Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization:	Transport Engineering and Logistics
Report number:	2013.TEL.7759
Title:	**Intelligent waterways for reducing congestion at locks**
Author:	R.J.W. van Gils

Title (in Dutch)	Intelligente waterwegen voor het reduceren van opstoppingen bij sluizen

Assignment:	Computer Assignment
Confidential:	no
Initiator (university):	Dr. R. Negenborn
Initiator (company):	-
Supervisor:	Dr. R. Negenborn
Date:	April 2, 2013

Delft University of Technology

FACULTY OF MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department of Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Student:	R.J.W. van Gils	Assignment type:	Computer
Supervisor (TUD):	Dr. Rudy Negenborn	Report number:	2013.TL.7759
Supervisor (Company):	-		
Specialization:	TEL	Confidential:	no
Creditpoints (EC):	15		

Subject: **Intelligent waterways for reducing congestion at locks**

In The Netherlands transport over water is already important now, and will become more important in the future as an increasing amount of containers and bulk materials will have to be transported over water ways. Currently, vessels that transport such materials do not take into account the presence of other vessel. They travel as quickly as possible from one location to another. However, in order to most effectively (safest, fastest, most energy efficient) use the existing infrastructure in the future, individual vessels will have to take into account the presence of the other vessels and adjust their own behavior (i.e., position, heading and speed) based on the behavior of the surrounding vessels. Therefore, currently, control algorithms and cooperation protocols are being developed aimed at achieving this.
One of the particular problems that such control algorithms could be beneficial for is for reducing waiting times at locks. Long waiting times at a lock are one of the problems in inland waterway navigation. Expectations for waiting times in the next decade show waiting times that can increase up to several hours. That means vessels are waiting in a queue at a lock for several hours before the vessel can pass the lock. Since the length of the waiting time is not known on beforehand, the long waiting times can cause unexpected delays for shippers and their cargo. The unexpected delays play an important role in inland waterway transport, because inland waterway transport is often a part of a logistic chain. That means more parties and actors are involved in the transport process of cargo and these parties and actors must rely on agreements made between other parties in the chain. Unexpected delays can cause that agreements are not met, which is not good for reliability.

To minimize waiting times at locks, so that unexpected delays for vessels are avoid and that reliability for inland waterway transport increases, this research assignment aims at addressing the following research question:

What are intelligent waterways? How could intelligent waterways reduce waiting times at locks?
What model could be developed to implement the concept of intelligent waterways?
In particular, how could a waterway network be modeled in a component wise, structured way?
How could intelligent components be added to this that can monitor and control segments?
How and what experiments could be implemented using this model to assess the potential performance improvements?

Based on your research study, it is expected that you conclude with a recommendation for future research opportunities and potential for more ideas and/or applications. The report must be written in English and must comply with the guidelines of the section. Details can be found on the website.

For more information, contact Dr. Rudy Negenborn (8B-1-05; r.r.negenborn@tudelft.nl).

Summary

The size and amount of inland shipping vessels on the waterways in the Netherlands will increase in the future. This will partly be the result of the growing throughput of the Port of Rotterdam to the hinterland and vice versa. In the inland shipping waterway network locks are the most important bottlenecks, which results in waiting times at the locks. Without intervention the expectations are that the waiting times will increase to unacceptable levels. This research will focus on the implementation of the concept of intelligent waterways on the inland waterway network in order to optimize the flow of inland shipping vessels and reduce waiting times at locks. This may not lead to an increase of the voyage costs for ship owners and should result also in a reduction of the emissions. The composed requirements are transformed to a simulation model specification, which is used for the model design. From there the design is implemented into Lazarus, using the simulation package TOMAS. The model is used to perform experiments of which the results must contribute in answering the research question.

The concept of intelligent waterways is implemented by a system with knowledge. For example a waterway segment knows the position of each ship in its domain and a lock complex knows the start times of the cycles. This information is accessible to a global controller, which uses the information to optimize the flow of vessels according to a defined objective. The results of the optimization are communicated back to the ships and locks. In response ships will for example adapt their velocities and locks their cycle scheme.

