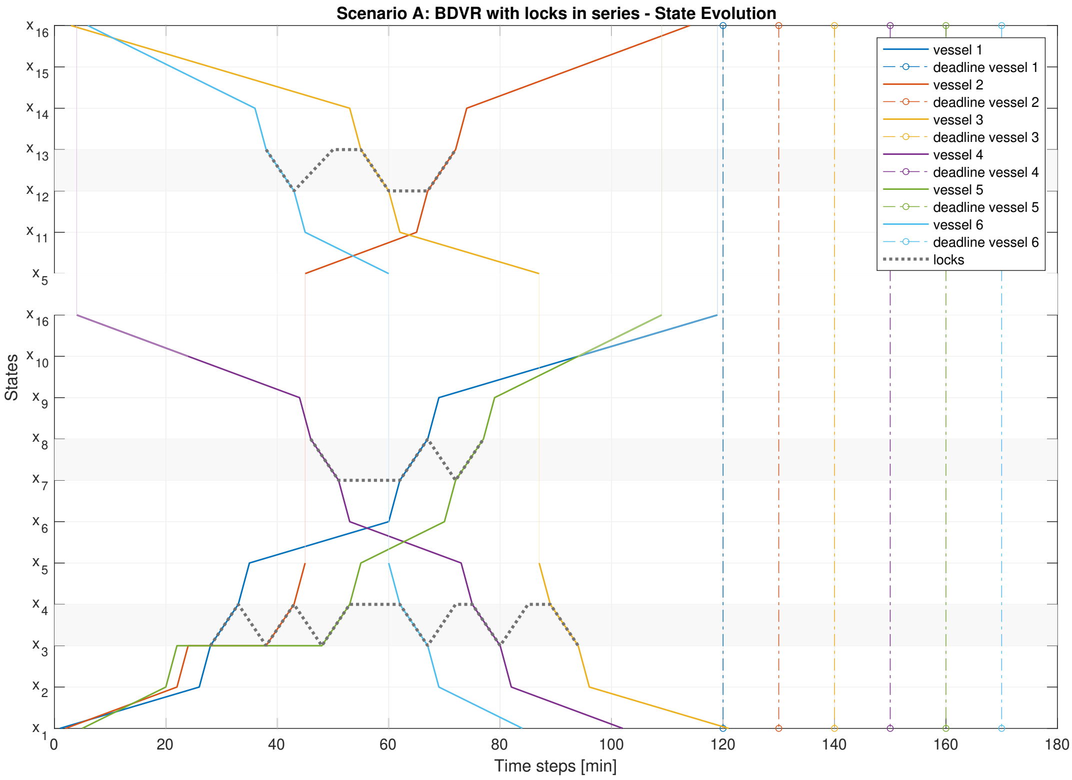


Figure 6-19: Visualisation of the optimal IWT schedule for the **scenario A** BDVR case with locks in series shown in Table C-1, with the grey shaded area representing the locks (note, the numbering on the state axis)

6-7-2 Scenario B: BDVR with locks in series with vessel priority

For scenario B, we consider the same data as presented in Table 6-35. However, in this case vessel 5 has a special type of cargo which makes it arriving earlier convenient, but not mandatory. To implement this we use the weight function $\sigma(k)$ of the output objective function $J_{\text{out}}(k)$, as defined in Section 6-1-1, with $\sigma_a(5) > \{\sigma_a(1), \sigma_a(2), \sigma_a(3), \sigma_a(4), \sigma_a(6)\}$. For scenario B, the optimal schedule and subsequent state evolution of the six vessels can be seen in Table C-2 and is visualised in Figure 6-20. As expected, we can see that vessel 5 is allowed to go through lock 1 before vessel 2. Moreover, note how vessel 1 is scheduled on the right route 2 instead of the left route 1 as in the previous example of Scenario B. This is to make room on route 1 for vessel 5.

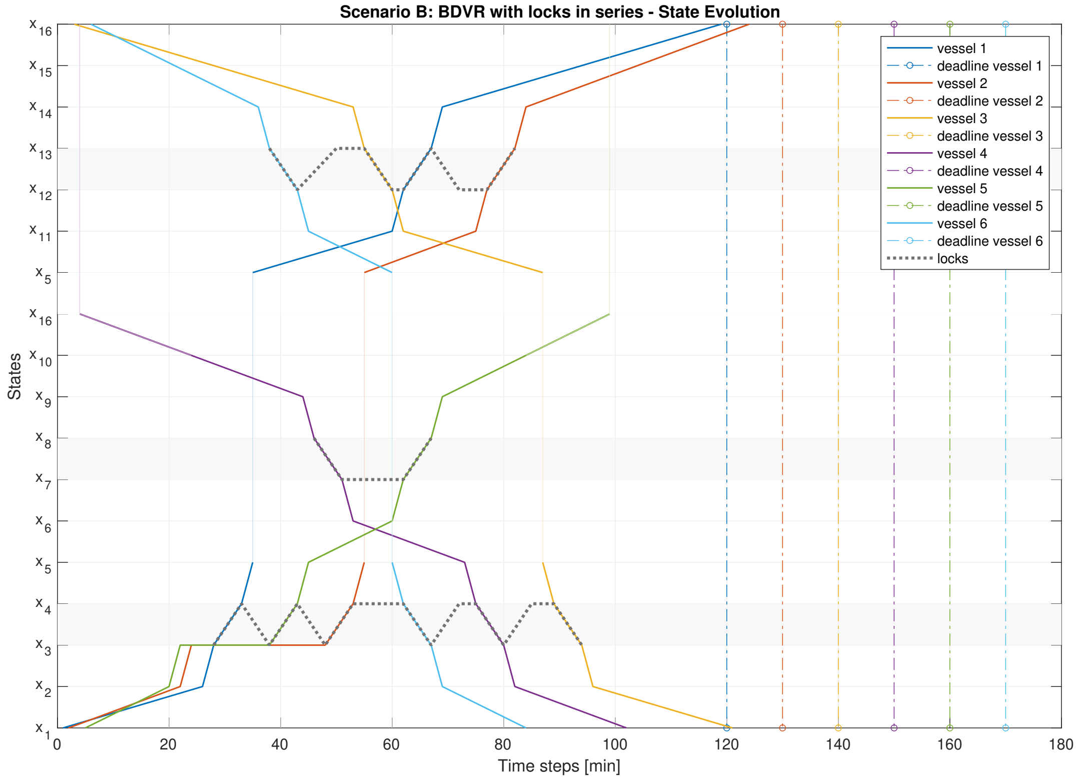


Figure 6-20: Visualisation of the optimal IWT schedule for the **scenario B** BDVR case with locks in series shown in Table C-2, with the grey shaded area representing the locks (note, the numbering on the state axis)

6-7-3 Scenario C: BDVR with locks in series with strict vessel deadline

For scenario C, we consider the same data as presented in Table 6-35, with only changing the arrival deadline for vessel 1 and vessel 2. The arrival deadline constraint on vessel 1 will be relaxed and for vessel 2 will be more strict, this can be seen in Table 6-36. The the weight function $\sigma_a(5) > \{\sigma_a(1), \sigma_a(2), \sigma_a(3), \sigma_a(4), \sigma_a(6)\}$ of scenario B.

Table 6-36: The vessel arrival deadlines $a_i(k)$ for 6 vessels sailing the waterway network of the **scenario C** BDVR case with locks in series in time units

Operations	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
$a_1(k)$	-	-	140	150	-	170
$a_{16}(k)$	**145**	**100**	-	-	160	-

For scenario C, the optimal schedule and subsequent state evolution of the six vessels can be seen in Table C-3 and is visualised in Figure 6-21. As expected, vessel 2 is allowed to pass lock 1 first, after which vessel 5 follows, and last is vessel 1. With this, it can be confirmed that the arrival time deadline outweighs the vessel priority.

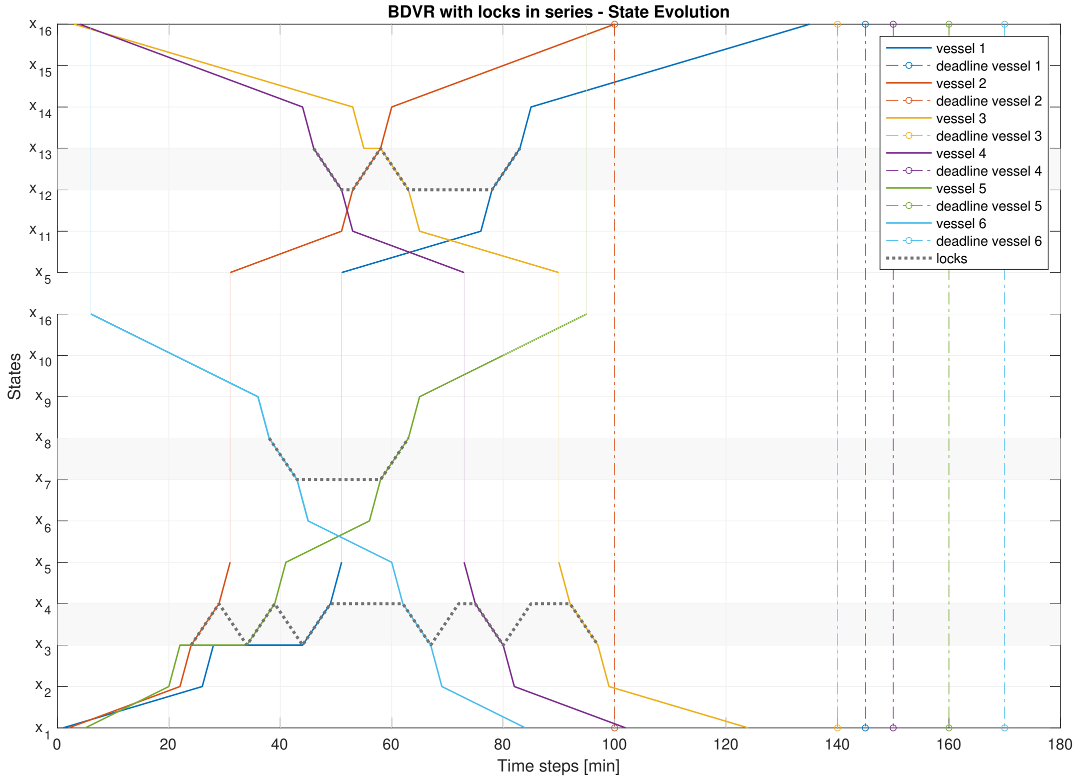


Figure 6-21: Visualisation of the optimal IWT schedule for the **scenario C** BDVR case with locks in series shown in Table C-3, with the grey shaded area representing the locks (note, the numbering on the state axis)

6-7-4 Scenario D: BDVR with locks in series with more vessels

At last, for scenario D, it will be shown that simulations can be created for an arbitrary number of vessels. This is done to prove that the scheduler automatically generates the routing constraints and ordering constraints for any number of vessels, as stated in Section 6-1. In Table 6-38 the schedule is shown for ten vessels sailing alternately downstream and upstream, all with the sailing time of a slow vessel as defined in scenario A in Section 6-7-1. In Figure C-1 in Appendix C-4, this is shown for 12 vessels.

In the previous sections, we always worked with six vessels because this is computationally convenient, the simulations can be performed quickly, and the visualisations of the schedules remain clear. Moreover, it is sufficient to prove the working principles of the scheduling strategy. However, increasing the number of vessels has a significant impact on the running time of the scheduler, which must be mentioned. This is because having a large $\mu \in \{\mu_{\min}, \ldots, \mu_{\max}\}$ range is computationally demanding, as mentioned in Section 3-5-4 and Section 3-6-2. Table 6-37, shows the number of vessels with the number of required Max-Plus binary control variables and the running time of the scheduler. For ordering n vessels, we require $n(n-1)/2$ Max-Plus binary control variables $w_\mu(k-\mu)$. This results in $2^{n(n-1)/2}$ possible combinations of control actions. While the number of possible permutations of n vessels is factorial, $n!$.

This means, that for $n \leq 3$ there are more control actions than possible permutations of n vessels. This is not desirable but solving this is not within the scope of the research. It is, however, interesting for future research, and several studies have already been performed in this direction [26, 28].
For now, $\mu_{\max}$ should be chosen well from a practical point of view. For example, it could be stated that it is only realistic that vessels within a range of 10 preceding vessels can overtake each other. Finally, a long-running time does not immediately have to be a problem. As the IWT systems are relatively slow, this will have less of an impact.

Table 6-37: Computation time of different numbers of vessels

Number of vessels	$w_\mu(k-\mu)$	**Running time [s]**
6	1	$\tilde{2}$ seconds
10	45	$\tilde{3}0$ seconds
12	66	$\tilde{8}500$ seconds

Table 6-38: The optimal IWT schedule for the BDVR case with locks in series in time units for 10 vessels sailing the waterway network based upon the data provided in **scenario A**

States	**Vessel 1**	**Vessel 2**	**Vessel 3**	**Vessel 4**	**Vessel 5**	**Vessel 6**	**Vessel 7**	**Vessel 8**	**Vessel 9**	**Vessel 10**
Route	Right	Left	Right	Right	Right	Left	Right	Left	Right	Right
Route	2	3	2	4	2	3	2	3	2	4
$x_1(k)$	1	120	3	130	5	140	7	160	9	150
$x_2(k)$	26	95	28	105	30	115	32	135	34	125
$x_3(k)$	28	93	58	103	38	113	68	133	48	123
$x_4(k)$	33	88	63	98	43	108	73	128	53	118
$x_5(k)$	35	86	65	88	45	96	75	106	55	99
$x_6(k)$	0	61	0	0	0	71	0	81	0	0
$x_7(k)$	0	59	0	0	0	69	0	79	0	0
$x_8(k)$	0	54	0	0	0	64	0	74	0	0
$x_9(k)$	0	52	0	0	0	56	0	58	0	0
$x_{10}(k)$	0	27	0	0	0	31	0	33	0	0
$x_{11}(k)$	60	0	90	63	70	0	100	0	80	74
$x_{12}(k)$	62	0	92	61	72	0	102	0	82	72
$x_{13}(k)$	67	0	97	56	77	0	107	0	87	67
$x_{14}(k)$	69	0	99	54	79	0	109	0	89	60
$x_{15}(k)$	94	0	124	29	104	0	134	0	114	35
$x_{16}(k)$	119	2	149	4	129	6	159	8	139	10

6-8 Conclusion

The goal of this Chapter 6 was to present the scheduling results generated by the scheduler. As stated in Chapter 1, this chapter addressed the research question:

How can we verify the designed scheduling strategy for IWT systems?

First of all, this chapter has shown what the scheduler requires as waterway network input data and vessel input data and how this should be presented to the scheduler. Furthermore, the scheduler outputs were defined, including the vessel schedule and properties, the object schedule and properties, and the global network properties. A difference was between output data that follows directly from the scheduler and output information which can be inferred

from the results. Moreover, it was shown how the results of the generated schedules are shared. For this purpose, a graphical method has been introduced.
Next, we have verified the designed scheduling strategy for IWT systems by executing various simulations to check if the scheduler behaves as expected. It can be seen from the results that the scheduling strategy ensures the vessels reach their arrival deadline, if feasible. Additionally, Section 6-2 and Section 6-7-1 show that if the scheduling strategy was based on the First-come, First-serve principle, vessels would have missed their arrival deadline. The schedule shows for every vessel its activity and location for every time unit. Moreover, the vessels are scheduled on a particular route and dynamically ordered through a lock while taking into account vessels coming from either side of the lock or coming from other routes. This can be done for different scenarios, for example, the number of vessels, different vessel sailing times, varying departure and arrival times or vessel priorities. Primarily the last result in Section 6-6 shows that we can model more types of waterway networks based on four cases.
This chapter finalises the core of the research. The next Chapter 7 will give some concluding remarks, state the contributions of this research and will present some recommendations for future research.

Chapter 7

Concluding remarks

At last, the final chapter of this thesis will review the previous chapters and give some concluding remarks. Chapter 7 is structured as follows: Firstly, the research questions as defined in Chapter 1 are answered in Section 7-1. Next, Section 7-2 will show the contribution of the research. At last, some final recommendations for future research are given in Section 7-3.

7-1 Conclusion and addressing the research questions

In this thesis, a comprehensive introduction to the Inland Waterway Transport (IWT) system and contemporary research on IWT scheduling was presented, which showed a lack of which showed that little research had been done on global waterway network optimisation. Moreover, a thorough introduction to the Max-Plus algebra and Switching Max-Plus Linear (SMPL) systems was given. Four IWT cases were modelled as SMPL systems, increasing in complexity to work towards a case that is as generic as possible. Additionally, it was described how SMPL systems could be used for optimal scheduling by transforming them into Mixed Integer Linear Programming (MILP) models. This was done for the four IWT cases as well. This resulted in a scheduling strategy that can be used to generate an optimal schedule in which the cumulative arrival time of an arbitrary number of vessels is minimised. The scheduling strategy was analysed and verified with various simulations. As a result, it can be concluded that SMPL systems can be used to describe IWT systems and used for scheduling within the scope and assumptions of this research. To answers the main research question:

'How to design and implement a Switching Max-Plus Linear scheduling strategy that allows multiple autonomous inland waterway vessels to optimally sail through a waterway network?'

In short, a Switching Max-Plus Linear scheduling strategy can be designed by describing the IWT system as Discrete Event System (DES) and denoting its dynamics with Max-Plus routing and Max-Plus ordering equations, these consist of Max-Plus binary control variables

and Max-Plus ordering control variables, which results in SMPL system. To implement a Switching Max-Plus Linear scheduling strategy the SMPL systems must be transformed to MILP models by making an approximating to convert the Max-Plus binary control variables to conventional control variables, which results in a large set of linear inequality constraints. These inequality constraints combined with a objective function which minimises the cumulative arrival time of all vessels results in a MILP model which produces a schedule which ensures inland waterway vessels to optimally sail through the waterway network.
Next we will review the in Chapter 1 formulated research questions which led to this aforementioned conclusion.

7-1-1 Current state of research on IWT scheduling

In Chapter 2, the following research questions were addressed:

1. *What is the current state of research on IWT scheduling?*
    - (a) *Which actors play a role in IWT systems and how do they operate?*
    - (b) *What research has been done on IWT scheduling?*
    - (c) *What area of IWT scheduling can be improved?*

In Chapter 2, we have described the current state of research on IWT scheduling, firstly by stating the operation principles of its most important actors (i.e. the waterways, the vessels and the locks), and secondly by giving an overview of contemporary research. It was found that there is currently no policy for coordination at infrastructure pieces, efficient scheduling of IWT could solve this problem and might yield a significant improvement in making IWT more attractive. There is limited research on the interaction between infrastructures and inland vessels. Moreover, communication between lock operators amongst each other and with inland vessel operators is lacking. As a result, there is minimal research on the operation and optimisation of infrastructure object. Moreover, there is no global network optimisation. When a delay at a particular lock occurs, the inland vessel's deadline or time window at the next infrastructure could be missed. This deadline missing will in turn, result in even more delays to the overall IWT system. Currently, lock operators are not reporting these delays. Scheduling between infrastructures or a global network optimisation could solve this.

7-1-2 Working principles of SMPL systems

In Chapter 3, the following research questions were addressed:

2. *How do SMPL systems work and what is required for modelling and scheduling?*
    - (a) *What is the motivation for SMPL modelling over other methods?*
    - (b) *What are the operators and definitions used in Max-Plus algebra?*
    - (c) *What is the system description of SMPL systems?*
    - (d) *What is required to model DESs as a SMPL systems?*
    - (e) *How to use SMPL systems for scheduling?*

(f) How to transform SMPL systems to MILP problems?

The main motivation for describing Discrete Event Systems with Max-Plus algebra is having the ability describe nonlinear DESs in a Max-Plus Linear way. Because of this, we end up with a state-space systems, which have a solid connection to conventional linear systems theory and can be analysed with some of the same tools. Moreover, there are system-theoretical analysis methods for Max-Plus Linear (MPL) systems. These MPL systems only consist of the 'oplus' and 'otimes' operators, which respectively correspond to the max-operator en plus-operator in conventional algebra. The drawback of MPL systems having a fixed model structure can be solved by allowing for switching. These systems are called Switching Max-Plus Linear (SMPL) systems. In order to model SMPL systems, three types of control decisions can be defined; routing, ordering and synchronisations. These SMPL systems can be used for scheduling by transforming them to Mixed Integer Linear Programming (MILP) models. This transformation is done by first transforming the Max-Plus binary control variables to conventional control variables using an approximation for the Max-Plus zero, after which the Max-Plus equations can be transformed to MILP constraints.

7-1-3 Modelling of IWT systems as a SMPL systems

In Chapter 4, the following research questions were addressed:

3. *How to model IWT systems as SMPL systems?*
   (a) *Why are SMPL systems useful for modelling IWT systems?*
   (b) *What assumptions have to be made to model IWT systems as SMPL systems?*
   (c) *What would the SMPL IWT system for different IWT cases look like?*

It was shown that SMPL systems are useful for modelling IWT systems when these are described as a DES. When describing two vessels sailing a simple IWT DES the in conventional algebra nonlinear maximum operator is required for defining the ordering relationship. This maximum operator is inherently nonlinear in conventional algebra. As the model structure of IWT systems is not fixed, for instance, the routing and ordering can change, SMPL systems are required, and MPL systems do not suffice. Several assumptions were made to model IWT systems as SMPL systems, with the main ones being; locks can only process one vessel at a time, vessel sailing speed is constant on the waterways, vessels cannot overtake inside locks and can overtake on waterways, and all information is assumed to know on beforehand. It is clear that in reality, locks can process multiple vessels at the same time and that this is a significant drawback of the model; however, using the synchronisation principle described in Section 3-5-3 it would be an interesting point of future research. At last, the SMPL systems for the four IWT cases were presented by stating their routing and ordering dynamics.