Several simulation experiments have shown that intelligent waterways with a global controller which uses an optimization algorithm, contribute to the reduction of waiting times at locks. Actually the waiting times are completely eliminated in the proposed experiments while simultaneously the total voyage costs and the emissions are reduced. However, the optimization algorithm induces an increase in the total travel time of the vessels.

It can be concluded that the implementation of the proposed intelligent waterways result in an optimized flow of inland shipping vessels through a lock. In the performed simulations the results of the suggested optimization algorithms were equal, despite of the huge difference in computational speed. However, due to some assumptions, further research is desired. It should focus on the performance of the optimization algorithms when simulation parameters and assumptions are changed.

Summary (in Dutch)

De grootte en het aantal binnenvaartschepen op de Nederlandse waterwegen zal in de toekomst toenemen. Dit zal deels het gevolg zijn van de groeiende doorvoer van goederen via de haven van Rotterdam naar het achterland en vice versa. Sluizen vormen de belangrijkste knelpunten in de binnenwateren, deze resulteren in wachttijden bij de sluizen. Zonder ingrijpen zullen de wachttijden in de toekomst naar verwachting toenemen tot onaanvaardbare niveaus. Dit onderzoek zal zich richten op de implementatie van het concept van intelligente waterwegen om de doorstroming van binnenvaartschepen bij sluizen te optimaliseren, met het oog op de reductie van de wachttijden. Dit mag niet leiden tot een verhoging van de reiskosten voor scheepseigenaren en zal ook moeten leiden tot een vermindering van de uitstoot van schadelijke stoffen. Het opgestelde programma van eisen is omgezet in een specificatie voor het model. Van daaruit wordt het ontwerp gemplementeerd in Lazarus, met behulp van het simulatie pakket TOMAS. De resultaten van de met het model uitgevoerde experimenten moeten bijdragen in het beantwoorden van de hoofdvraag.

Het begrip intelligente waterwegen wordt gemplementeerd door een systeem met kennis. Een waterweg segment weet bijvoorbeeld de positie van elk schip in haar domein en een sluizencomplex weet de aanvangstijden van de schuttingen. Deze informatie is toegankelijk voor een globale controller, die de informatie gebruikt om de doorstroming van binnenvaartschepen te optimaliseren voor een vastgestelde doelstelling. De resultaten van de optimalisatie worden gecommuniceerd naar de schepen en sluizen. In reactie hierop passen de schepen bijvoorbeeld hun snelheden aan en de sluizen het schuttingsschema.

Verschillende simulatie experimenten hebben aangetoond dat intelligente waterwegen met een globale controller die een optimalisatie algoritme gebruikt, bij draagt tot de vermindering van de wachttijden bij sluizen. De wachttijden zijn zelfs volledig gelimineerd in de uitgevoerde experimenten, tegelijkertijd zijn de totale reiskosten en de uitstoot van schadelijke stoffen gereduceerd. Echter het optimalisatie algoritme veroorzaakt een toename van de totale reistijd.

Geconcludeerd kan worden dat de implementatie van de gepresenteerde intelligente waterwegen leidt tot een geoptimaliseerde doorstroming van binnenvaartschepen door sluizen. In de uitgevoerde simulaties waren de resultaten van de twee bedachte optimalisatie algoritmen gelijk, ondanks het enorme verschil in rekentijd. Vanwege een aantal gemaakte aannames is verder onderzoek gewenst. Dit moet zich richten op de prestaties van de optimalisatie algoritmen als simulatie parameters en aannames worden gewijzigd.