7-1-4 Transformation of SMPL IWT systems to MILP models

In Chapter 5, the following research questions were addressed:

4. *How to transform SMPL IWT systems to MILP models?*

    (a) *How to realise the objective of the scheduler?*

    (b) *What assumptions have to be made to transform SMPL IWT systems to MILP problems?*

    (c) *What would the MILP model for different IWT cases look like?*

In the context of this research, the objective of the scheduler is minimising the cumulative arrival time of all vessels. All information is assumed to be known constant and known, thus the scheduler operates in an offline open-loop way. SMPL IWT systems can be transformed to MILP models by transforming the Max-Plus routing equations and Max-Plus ordering equations to, respectively, MILP routing inequality constraints and MILP ordering inequality constraints. This is done by transforming the Max-Plus routing binary control variables and the Max-Plus ordering binary control variables to their conventional control variable counterparts by using an approximation for the Max-Plus zero. This results in a large set of linear inequality constraints. At last, the IWT models for the four IWT cases were presented by stating their routing and ordering dynamics.

7-1-5 Verification of MILP scheduling strategy for IWT systems

At last, in Chapter 6, the following research questions were addressed:

5. *How can we verify the designed scheduling strategy for IWT systems?*

    (a) *What does the overall scheduler optimisation architecture look like?*

        (i) *What information is required by the scheduler?*

        (ii) *What information is produced by the scheduler?*

    (b) *What are the results of the scheduling strategy for different IWT cases?*

The designed scheduling strategy was verified for IWT systems through various simulations for the four IWT cases. This showed showed that the scheduler behaves as expected and that the scheduling strategy ensures the vessels reach their arrival deadline. Moreover, an additional case was created to prove the scheduler can be extended to different waterway networks including multiple locks in a row. Some scenarios show that if the scheduling strategy was based on the First-come, First-serve principle, vessels would have missed their arrival deadline. However, this is not validated by comparing it with other scheduling strategies. This would be an interesting point for future research. Additionally, the scheduler requires waterway network input data and vessel input data. The scheduler produces the vessel schedule and properties, the object schedule and properties, and the global network properties. At last, the IWT models for the four IWT cases were presented by stating their routing and ordering dynamics.

7-2 Contribution of this research

In this research, we have developed a SMPL scheduling strategy for IWT systems. The aim of this research was to be relevant and contribute on an applied level to the development of IWT and on a theoretical level to extend the field of Max-Plus scheduling, as mentioned in Section 1-2 To the best knowledge of the author, seven contributions of this research have been identified and will be summarised in this section.

1) Literature review on IWT scheduling

First of all, a literature review on the current state of IWT scheduling was presented. This was done with a focus on the infrastructure perspective and overall IWT network optimisation since this seemed to be lacking in other available literature reviews.

2) Motivation for modelling IWT systems as SMPL systems

Next, the research thoroughly described the motivation and advantage of scheduling DESs with Max-Plus algebra. This was already done in literature for general systems as was shown in Appendix A-1 and Appendix A-3. However, it is a novelty and a contribution that in this research, the motivation and advantage of scheduling IWT systems as SMPL system. To recap, this is useful because of the appearance of the max-operator when a vessel is approaching a lock with another vessel being procces by the lock.

3) Developed method for modelling IWT systems as DESs and SMPL systems

Thirdly, in this research, we have developed a method for modelling IWT systems as DESs and SMPL. The first, a method for modelling IWT systems as DESs is very limited described in current research. To the best knowledge of the author, the latter, a method for modelling of IWT systems as SMPL systems is a novelty. The method presented showed how by starting with a small, simple waterway network and slowly building up the complexity and features over four cases, we could arrive at a system that resembles actual IWT systems quite well, based on simulations.

4) Modelled SMPL IWT systems for four cases

After developing the method for modelling IWT systems as DESs and SMPL systems, this method was used actually used for modelling four different SMPL IWT systems. This is also a contribution to the Max-Plus field as to the best knowledge of the author IWT systems have never been described as SMPL systems.

5) Transformed SMPL IWT systems to MILP models for four cases

Fifthly, MILP problems for IWT systems have been considered in literature before. For instance, the allocation of inland vessels to berth locations in port areas. However, the combination of both choosing the route of a vessel and the order in which they are allowed to enter a lock is a novelty to the best knowledge of the author.

6) Implementation and simulation of MILP IWT models
Next, we implemented and simulated the designed MILP IWT models, which resulted in a schedule plan for the vessels and locks. A visualisation technique for sharing the schedule plans was shown, from which the routing and ordering of the individual vessels and locks, can be inferred.

7) General overtaking for products within SMPL systems
At last, as a result of the vessels being allowed to overtake on waterways, we have described SMPL systems which can handle this. To the best knowledge of the author, this is a novelty and contributes to the field of Max-Plus. This can be used for general SMPL systems (e.g. a benchmark production system) in which products are allowed to pass each other, which was not described in literature before. Although it seems quite straightforward, it does create a interesting situation at the incoming node, as this node actually has an infinite number of arcs going into it.

7-3 Recommendations for future research

The aim of this research was to be as complete as possible, however, many research opportunities could not be explored yet. As this is the first study on modelling IWT systems as SMPL systems and we have shown it to be promising, there are several recommendations for future research opportunities and further modelling improvements. Five logical and relevant initial future research directions have been identified and will be summarised in this section.

1) Model and scheduling strategy validation
Firstly, we will describe possible ways to validate the designed model and scheduling strategy. Currently, we only verified the model and the scheduling strategies with simulations for arbitrary scenarios based on fictional data. As mentioned, these simulations have shown to behave as expected; however, to make sure that the model can be used to represent a real IWT system with some degree of accuracy, it has to be validated.
Of course, the model could be validated on real-world data; this should then include departure, intermediate and arrival times of vessels on a waterway network. Possibly the Automatic Identification System (AIS) data could be used for this validation. In the case the data is not available or does not suffice, the SMPL system could also be validated on other models which have already proven themselves. Two examples are SIVAK and OpenTNSim.
The *SIVAK* tool (Simulation Package for Traffic Flow at Engineering Structures) [5] was developed by the Dutch Ministry of Transport, Public Works and Water Management in 1998. SIVAK is a software system based on simulation models aimed at supporting studies on the vessel traffic flow on inland waterways and road traffic at bridges and locks. The designed SMPL system could be validated on this simulation package.
The *Open source Transport Network Simulation (OpenTNSim)* python package [52] for discrete-event-simulation of nautical-traffic, that offers the tools to simulate an inland fleet on an inland waterway network. Moreover, different lock scheduling policies can be tested and analysed.
At last, not only the model should be validated, but also the performance of the designed scheduling strategy should be validated. The current method for scheduling inland vessels

through locks is on *First-come First-serve* basis. It would be interesting to compare the results of the designed scheduling strategy with a scheduling strategy based on the First-come First-serve principle to assess which strategy results in fewer overall vessel delays.

2) Model extension

Next, the SMPL IWT system can be extended in multiple ways. First, the waterway network could be extended by considering *multiple chambers* within a single lock. This can be relatively easily implemented within Max-Plus routing equations as an additional route and will result in extra Max-Plus binary routing control variables. Furthermore, the waterway network could be extended by allowing for *multiple vessels within a single chamber*, which could be based on the synchronisation principle described in Section 3-5-3. We expect this to be more challenging. At last, regarding the model extension, it would be interesting for future research to extend the waterway network to not just an arbitrary waterway network, but to *existing waterway networks*, for instance, the Netherlands or Europe. This could be done with a node-set, which resembles that particular waterway network. Some data-sets can be found online, for example, for the waterway corridor between Rotterdam and Antwerp [52], this is shown in Figure 7-1.
Another extension could be to look at adding additional infrastructure pieces in the waterway network besides just locks. For instance, other objects such as movable bridges, inland ports or berth locations.

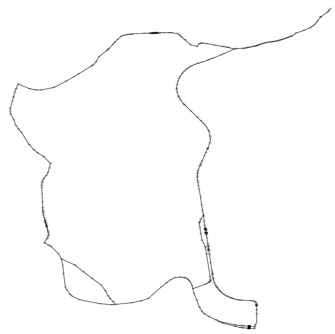


Figure 7-1: Node data set which describes the Rotterdam to Antwerpen waterway corridor from OpenTNSim [52]

3) Scheduling strategy extension

Thirdly, an essential addition to the scheduling strategy is to make it operate in a closed-loop online manner. This would be called operational scheduling or adaptive scheduling. This would allow for rescheduling in response to new vessel data and measurements (e.g. updated vessel speeds or arrival times) and unexpected events, such as disturbances (e.g. unexpected vessel departures) or disruptions (e.g. blocked waterway or broken lock). This could be done through *Model Predictive Scheduling* with a receding horizon principle which is extensively described in literature [51].
Another future research opportunity concerning the scheduling strategy is studying different *Different objective functions*. For instance, as mentioned in Section 5-1, in reality, it is not convenient for a vessel to arrive much earlier than the desired arrival deadline. Therefore, an

objective function could be developed which ensures arrival close to the arrival time. In that case, vessels might slow down vessels on particular waterways to reserve fuel and sail more efficiently.

4) Analysing the IWT networks

Next, as described in Section 3-1, the advantage of modelling nonlinear DESs as MPL systems or SMPL systems is the ability to use *system-theoretical methods* [22, 51] for analysing the network. By using these analysis methods we can measure how delays propagate through the waterway network or determine what the maximum vessel throughput of a IWT system could be (e.g. maximum growth rate). This could help answer practical questions like; what waterway would have the most significant impact on the overall performance of the waterway network when a vessel gets stuck? With the current waterway network infrastructure in place (i.e. locks) how much can we increase the marine traffic flow with while still having no vessels ever wait for a lock? Which lock forms a bottleneck in the IWT system and should therefore be upgraded (e.g. add a chamber) first? Moreover, even SMPL systems that switch randomly to different modes have been extensively described in literature [48, 49] which can be used to measure the sensitivity of a IWT system with respect to disturbances (e.g. unexpected vessel departures) or disruptions (e.g. blocked waterway or broken lock).

5) Improve computational efficiency

At last, the computational efficiency of the scheduler could be improved through *Distributed Model Predictive Scheduling*. In [28] it has been shown that MILP problems such as derived in this research can be partitioned into several parallel solvable sub-problems. This resulted in a significant reduction of the computation time of the scheduler. The application it was used on, a railway network, required this fast computation time because there are relatively quick decisions to be made such as whether a train should be delayed at a particular station or not. We do not expect this to be the case for the IWT system because vessels move a lot slower than trains, so the speed of these control actions is less critical. However, enlarging the waterway network (e.g. extending it to cover the whole of the Netherlands or Europe) will make the MILP problem enormous, and Distributed Model Predictive Scheduling could help to reduce this a lot. Naturally, this future research opportunity should only be explored after studying a centralised Model Predictive Scheduling strategy.
Moreover, to deal with the problem of having more ordering control actions than vessel permutations as described in Section 6-7-4, constraints could be added to eliminate these infeasible combinations, as presented in [26]. This would Improve computational efficiency as well, as fewer control actions have to be evaluated.

Appendix A

Additional theory on Max-Plus Algebra

This Appendix will present an example to clearly show the motivation behind Max-Plus algebra, additional information and statements of the Max-Plus algebra, and an example of the Max-Plus Linear (MPL) system description

A-1 Motivation for Max-Plus algebra using a Petri net example

This section will clearly show the motivation for using Max-Plus algebra, as supplement to Section 3-1. Many models for graphically visualising Discrete Event System (DES)s have been developed. In this example we will use the Petri net framework [8]. Petri net graphs capture a lot of information about the system and explicitly represent the transition of a DES. In addition, they are intuitive and clearly show the motivation behind the use of Max-Plus algebra.

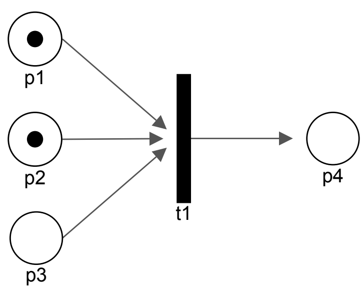


Figure A-1: Basic Petri net example, which consists of the four places $\{p_1, p_2, p_3, p_4\}$, one transition $\{t_1\}$ with two tokens being in the places $\{p_1, p_2\}$. Adapted figure from [22].

In Figure A-1, a basic example of a Petri net is given that consist of three essential components. The *places* $\{p_1, p_2, p_3, p_4\}$ indicate the necessary condition for a *transition* $\{t_1\}$ to occur. This condition is satisfied if a *token* is in every place corresponding to the specific transition. For this example the transition $\{t_1\}$ will fire if a token is in the places $\{p_1, p_2, p_3\}$. As it is drawn

now, there are only tokens in the places $\{p_1, p_2\}$ thus the transition $\{t_1\}$ cannot fire. Assigning tokens to places is analogous to describing the state of a DES and transitions are analogues to events which in term can be dependent on the state. The activation or firing of a transition sets off a state or event transition and moves the tokens from transition input to transition output. This means that for the example in Figure A-1, putting a token in place $\{p_3\}$ would result in firing transition $\{t_1\}$ and thus the three tokens would move to $\{p_4\}$. This progression of tokens through the Petri net graph makes it very useful for modelling the order of events and evolution of states.
Following the example of the baggage handling system introduced in Section 1-1, we could view the places $\{p_1, p_2, p_3\}$ as a storage room that can hold three luggage pieces. Place $\{p_4\}$ would be the event when the storage is full and the plane can be loaded.

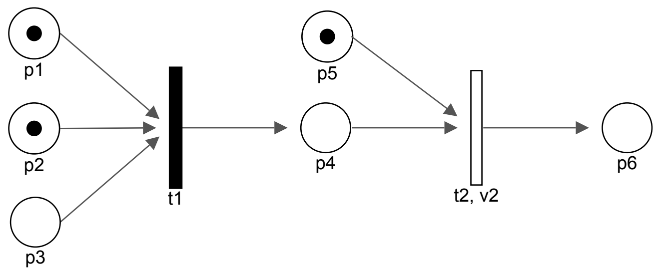


Figure A-2: Extended Petri net example which consists of, six places $\{p_1, p_2, p_3, p_4, p_5, p_6\}$, one untimed transition $\{t_1\}$, one timed transition $\{t_2\}$ with three tokens being in the places $\{p_1, p_2, p_5\}$. Adapted figure from [22].

The transition $\{t_1\}$ is not affected by a transition time and we assumed that the transition would immediately fire when a token was placed in $\{p_3\}$. Thus for the previous example of Figure A-1, the Petri net's state evolution can be completely described by a sequence of states. If we denote x_k as the state of the system after event k then the complete untimed state evolution can be described by sequence expressed as $\{x_0, x_1, ...\}$. This is called *untimed.* In contrast, a system is called *timed* when some form of timing is involved in order to describe the state evolution. If we denote t_k as the moment when the kth transition occurs then the timed state evolution sequence can be described as $\{(x_0, t_0), (x_1, t_1), ...\}$. This can be seen in the next example in Figure A-2. Again, following the same baggage handling system example, we could see $\{p_5\}$ as a worker that has to be ready to load luggage into the plane and $\{p_6\}$ could be that the place is ready to leave.
Here we have the same untimed transition t_1 and a timed transition t_2 with a *clock sequence* denoted by v_2. This clock sequence is defined by [8] as a positive real number, $v_{j,k}$, assigned to transition t_j. It behaves as follows, when transition t_j is enabled for the kth time, it does not fire immediately, but incurs a firing delay of $v_{j,k}$ time units; during this delay, tokens are kept in the input places of t_j. Thus the clock sequence has the following form:

$$v_j = \{v_{j,1}, v_{j,2}, \ldots\}, \quad v_{j,k} \in \mathbb{R}^+, \quad k = 1, 2, \ldots \tag{A-1}$$

If we relate this to the example in Figure A-2, then when a token arrives in place p_3 the untimed transition t_1 will immediately fire and the tokens are transferred to place p_4. Next, the timed transition t_2 will fire after $v_{2,1}$ time units. Then when there are tokens in the places $\{p_4, p_5\}$ again, the timed transition t_2 will fire after $v_{2,2}$ time units, which could be equal to

the earlier clock sequence $v_{2,1}$ but not necessarily.
Next we extend this model. First, we define $\pi_{i,k}$ as the time that place p_i receives a token for the kth time and $x(p_i)$ denotes the number of tokens at place p_i. If we denote firing time of transition t_j for the kth time by $\tau_{j,k}$ and we assume a very simple example where p_i is the only input place of untimed transition t_j the the following relationship hold:

$$\tau_{j,k} = \pi_{i,k}, \quad k = 1, 2, \ldots \tag{A-2}$$

This state the fact that the untimed transition t_j fires immediately when the only input place p_i receives a token. If t_j would be a timed transition will clock sequence v_k the relationship would look like this:

$$\tau_{j,k} = \pi_{i,k} + v_{j,k}, \quad k = 1, 2, \ldots \tag{A-3}$$

Modelling the dynamics of the first example in Figure A-1 requires some additional attention. Transition t_1 will only fire after the last one of the three input places $\{p_1, p_2, p_3\}$ receives a token. For how the tokens are currently places this would be place p_3 but of course this does not have to be the case. The last input place of transition t_j that receives a token is denoted by p_s and thus the time that this token is received is $\pi_{s,k}$. Then we get the following relationship:

$$\pi_{s,k} \geq \pi_{i,k}, \quad \forall p_i \in I(t_j) \tag{A-4}$$

Where $I(t_j)$ is the set of all input places of transition t_j. Of course it is not always known which places receives the last token, so we do not know which place is p_s. Therefore, the time the timed transition t_j fires is defined as follows:

$$\tau_{j,k} = \max_{p_i \in I(t_j)} \{\pi_{i,k}\} + v_{j,k}, \quad k = 1, 2, \ldots \tag{A-5}$$

Note the appearance of the max-operator, which is inherently nonlinear in conventional algebra. As shown, some mathematical models that use conventional algebra to describe the behaviour of DESs will result in a nonlinear system description due to this max-operator [3]. As explained in Chapter 3, we can 'linearise' this model by using Max-Plus algebra (i.e. only maximisation and addition operators), which results in a MPL systems. Further advantages are given in Chapter 3.

A-2 Algebraic properties of the Max-Plus operators

This section will elaborate on algebraic properties of the Max-Plus algebra, as supplement to Section 3-2. Assume $\forall\{x, y, z\} \in \mathbb{R}_{\max}$ then the following statements hold:

- *Associativity*
  Rearranging the parentheses in an expression will not change the result. In other words, the order of operations does not matter when, the operator sequence remains the same:

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z \quad \text{and} \quad x \otimes (y \otimes z) = (x \otimes y) \otimes z \tag{A-6}$$

- *Commutativity*
  Rearranging the order of elements in an expression will not change the result

$$x \oplus y = y \oplus x \quad \text{and} \quad x \otimes y = y \otimes x \tag{A-7}$$

- *Distributivity of $\otimes$ over $\oplus$:*
  When the 'otimes' $\otimes$ operation is performed on a 'oplus' $\oplus$ operation, each individual element of the 'oplus' $\oplus$ operations is max-plus multiplied. Compared to conventional algebra, this is quite intuitive:

$$x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z) \tag{A-8}$$

- *Existence of a zero-element:*
  As shown in Equation 3-3, the zero-element is defined as:

$$x \oplus \varepsilon = \varepsilon \oplus x = x \tag{A-9}$$

- *Existence of a unit-element:*
  As shown in Equation 3-4, the unit-element is defined as:

$$x \otimes e = e \otimes x = x \tag{A-10}$$

- *The zero-element ϵ is absorbing for $\otimes$*
  The zero-element in combinations with the 'otimes' $\otimes$ operator absorbs the value of x:

$$x \otimes \varepsilon = \varepsilon \otimes x = \varepsilon \tag{A-11}$$

- *Idempotency of $\oplus$*
  Multiple times of performing the 'oplus' $\oplus$ operation will not change the result:

$$x \oplus x \oplus x \oplus x = x \tag{A-12}$$

All the above listed properties apply to conventional algebra, excluding the last one idempotency. Therefore, the max-plus algebra is defined as a so called *idempotent semiring* which is defined in [22] as:

Definition 4 (*Idempotent semiring*)**.** *"A semiring is a nonempty set R endowed with two binary operations $\oplus_R$ and $\otimes_R$ such that:*

- *$\oplus_R$ is associative and commutative with zero-element ϵ_R;*
- *$\otimes_R$ is associative, distributes over $\oplus_R$, and has unit element e_R;*
- *ϵ_R is absorbing for $\otimes_R$.*

Such a semiring is denoted by $\mathcal{R} = (R, \oplus_R, \otimes_R, \varepsilon_R, e_R)$. If $\otimes_R$ is commutative, then $\mathcal{R}$ is called commutative, and if $\oplus_R$ is idemempotent, then it is called idempotent"

These are also summarised and compared to conventional algebra in Table 3-1

A-3 Simple Max-Plus Linear model railway example

This section will give a by hand calculated worked out example of a MPL railway system [22], as supplement to Section 3-3.

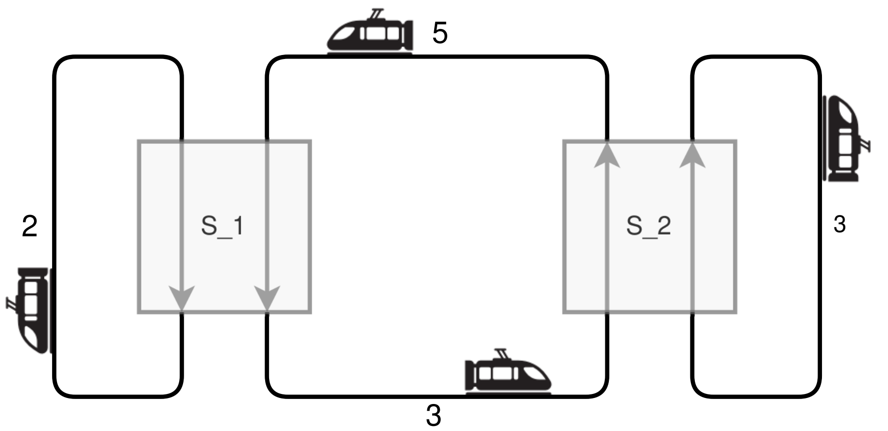


Figure A-3: Railway system example with two stations and four track connections. The numbers along the tracks refer to the travel times. Adapted figure from [22].