Contents

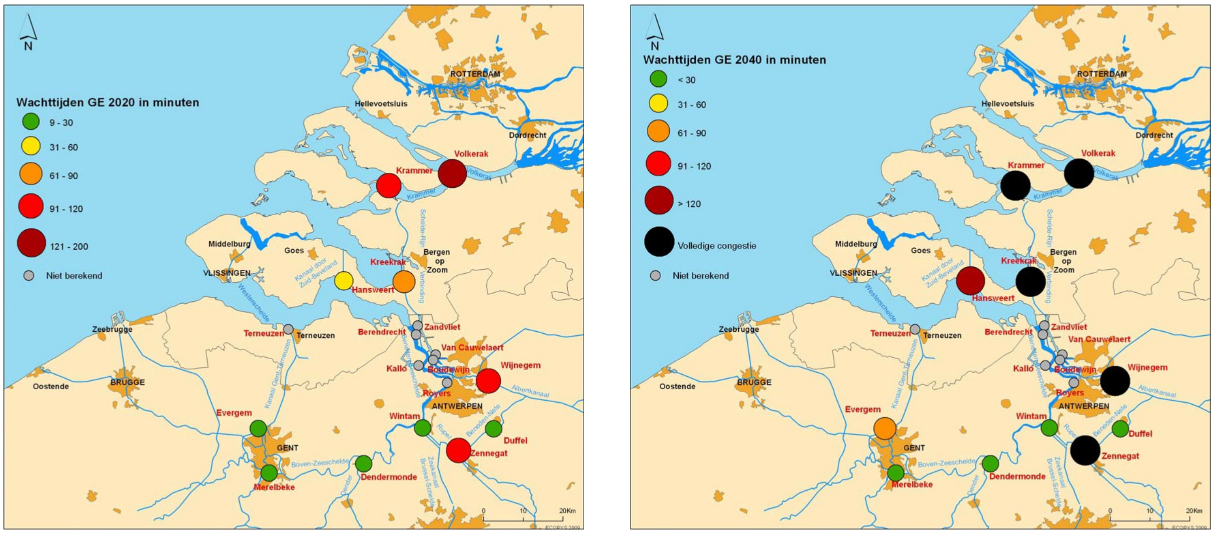
Chapter 1

Introduction

This research focusses congestions at locks which is a growing problem on the inland waterways. In Section 1.1 of this chapter the problem is illustrated with some background information. Using intelligent waterways could contribute to the solution, the corresponding research question and sub questions are discussed in Section 1.2. Finally in section 1.3 the research approach and the structure of the report is outlined.

1.1 Background

The size and amount of inland shipping vessels on the waterways in the Netherlands will increase in the future. This will partly be the result of the growing throughput of the Port of Rotterdam to the hinterland and vice versa. Locks are the most important bottlenecks in the inland shipping network, which results in waiting times at the locks (Groenveld et al., 2006). The development of waiting times at lock have been studied by Buckmann et al. (2009), the predictions for 2020 can be seen in Figure 1.1a. For the Volkerak lock complex the waiting times are expected to



(a) In 2020

(b) In 2040

Figure 1.1: Future expected waiting times at locks according to the Global Economy Scenario (Buckmann et al., 2009).

increase up to 3 hours in 2020. With no intervention there the expectation is that there will be a complete congestion at several locks in 2040, as can be seen in Figure 1.1b. The opportunities for expanding the capacity of the Volkerak lock complex have been investigated by Nieuwkamer and Rouwette (2012). Extending chamber(s) in transversal of longitudinal direction, building an extra chamber or creating a semi-open connection are the most expensive possibilities. The least expensive proposed solution is a combination of minor actions including improving maintenance management, acceleration of levelling and opening and closing time of lock gates, and introducing a traffic management system. Extending such a traffic management system for a single lock complex to a bigger or even nationwide management system could improve the waiting times at locks. Further, introducing a traffic management system may reduce costs for the shipowners. Nowadays many skippers sail as fast as possible to a lock, once arrived they often have to wait before the lock is free to enter. Reducing the vessels velocity such that they can directly enter the lock can result in a lower fuel consumption, because the vessel's engine is more economical.

Intelligence for optimizing flow has already been used in road traffic, like the 'Green Wave' concept which is much discussed in the literature, for example by Kelly (2012). Driving in formation/convoy with the use of communication between cars, as discussed by Rahman and Rideout (2012), is another way of reducing road traffic congestion. Using optimization for the reduction of waiting times at lock has been implemented by Hengeveld (2012). The explanation about the used optimization function is missing, nevertheless the results indicate that a reduction of the waiting times is possible. However, because the total travel time is increased due to decreasing velocities and the fuel consumption is assumed to be constant, the costs increase. Incorporating the fuel consumptions dependence of the velocity (Klein Woud and Stapersma, 2003), could lead to different results. Besides potential costs reduction due to a decrease in fuel consumption, this could also lead to an emissions reduction. In research these subjects will be investigated.