The example consist of two train stations and four connection tracks with corresponding travel times as in Figure A-3. At the stations the passengers will change between trains thus one train cannot leaver before the arrival of the other train and the trains will depart always at the same time. We denote the kth departure at station i with $x_i(k)$. Then we can give a lower-bound on the next $(k+1)$th departure time. For station S_1 and S_2 this becomes respectively:

$$\begin{aligned} x_1(k+1) &\geq x_1(k) + 2, \quad x_2(k+1) \geq x_2(k) + 3 \\ x_1(k+1) &\geq x_2(k) + 5, \quad x_2(k+1) \geq x_1(k) + 3 \end{aligned} \tag{A-13}$$

Then using the max-operator we get:

$$x_1(k+1) = \max(x_1(k) + 2, x_2(k) + 5), \quad x_2(k+1) = \max(x_1(k) + 3, x_2(k) + 3) \tag{A-14}$$

If we denote the number of stations with n and the travel time between station j and i by a_{ij}, we get the following generalised departure times:

$$x_i(k+1) = \max_{j=1,2,\ldots,n} \left(x_j(k) + a_{ij} \right), \quad i = 1, 2, \ldots, n \tag{A-15}$$

We can write this into max-plus algebra as:

$$x_i(k+1) = \bigoplus_{j=1}^{n} x_j(k) \otimes a_{ij}, \quad i = 1, 2, \ldots, n \tag{A-16}$$

Again, note the resemblances with conventional algebra :

$$y_i(k+1) = \sum_{j=1}^{n} y_j(k) \times c_{ij}, \quad i = 1, 2, \ldots, n \tag{A-17}$$

This can be written in state-space form as:

$$x\,(k+1) = A \otimes x\,(k) \tag{A-18}$$

Working this out will look like this:

$$\begin{aligned} x\,(1) &= A \otimes x\,(0) \\ x\,(2) &= A \otimes x\,(1) \\ &= A \otimes (A \otimes x\,(0)) \\ &= A^{\otimes 2} \otimes x\,(0) \end{aligned} \tag{A-19}$$

This can be generalised for $x(k)$ in the following way:

$$\begin{aligned} x(k) &= A \otimes x(k-1) \\ &= A \otimes \left(A^{\otimes(k-1)} \otimes x(0)\right) \\ &= (\underbrace{A \otimes A \otimes \ldots \otimes A}_{k \text{ times}}) \otimes x(0) \\ &= A^{\otimes k} \otimes x(0) \end{aligned} \tag{A-20}$$

The matrix A accounts for the recurrence of Equation (A-14), which is represented by the following states and A matrix:

$$x(k) = \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix} \tag{A-21}$$

Thus we get the max-plus-linear state-space description:

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix} \otimes \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} \tag{A-22}$$

Now taking a arbitrary initial departure time for instance $x(0) = (1, 0)^T$ the timing of the third departure will be given by $x(2)$, thus:

$$\begin{aligned} x(2) &= A^{\otimes 2} \otimes x(0) \\ &= \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix} \otimes \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix} \otimes x(0) \\ &= \begin{pmatrix} (2 \otimes 2) \oplus (5 \otimes 3) & (2 \otimes 5) \oplus (5 \otimes 3) \\ (3 \otimes 2) \oplus (3 \otimes 3) & (3 \otimes 5) \oplus (3 \otimes 3) \end{pmatrix} \otimes x(0) \\ &= \begin{pmatrix} 8 & 8 \\ 6 & 8 \end{pmatrix} \otimes x(0) \\ &= \begin{pmatrix} 9 \\ 8 \end{pmatrix} \end{aligned} \tag{A-23}$$

This calculation can be massively simplified by using the eigenvalues and eigenvectors of A. Looking back at the general statement in Equation (A-20), this calculation can be simplified to

$$x(k) = \lambda^{\otimes k} \otimes x(0), k \geq 0 \tag{A-24}$$

If and only if, $x(0)$ is an eigenvector of A that belongs to the corresponding eigenvalue λ. Then, using the earlier defined relationship for max-plus eigenvectors and eigenvalues in Equation (3-14), the following relationship for the A matrix of the max-plus-linear system in Equation A-22 holds:

$$\begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix} \otimes \begin{pmatrix} 1+h \\ h \end{pmatrix} = 4 \otimes \begin{pmatrix} 1+h \\ h \end{pmatrix}, \quad \forall h \in \mathbb{R} \tag{A-25}$$

Thus, the railway system from Figure A-3 has an eigenvalue $\lambda = 4$ corresponding to an eigenvector of the form $v = (1+h, h)^T$. Then using Equation (A-24) we can easily compute the next departure or synchronisation times:

$$x(2) = \lambda^{\otimes 2} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = (4 \otimes 4) \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 9 \\ 8 \end{pmatrix} \tag{A-26}$$

The railway system's eigenvalue denotes the average time between departures and is thus a measure of the network's capacity. Taking the corresponding eigenvector as the initial condition leads to an easily computable timetable. As is shown, these two characteristics of a MPL system are beneficial when optimising network throughput.

Appendix B

Addition to the results for the cases

The following Appendix will present the definitions of states and some additional visualisations for the results for the cases presented in Chapter 6.

B-1 Uni-Directional Fixed Routing case

B-1-1 State definitions

Table B-1: Definitions of state variables UDFR case

State event variable	Description
$x_1(k)$	Time instant when vessel k enters waterway (1,2)
$x_2(k)$	Time instant when vessel k enters waiting area (2,3) and leaves waterway (1,2)
$x_3(k)$	Time instant when vessel k enters the lock (3,4) and leaves waiting area (2,3)
$x_4(k)$	Time instant when vessel k enters waiting area (4,5) and leaves the lock (3,4)
$x_5(k)$	Time instant when vessel k enters waterway (5,6) and leaves waiting area (4,5)
$x_6(k)$	Time instant when vessel k leaves waterway (5,6)

B-1-2 Visualisation of delays

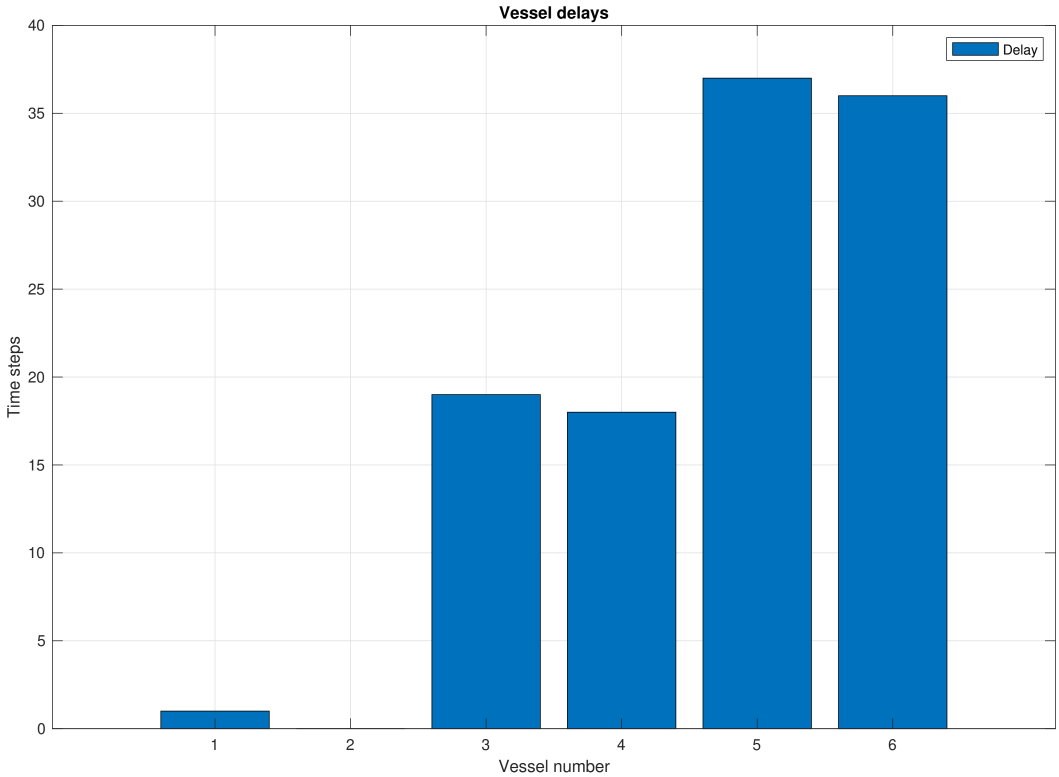


Figure B-1: UDFR case: Vessel delay visualisation

B-2 Uni-Directional Variable Routing case

B-2-1 State definitions

Table B-2: Definitions of state variables UDVR case

State event variable	Description
$x_1(k)$	Time instant when vessel k enters waterway (1,2) or (1,7)
$x_2(k)$	Time instant when vessel k enters waiting area (2,3) and leaves waterway (1,2)
$x_3(k)$	Time instant when vessel k enters the lock (3,4) and leaves waiting area (2,3)
$x_4(k)$	Time instant when vessel k enters waiting area (4,5) and leaves the lock (3,4)
$x_5(k)$	Time instant when vessel k enters waterway (5,6) and leaves waiting area (4,5)
$x_6(k)$	Time instant when vessel k leaves waterway (5,6) and enters waterway (6,12)
$x_7(k)$	Time instant when vessel k enters waiting area (7,8) and leaves waterway (1,7)
$x_8(k)$	Time instant when vessel k enters the lock (8,9) and leaves waiting area (7,8)
$x_9(k)$	Time instant when vessel k enters waiting area (9,10) and leaves the lock (8,9)
$x_{10}(k)$	Time instant when vessel k enters waterway (10,11) and leaves waiting area (9,10)
$x_{11}(k)$	Time instant when vessel k enters waterway (11,12) and leaves waiting area (10,11)
$x_{12}(k)$	Time instant when vessel k leaves waterway (11,12) or leaves waterway (6,12)

B-3 Bi-Directional Fixed Routing case

B-3-1 State definitions

Table B-3: Definitions of state variables BDFR case

State event variable	Description
$x_1(k)$	Time instant when vessel k enters waterway (1,2)
$x_2(k)$	Time instant when vessel k enters waiting area (2,3) and leaves waterway (1,2)
$x_3(k)$	Time instant when vessel k enters the lock (3,4) and leaves waiting area (2,3)
$x_4(k)$	Time instant when vessel k enters waiting area (4,5) and leaves the lock (3,4)
$x_5(k)$	Time instant when vessel k enters waterway (5,6) and leaves waiting area (4,5)
$x_6(k)$	Time instant when vessel k leaves waterway (5,6)