1.2 Research question

As discussed in the preceding section, locks in the inland waterways induce congestions which results in waiting times at the locks. In addition, these waiting times could be used to lower the vessels velocity which possibly results in less emissions and lowering the fuel costs, which have an effect on the total voyage costs. Introducing an intelligent system which tries to optimize the flow of vessels on the waterways, might reduce the waiting times at locks and emissions, this results in the main research question:

Is it possible to develop intelligent waterways for the optimization of the flow of inland shipping vessels in order to reduce waiting times at locks and lowering the emission without an increase of the voyage costs for the vessels owner?

From this main research question following sub-questions are:

- What are intelligent waterways?
- How could intelligent waterways reduce waiting times at locks?
- How will such system affect the emissions?
- How will the voyage costs be affected by such system?
- What model could be developed to implement the concept of intelligent waterways?
- How could a waterway network be modelled in a component wise, structured way?
- How could intelligent components be added to this that can monitor and control segments?
- How and what experiments could be implemented using this model to asses the potential performance improvements?

These questions can be summarized in an overall objective of this assignment, which reads:

Design and build a model to implement the concept of intelligent waterways. The model must be able to investigate the affect of intelligent waterways on the considered system. The waterway network must be constructed with adjustable standard components (building blocks) and must be able to approximate the real world.

This report focusses on these questions and objective, the used approach is discussed in the next section.

1.3 Research approach

Using the listed questions and background information discussed in Section 1.1 a specification is made. In Chapter 2 all main requirements of the model are discussed. Including model properties, required output parameters, desired variably input parameters and which kind of experiments need to be performed. With these requirements a choice for an appropriate modelling tool is made. The implementation of the designed model in a computer model is discussed in Chapter 3. In Chapter 4 the settings and results of the performed experiments will be discussed. This report concludes with Chapter 5 in which the conclusion is discussed and a recommendation is made.

Chapter 2

Model specification

In order to implement the concept of intelligent waterways, a model has to be designed. Using the research questions and background information proposed in Chapter 1 a specification of the model is made in this chapter. The main requirements are discussed and explained in Section 2.1. These requirements should be processed into the design of the model, this is the subject of Section 2.2. Finally, having a model designed on paper a choice can be made regarding the simulation type and software, which is discussed in Section 2.3.

2.1 Main requirements

A model must be designed which can be used to answer the main research question and the related sub-questions. The main goal of the model is implementing the concept of intelligent waterways and used it to investigate the effect of this concept. Therefore, a part of the inland waterway network must be modelled for this investigation. Components of a real inland waterway network, like waterways, locks and bridges, must be modelled. If the dimensions of these components are adjustable and can be connected to each other, various parts of the inland waterway network can be constructed. To complete the waterway network, vessel of different sizes must be modelled. These vessels must be able to move through the waterway network. To perform tasks, some components have intelligence. For example, a task of a lock is to transfer ships from one side to the other side of the lock and the task of a ship is sailing through the network. This intelligence has to be extended in order to reduce the waiting times at locks by optimizing the flow of the ships through the network. The effect of the concept of intelligent waterways can be seen when simulations are performed with and without this concept. Therefore measurements like waiting times at locks and voyage costs of the vessels must be monitored for comparison. The above discussed main requirements are listed below:

- The model should assist in answering the research questions.
- The model must be able to represent a piece of the real inland waterway network.
- The waterway network must be constructed in a structured and component wise way.
- Components of the waterway network must have some kind of intelligence.
- The concept of intelligent waterways must be implemented in the model.
- The model must be able to run with and without the concept of intelligent waterways.
- The behaviour of the system must be monitored.