B-3-2 Visualisation of delays

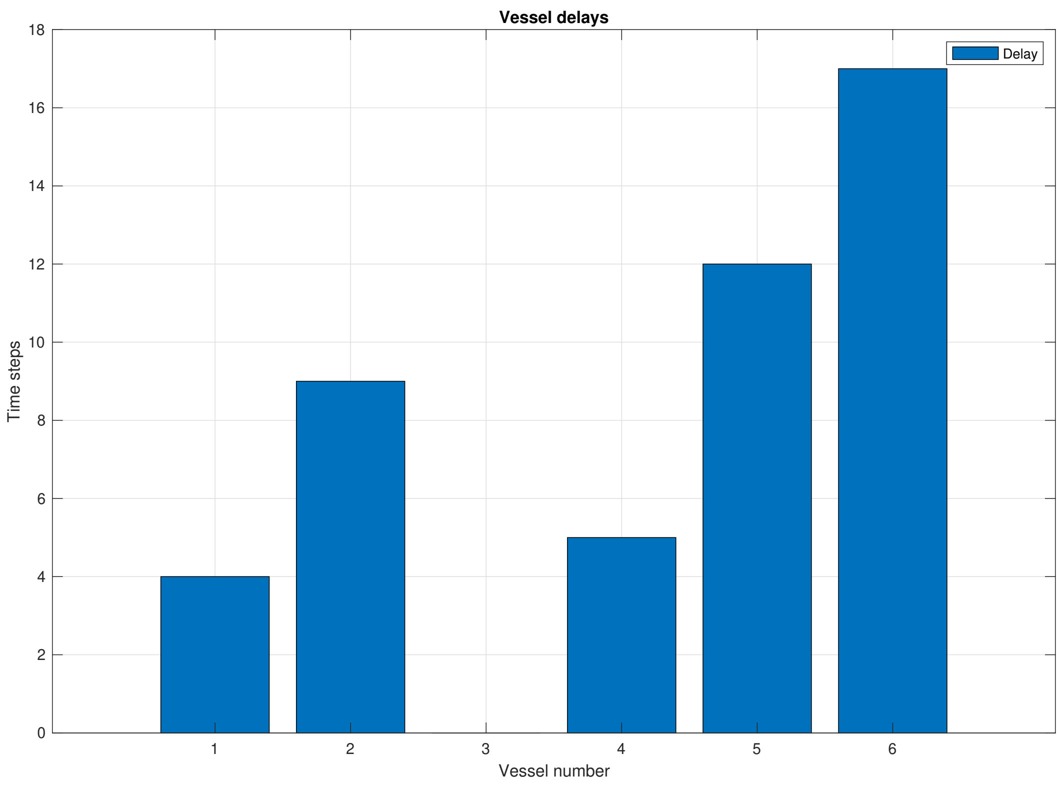


Figure B-2: BDFR case: Vessel delay visualisation

B-4 Bi-Directional Variable Routing case

B-4-1 State definitions

Table B-4: Definitions of state variables BDVR case

State event variable	Description
$x_1(k)$	Time instant when vessel k enters waterway (1,2) or (1,7)
$x_2(k)$	Time instant when vessel k enters waiting area (2,3) and leaves waterway (1,2)
$x_3(k)$	Time instant when vessel k enters the lock (3,4) and leaves waiting area (2,3)
$x_4(k)$	Time instant when vessel k enters waiting area (4,5) and leaves the lock (3,4)
$x_5(k)$	Time instant when vessel k enters waterway (5,6) and leaves waiting area (4,5)
$x_6(k)$	Time instant when vessel k leaves waterway (5,6) and enters waterway (6,12)
$x_7(k)$	Time instant when vessel k enters waiting area (7,8) and leaves waterway (1,7)
$x_8(k)$	Time instant when vessel k enters the lock (8,9) and leaves waiting area (7,8)
$x_9(k)$	Time instant when vessel k enters waiting area (9,10) and leaves the lock (8,9)
$x_{10}(k)$	Time instant when vessel k enters waterway (10,11) and leaves waiting area (9,10)
$x_{11}(k)$	Time instant when vessel k enters waterway (11,12) and leaves waiting area (10,11)
$x_{12}(k)$	Time instant when vessel k leaves waterway (11,12) or leaves waterway (6,12)

B-4-2 Visualisation of delays

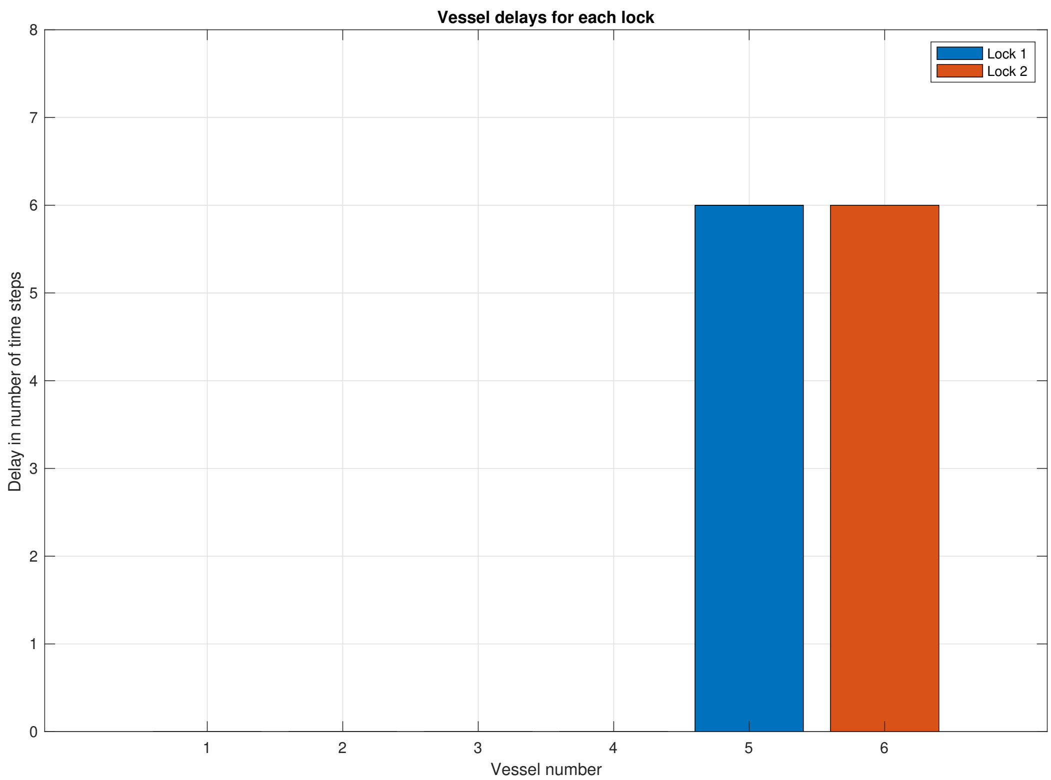


Figure B-3: BDVR case: Vessel delay visualisation

B-5 Bi-Directional Variable Routing case with locks in series

B-5-1 State definitions

Table B-5: Definitions of state variables BDVR case with locks in series

State event variable	Description
$x_1(k)$	Time instant when vessel k enters waterway (1,2)
$x_2(k)$	Time instant when vessel k enters waiting area (2,3) and leaves waterway (1,2)
$x_3(k)$	Time instant when vessel k enters the lock (3,4) and leaves waiting area (2,3)
$x_4(k)$	Time instant when vessel k enters waiting area (4,5) and leaves the lock (3,4)
$x_5(k)$	Time instant when vessel k enters waterway (5,6) or (5,11) and leaves waiting area (4,5)
$x_6(k)$	Time instant when vessel k leaves waterway (5,6) and enters waiting area (6,7)
$x_7(k)$	Time instant when vessel k enters the lock (7,8) and leaves waiting area (5,6)
$x_8(k)$	Time instant when vessel k enters waiting area (8,9) and leaves the lock (7,8)
$x_9(k)$	Time instant when vessel k enters waterway (9,10) and leaves waiting area (8,9)
$x_{10}(k)$	Time instant when vessel k enters waterway (10,16) and leaves waterway (9,10)
$x_{11}(k)$	Time instant when vessel k leaves waterway (5,11) and enters waiting area (11,12)
$x_{12}(k)$	Time instant when vessel k enters the lock (12,13) and leaves waiting area (11,12)
$x_{13}(k)$	Time instant when vessel k enters waiting area (13,14) and leaves the lock (12,13)
$x_{14}(k)$	Time instant when vessel k enters waterway (14,15) and leaves waiting area (13,14)
$x_{15}(k)$	Time instant when vessel k enters waterway (15,16) and leaves waterway (14,15)
$x_{16}(k)$	Time instant when vessel k leaves waterway (15,16) or leaves waterway (10,16)

B-5-2 Visualisation of delays

No delays occur at lock 2, therefore no red bar is visible in the graph, see Section 6-6.

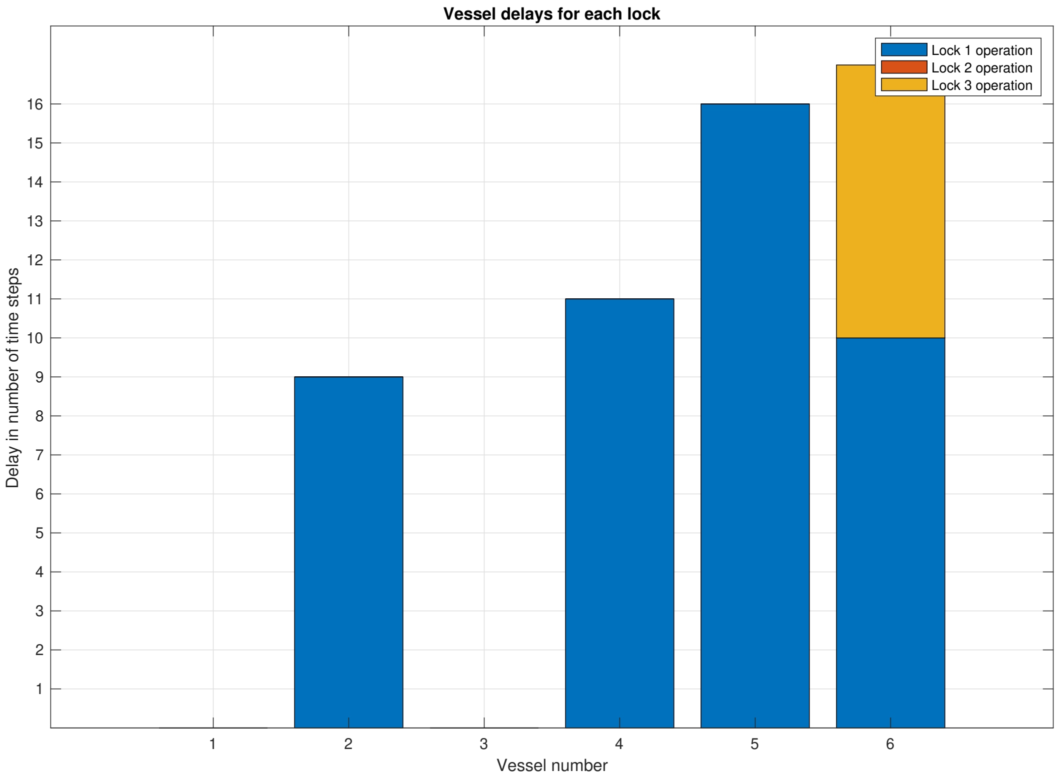


Figure B-4: BDVR case with locks in series: Vessel delay visualisation

Appendix C

Addition to the results for the scenarios

The following Appendix will present the data tables for the results for the scenarios presented in Chapter 6-7.

C-1 Scenario A

Table C-1: The optimal Inland Waterway Transport (IWT) schedule for the **scenario A** Bi-Directional Variable Routing (BDVR) case with locks in series in time units, which includes; the route the vessels are scheduled on, the intermediate event timings $x_i(k)$ and the arrival time at $x_1(k)$ or $x_{16}(k)$ for 6 vessels sailing the waterway network

States	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Route	Left	Right	Right	Left	Left	Right
$x_1(k)$	1	2	121	102	5	84
$x_2(k)$	26	22	96	82	20	69
$x_3(k)$	28	38	94	80	48	67
$x_4(k)$	33	43	89	75	53	62
$x_5(k)$	35	45	87	73	55	60
$x_6(k)$	60	-	-	53	70	-
$x_7(k)$	62	-	-	51	72	-
$x_8(k)$	67	-	-	46	77	-
$x_9(k)$	69	-	-	44	79	-
$x_{10}(k)$	94	-	-	24	94	-
$x_{11}(k)$	-	65	62	-	-	45
$x_{12}(k)$	-	67	60	-	-	43
$x_{13}(k)$	-	72	55	-	-	38
$x_{14}(k)$	-	74	53	-	-	36
$x_{15}(k)$	-	94	28	-	-	21
$x_{16}(k)$	119	114	3	4	109	6

C-2 Scenario B

Table C-2: The optimal IWT schedule for the **scenario B** BDVR case with locks in series in time units, which includes; the route the vessels are scheduled on, the intermediate event timings $x_i(k)$ and the arrival time at $x_1(k)$ or $x_{16}(k)$ for 6 vessels sailing the waterway network

States	**Vessel 1**	**Vessel 2**	**Vessel 3**	**Vessel 4**	**Vessel 5**	**Vessel 6**
Route	Right	Right	Right	Left	Left	Right
$x_1(k)$	1	2	121	102	5	84
$x_2(k)$	26	22	96	82	20	69
$x_3(k)$	28	48	94	80	38	67
$x_4(k)$	33	53	89	75	43	62
$x_5(k)$	35	55	87	73	45	60
$x_6(k)$	-	-	-	53	60	-
$x_7(k)$	-	-	-	51	62	-
$x_8(k)$	-	-	-	46	67	-
$x_9(k)$	-	-	-	44	69	-
$x_{10}(k)$	-	-	-	24	84	-
$x_{11}(k)$	60	75	62	-	-	45
$x_{12}(k)$	62	77	60	-	-	43
$x_{13}(k)$	67	82	55	-	-	38
$x_{14}(k)$	69	84	53	-	-	36
$x_{15}(k)$	94	104	28	-	-	21
$x_{16}(k)$	119	124	3	4	99	6

C-3 Scenario C

Table C-3: The optimal IWT schedule for the **scenario B** BDVR case with locks in series in time units, which includes; the route the vessels are scheduled on, the intermediate event timings $x_i(k)$ and the arrival time at $x_1(k)$ or $x_{16}(k)$ for 6 vessels sailing the waterway network

States	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Route	Right	Right	Right	Right	Left	Left
$x_1(k)$	1	2	124	102	5	84
$x_2(k)$	26	22	99	82	20	69
$x_3(k)$	44	24	97	80	34	67
$x_4(k)$	49	29	92	75	39	62
$x_5(k)$	51	31	90	73	41	60
$x_6(k)$	-	-	-	-	56	45
$x_7(k)$	-	-	-	-	58	43
$x_8(k)$	-	-	-	-	63	38
$x_9(k)$	-	-	-	-	65	36
$x_{10}(k)$	-	-	-	-	80	21
$x_{11}(k)$	76	51	65	53	-	0
$x_{12}(k)$	78	53	63	51	-	0
$x_{13}(k)$	83	58	58	46	-	0
$x_{14}(k)$	85	60	53	44	-	0
$x_{15}(k)$	110	80	28	24	-	0
$x_{16}(k)$	135	100	3	4	95	6

C-4 Scenario D

```
Explored 9324937 nodes (263460910 simplex iterations) in 8406.87 seconds
Thread count was 8 (of 8 available processors)

Solution count 10: 1812.85 1812.85 1812.85 ... 1959.93

Optimal solution found (tolerance 1.00e-04)
Best objective 1.812854600000e+03, best bound 1.812698825278e+03, gap 0.0086%
```

Vessel	Route	x01	x02	x03	x04	x05	x06	x07	x08	x09	x10	x11	x12	x13	x14	x15	x16
1	2	1	26	28	33	35	0	0	0	0	0	60	62	67	69	94	119
2	3	120	95	93	88	86	61	59	54	52	27	0	0	0	0	0	2
3	2	3	28	58	63	65	0	0	0	0	0	90	92	97	99	124	149
4	4	140	115	113	108	88	0	0	0	0	0	63	61	56	54	29	4
5	2	5	30	68	73	75	0	0	0	0	0	100	102	107	109	134	159
6	4	160	135	133	128	109	0	0	0	0	0	84	82	77	56	31	6
7	1	7	32	48	53	55	80	82	87	89	114	0	0	0	0	0	139
8	4	150	125	123	118	99	0	0	0	0	0	74	72	67	58	33	8
9	2	9	34	38	43	45	0	0	0	0	0	70	72	77	79	104	129
10	3	130	105	103	98	96	71	69	64	60	35	0	0	0	0	0	10
11	2	11	36	78	83	85	0	0	0	0	0	110	112	117	119	144	169
12	3	170	145	143	138	106	81	79	74	62	37	0	0	0	0	0	12

Figure C-1: MATLAB GUROBI results for the optimal IWT schedule for **scenario C** BDVR case with lock in series in time units

Bibliography

[1] M. Alirezaei, Ton J.J. van den Boom, and R. Babuska. Max-plus algebra for optimal scheduling of multiple sheets in a printer. *Proceedings of the American Control Conference*, (3):1973–1978, 2012.

[2] Koh Ann, El Wardany Salma, and Clark Aaron. Suez Canal Snarled by Giant Ship Choking Key Trade Route, 2021.

[3] Francois Baccelli, Charles Leake, Guy Cohen, Geert Jan Olsder, and Jean-Pierre Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems.*, volume 45. 1992.

[4] Jeff Belt. New York State Canal System. *New York State Department of Transportation*, 2010.

[5] R Bijlsma. *SIVAK Documentation.* Date accessed: 2022-02-15, 2019.

[6] James F. Campbell, L. Douglas Smith, and Donald C. Sweeney. A robust strategy for managing congestion at locks on the upper Mississippi river. *Proceedings of the 42nd Annual Hawaii International Conference on System Sciences, HICSS*, pages 1–10, 2009.

[7] James F. Campbell, L. Douglas Smith, Donald C. Sweeney, Ray Mundy, and Robert M. Nauss. Decision tools for reducing congestion at locks on the upper Mississippi river. *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 1–10, 2007.

[8] Christos G. Cassandras and Stéphane Lafortune. Introduction to discrete event systems. *Springer Science & Business Media*, 37(10):1682–1685, 2008.

[9] Centrale Commissie voor de Rijnvaart (CCR). Informatieblad Inland AIS. 2011.

[10] Bo Chen, Chris N Potts, and Gerhard J Woeginger. A Review of Machine Scheduling: Complexity, Algorithms and Approximability B. pages 1493–1641. Springer US, Boston, MA, 1998.

[11] L. Chen, Rudy Negenborn, Yamin Huang, and Hans Hopman. Survey on Cooperative Control for Waterborne Transport. *IEEE Intelligent Transportation Systems Magazine*, (October), 2020.

[12] George Bernard Dantzig, Alexander Ordern, and Philip Wolfe. The Generalized Simplex Method. *Pacific Journal of Mathematics*, 5(2):183–195, 1955.

[13] B de schutter and T van den Boom. Model predictive control for max-plus-linear discrete-event systems: Extended report. *Automatica*, 37(7):1049–1056, 2001.

[14] Bart de Schutter, Ton J.J. van den Boom, Jia Xu, and Samira S. Farahani. Analysis and control of max-plus linear discrete-event systems: An introduction. *Discrete Event Dynamic Systems: Theory and Applications*, 30(1):25–54, 2020.

[15] Kees de Vries. Waardevol Transport De toekomst van het goederenvervoer en de binnenvaart in Europa. *Bureau Voorlichting Binnenvaart*, 2017.

[16] Ton J.J.van den Boom, Hilco de Bruijn, Bart De Schutter, and Leyla Özkan. The interaction between scheduling and control of semi-cyclic hybrid systems. *IFAC-PapersOnLine*, 51(7):212–217, 2018.

[17] Economic Commission for Europe Inland Transport Commottee. Inventory of Most Important Bottlenecks and Missing Links in the E Waterway Network. (49), 2013.

[18] European Commission. Roadmap to a Single European Transport Area – Towards a competitive and resource efficient transport system. 2011.

[19] European Commission. Promotion of inland waterway transport, 2015.

[20] Andrzej S. Grzelakowski. Assessment of the current state of the EU inland shipping development and its perspectives from the policy and transport market point of view. *Scientific Journals of the Maritime University of Szczecin*, 58(130):74–83, 2019.

[21] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[22] Bernd Heidergott, Geert Jan Olsder, and Jacob van der Woude. *Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*. 2005.

[23] Inland Navigation Europe. Towards a Strategic research Agenda: Inland Waterways Transport and ports in Horizon Europe. 2019.

[24] Inland Navigation Europe. Inland Navigation Europe: Annual report 2019. 5, 2020.

[25] Bart Kersbergen, János Rudan, Ton van den Boom, and Bart De Schutter. *Towards railway traffic management using switching Max-plus-linear systems: Structure analysis and rescheduling*, volume 26. 2014.

[26] Bart Kersbergen, János Rudan, Ton van den Boom, and Bart De Schutter. Towards railway traffic management using switching Max-plus-linear systems: Structure analysis and rescheduling. *Discrete Event Dynamic Systems: Theory and Applications*, 26(2):183–223, 2016.

[27] Bart Kersbergen, Ton van den Boom, and Bart de Schutter. On implicit versus explicit max-plus modeling for the rescheduling of trains. 15, 2013.