2.2 Model design

The translation of the requirements into a model description is discussed in this section. The model description is divided into a number of subsections, in which the structure and the behaviour of the model is explained. The first to be discussed is the model representation of the inland waterway network, followed by the vessels which virtually sail through the waterway network. Vessels and some of the components of the waterway network that can perform tasks, these tasks are controlled by a controller. These controllers have some kind of intelligence for making decisions on how tasks must be performed. How these controllers are implemented and what make them intelligent, is explained in the intelligent control part. Finally the required input and output will be discussed for performing comparable experiments with the model.

2.2.1 The inland waterway network

For the creation of the inland waterway network segments are used, these segments represent a part of the network and are connected to each other by nodes. In Brolsma and Roelse (2011) the distinction is made between waterways (rivers or canals), locks, bridges and ports as being parts of the waterway network. The relevant waterway network segments for this investigation follows from the research question, these are waterways and locks. The considered system is a lock with on one side (the side from which the vessels approaching the lock) a waterway, which can be built up out of several waterway segments. Ships will enter the system at the beginning of the waterway and leave the system on the other side of the lock, as is depicted in figure 2.1. For

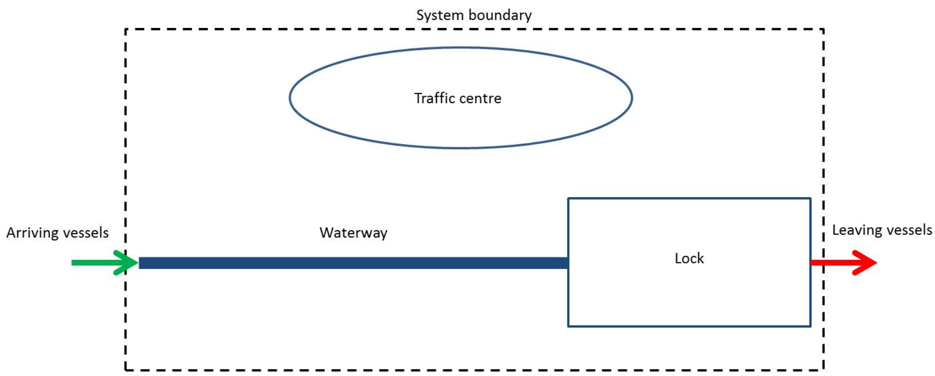


Figure 2.1

simplicity the ships will only approach the lock at one side, hence the waterways on the other side of the lock will not influence the behaviour of the system and therefore the ships will leave the system when they have passed the lock. Bridges could also be a part of the considered waterway network, they should be taken into consideration when they can influence the behaviour of the system. For example when a ships has to wait for a bridge to open before he can pass the bridge, his arrival at the lock will be delayed. Ships begin and end their voyages in a port, the start time of a voyage depends on when a ship is finished with loading or unloading. Ports are not part of the considered system, instead, ships will enter the system in a predetermined arrival pattern. Traffic centres are present in ports and busy parts of the waterway network. They supervise a part of the waterway network and give support to skippers during navigation by providing information about other ships, obstructions, berths, etc. (Groenveld et al., 2006). In this research the traffic centre will be used to implement the global controller of the system, the optimization calculations will be performed by the traffic centre and it will control the other actors in the considered system.

2.2.2 Ships

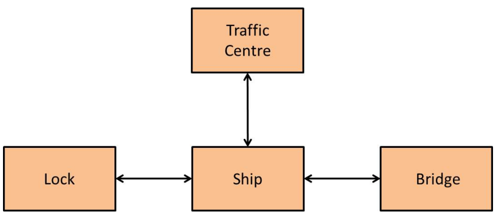
With only a model representation of the waterway network, the research questions cannot be answered. Therefore vessels need to sail (virtually) through the network. In order to regulate the flow of vessel, their velocity must be adjustable. In an attempt to find the optimum velocities for all vessels, regarding the total costs for the ship owners, more information has to be known. The total costs for a ship owner, can be divided in two main parts: Fixed costs and variable costs. The fixed costs include depreciation, interest, insurance, maintenance and labour costs. Fuel costs are the variable costs. The fuel consumption, which is directly related to the fuel costs by the fuel price, depends on the engine output. The required engine output depends again on the required velocity of the vessel. Together with the travelling distance, the total fuel costs for a trip can be calculated. Emissions are directly related to the fuel consumption (Gon and Hulskotte, 2010), a reduction in the fuel consumption results in a reduction of the emissions.