[28] Bart Kersbergen, Ton van den Boom, and Bart de Schutter. Distributed model predictive control for railway traffic management. *Transportation Research Part C: Emerging Technologies*, 68:462–489, 2016.

[29] Shijie Li, R. R. Negenborn, and Jialun Liu. Stimulating inland waterway transport between seaports and the hinterland from a coordination perspective. In *Lecture Notes in Computer Science*, pages 67–85, 10 2018.

[30] Shijie Li, R. R. Negenborn, and Gabriel Lodewijks. Planning inland vessel operations in large seaports using a two-phase approach. *Computers and Industrial Engineering*, 106:41–57, 4 2017.

[31] Shijie Li, Rudy R. Negenborn, and Gabriel Lodewijks. Survey on planning problems in inland waterway transport: Current status and future perspectives. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (Itsc):1231–1237, 2013.

[32] Shijie Li, Rudy R Negenborn, and Gabriel Lodewijks. Approach Integrating Mixed-Integer Programming and Constraint Programming for Planning Rotations of Inland Vessels in a Large Seaport. *Transportation Research Record*, 2549(1):1–8, 2016.

[33] Wenjie Li, Jialing Dai, Yi Xiao, Shengfa Yang, and Chenpeng Song. Estimating waterway freight demand at Three Gorges ship lock on Yangtze River by backpropagation neural network modeling. *Maritime Economics and Logistics*, 2020.

[34] Gabriel A.D. Lopes, Bart Kersbergen, Ton J.J. van den Boom, Bart de Schutter, and Robert Babuska. Modeling and control of legged locomotion via switching max-plus models. *IEEE Transactions on Robotics*, 30(3):652–665, 2014.

[35] Panama Canal Authority. Proposal for the expansion of the Panama Canal. 2006.

[36] Ward Passchyn, Sofie Coene, Dirk Briskorn, Johann L. Hurink, Frits C.R. Spieksma, and Greet vanden Berghe. The lockmaster's problem. *European Journal of Operational Research*, 251(2):432–441, 6 2016.

[37] Gernot Pauli. Sustainable transport: A case study of Rhine navigation. *Natural Resources Forum*, 34(4):236–254, 11 2010.

[38] Port of Rotterdam. Port of Rotterdam Vision 2030. 2011.

[39] Port of Rotterdam. Progress Report 2014 - Port Vision 2030. 2014.

[40] Rijkswaterstaat. Herijking capaciteitanalyse Vaarwegen. 2020(November), 2010.

[41] Rijkswaterstaat. *Richtlijnen Vaarwegen 2020*. 2020.

[42] Robert Stahlbock and Stefan Voß. Operations research at container terminals: a literature update. *OR Spectrum*, 30(1):1–52, 2008.

[43] Ching-Jung Ting and Paul Schonfeld. Control Alternatives at a Waterway Lock. *Journal of Waterway Port Coastal and Ocean Engineering*, 127, 4 2001.

[44] Ching-Jung Ting and Paul Schonfeld. Efficiency versus Fairness in Priority Control: Waterway Lock Case. *Journal of Waterway Port Coastal and Ocean Engineering*, 127, 4 2001.

[45] U.S. Army Corps of Engineers. Waterborne commerce from the United States - Technical report. Technical report, 2009.

[46] Ton J.J. van den Boom and B. de Schutter. Modelling and control of discrete event systems using switching max-plus-linear systems. *Control Engineering Practice*, 14(10):1199–1211, 2006.

[47] Ton J.J. van den Boom and Bart de Schutter. MPC of implicit switching max-plus-linear discrete event systems - Timing aspects. *Proceedings - Eighth International Workshop on Discrete Event Systems, WODES 2006*, 73:457–462, 2006.

[48] Ton J.J. van den Boom and Bart de Schutter. Stabilizing model predictive controllers for randomly switching max-plus-linear systems. *2007 European Control Conference, ECC 2007*, 19:4952–4959, 2007.

[49] Ton J.J. van den Boom and Bart de Schutter. Modeling and control of switching max-plus-linear systems with random and deterministic switching. *Discrete Event Dynamic Systems: Theory and Applications*, 22(3):293–332, 2012.

[50] Ton J.J. van den Boom, Gabriel Delgado Lopes, and Bart de Schutter. A modeling framework for model predictive scheduling using switching max-plus linear models. *Proceedings of the IEEE Conference on Decision and Control*, pages 5456–5461, 2013.

[51] Ton J.J. van den Boom, Marenne van den Muijsenberg, and Bart de Schutter. Model predictive scheduling of semi-cyclic discrete-event systems using switching max-plus linear models and dynamic graphs. *Discrete Event Dynamic Systems: Theory and Applications*, 30(4):635–669, 2020.

[52] M van Koningsveld and J den Uijl. *Open source Transport Network Simulation Documentation.* Date accessed: 2022-02-15, 2020.

[53] Huibert Van Rossum. Governance Energy transition Port of Rotterdam and Industrial complex Rotterdam. 2018.

[54] J Verstichel. The lock scheduling problem. *Ph.D. dissertation, Katholieke Universiteit Leuven, Leuven, Belgium*, (November):160, 2013.

[55] J. Verstichel, P. De Causmaecker, F. C.R. Spieksma, and G. Vanden Berghe. Exact and heuristic methods for placing ships in locks. *European Journal of Operational Research*, 235(2):387–398, 2014.

[56] Jannes Verstichel and Greet Vanden Berghe. A Late Acceptance Algorithm for the Lock Scheduling Problem. pages 457–478, Heidelberg, 2009. Physica-Verlag HD.

[57] Jannes Verstichel, Patrick De Causmaecker, Frits Spieksma, and Greet Vanden Berghe. The generalized lock scheduling problem: An exact approach. *Transportation Research Part E: Logistics and Transportation Review*, 65(1):16–34, 5 2014.

[58] Jannes Verstichel, Patrick de Causmaecker, and Greet vanden Berghe. Scheduling algorithms for the lock scheduling problem. In *Procedia - Social and Behavioral Sciences*, volume 20, pages 806–815. Elsevier, 1 2011.

[59] Yihui Wang, Tao Tang, Bin Ning, Ton J.J. van den Boom, and Bart De Schutter. Passenger-demands-oriented train scheduling for an urban rail transit network. *Transportation Research Part C: Emerging Technologies*, 60:1–23, 2015.

[60] L A Wolsey. *Integer Programming*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 1998.

Glossary

List of Acronyms

DES	Discrete-Event System
IWT	Inland Waterway Transport
AIS	Automatic Identification System
V2I	Vessel-to-Infrastructure
DES	Discrete Event System
MPL	Max-Plus Linear
SMPL	Switching Max-Plus Linear
MILP	Mixed Integer Linear Programming
DES	Discrete Event System
MPL	Max-Plus Linear
SMPL	Switching Max-Plus Linear
LP	Linear Programming
MILP	Mixed Integer Linear Programming
OpenTNSim	Open source Transport Network Simulation
UDFR	Uni-Directional Fixed Routing
UDVR	Uni-Directional Variable Routing
BDFR	Bi-Directional Fixed Routing
BDVR	Bi-Directional Variable Routing

List of Symbols

L_{tot}	The total number of Max-Plus binary control variables
L	The number alternative sets of routes

	Departure location for vessel k
$\mathcal{B}_{\text{net}}$	Bottleneck in the waterway network
$\mathcal{C}_{\text{max}}$	Make-span of the waterway network
$\mathcal{D}_{\text{net}}$	Cumulative delays in the network
$\mathcal{L}_{\text{e,i}}$	Number of empty lock levellings for lock i
$\mathcal{L}_{\text{i}}$	Number of lock levellings for lock i
$\Delta_{\text{lock ,i}}(k)$	Delays at lock i for vessel k
$\mathcal{A}_{\text{net}}$	Cumulative arrival time of the network
$\mathcal{O}_{\text{lock ,i}}(t)$	Occupancy of lock i at time unit t
$\mathcal{Q}_{\text{lock ,i}}(t)$	Queue order and length of lock i at time unit t
$A_0^{\text{job}}(k))$	The routing matrix for vessel k
$A_\mu^{\text{ord}}(k-\mu))$	The ordering matrix which describes the relation between vessel k and $k\mu$
$E(n,n)$	The $n \times n$ matrix with element e on the diagonal and and ε elsewhere
J	Cost function
J_{in}	Input cost function
J_{out}	Output cost function
T	Topology matrix associated with topology graph
T_i	Topology matrix for case i
$[A]_{ij}$	Element (i,j) of matrix A
$\lambda(k)$	weight function for the input objective function for vessel k
$\mathbb{B}$	The set $\{0,1\}$
$\mathbb{B}_\varepsilon$	The set $\{e,\varepsilon\}$
$\mathcal{E}(n,m)$	The $n \times m$ matrix with all elements equal to to ε
$\odot$	Max-plus Schur product
$\oplus$	Maximization or max-plus addition
$\otimes$	Addition or max-plus multiplication
$\sigma(k)$	weight function for the output objective function for vessel k
$\sigma_a(k)$	Output weight element for departure and intermediate locations for vessel k
$\sigma_d(k)$	Output weight element for the arrival location for vessel k
$\tau_{L(i,j)}$	The constant lock operation time for moving from event i to event j
$\tau_{safety}(k)$	Safety factor time for when a vessel is coming out of the lock and vessel k wants to enter directly
$\tau_{w(i,j)}(k)$	The constant sailing time of vessel k on the waterway from node i to node j
ε	The zero element in max-plus algebra
$a_i(k)$	The time instant at which vessel k should arrive at node i
$d(k)$	Departure location for vessel k
e	The unit in max-plus algebra
k	Represents a vessel number
k_{max}	Number of vessels sailing the network
$s_l(k)$	Binary control variable that determines if vessel k is scheduled on route l

$s_l(k)$	The time Max-Plus binary routing control variable for selecting route l by vessel k
$s_{i,j}(k)$	The time Max-Plus binary routing control variable which denotes if route the arc from node i to node j is used by vessel k
$s_{ij}(k)$	Binary control variable that determines if arc (i, j) is used by vessel k
t	Represents the time unit
$u_i(k)$	The time instant at which vessel k departs at node i
$w_{DD,\mu}(k-\mu)$	Max-Plus ordering binary control variable when both vessels are sailing downstream
$w_{DU,\mu}(k-\mu)$	Max-Plus ordering binary control variable when vessel k is sailing downstream and vessel $k-\mu$ is sailing upstream
$w_{UD,\mu}(k-\mu)$	Max-Plus ordering binary control variable when vessel $k-\mu$ is sailing downstream and vessel k is sailing upstream
$w_{UU,\mu}(k-\mu)$	Max-Plus ordering binary control variable when both vessels are sailing upstream
$w_{i,\mu}(k)$	Ordering control variable that determines if the order in node i is the same as in $\pi(i)$
$x_i(k)$	The time instant event i happens for vessel k
$x_a(k))$	Optimised destination arrival time at location a for vessel k
$v(k)$	Vector containing all control variables