2.2.3 Intelligent control

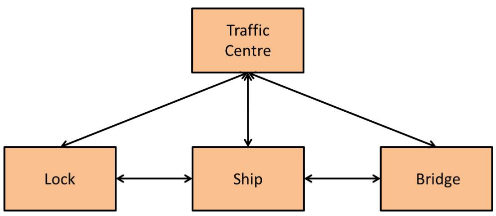
As in reality, some of the previous discussed objects are able to perform tasks. These tasks may differ per object type and can be controlled by either humans, computers or a combination of these two. These controllers are able to control a process because they have a form of intelligence. Using information from their surroundings, decisions can be made regarding how to respond on a certain occurrence and decide what actions should be taken. The actors with intelligent control in the model are:

- Vessels: Sailing through the waterway network. The skipper is the intelligent controller using a board computer system
- Lock: Transferring vessels from one side of the lock to the other side. The lock master is the intelligent controller using a lock computer system.
- Bridge : (in case of a movable bridge) Opening and closing the bridge passage: The bridge master is the intelligent controller using a bridge computer system.
- Traffic control: Supervising the waterway network. Traffic control employee.

As said before, an intelligent controller uses information to make decisions on how to perform tasks. Communication is an information stream which is used in the inland waterway network. Nowadays the radio is the way communication between vessels and locks, bridges, traffic centres and other vessels, takes place. In figure 2.2 the communication scheme is depicted for the situations with and without a global controller. The situation without a global controller



(a) Current situation



(b) New situation

Figure 2.2: Communication schemes of the model.

corresponds with the current situation, the new situation will be with the implementation of a global controller. The difference between these situations is that in the current situation the traffic centre only has a supervising function and only communicates with vessels. In the new situation the traffic centre will also communicate with the locks and bridges. Other information streams which are used in the decision processes of the intelligent controllers, are radar and the electronic waterway chart in combination with the AIS. The latter results in a digital waterway chart on which all ships with AIS are depicted, information about these ships like ship size, cargo, velocity and destination can be obtained.

In the new situation, the traffic centre will have the function of the global controller and its main task will be to optimize the flow of the vessels, according to some objective function. The objective will be to minimize the total costs, which consists (as discussed earlier) of fixed costs and fuel costs. The implementation of the objective function and the corresponding constraints is discussed in Section 3.3.

2.2.4 Model input and output

As discussed in section 1.2 the goal of the model is to investigate the affect of intelligent waterways on the considered system. Therefore the results of several experiments, with and without global controller, must be compared. For the construction of these experiments, some of the input parameters needs to be varied, which is discussed first. Thereafter the output parameters for results comparison will be discussed.

Model input

The influence of the global controller on the simulated system can be investigated by running several experiments with different settings. For the construction of these experiments, the following input parameters will be adjustable:

- The size of the waterway network in the simulation model.
- The vessel inter-arrival distribution (to simulate future growth).
- The objective function and/or algorithm.

The size of the waterway network will influence the optimization results. Considering a longer waterway ahead of a lock, will increase the number of possible solutions. As discussed in Section 1.1, the expected waiting times at locks will increase in the future due to an increase of the number of vessels. In order to be able to also simulate this scenario the inter-arrival time distribution needs to be adaptable. If more than one optimization algorithm is available, a comparison can be made by implementing them all.

Model output

To measure the eventual improvements, reached with the controller, the results of a simulation with the current situation and one with the new situation, need to be compared. The most important measurements, which follow from the research question in Section 1.2, are listed down here and need to be the output of the model.

- Mean and total costs
- Mean and maximum waiting times at locks
- Waiting times distribution
- Mean and maximum time in system (this depends on the (mean) velocity of the vessel)

Comparing these results for different experiments will give a good impression of the workability of the optimization algorithm.

2.3 Modelling tool

In the preceding sections the model specification is made, before it can be implemented a choice has to be made considering the simulation type, which will be discussed in the next section. In the last section of this chapter the choice of the simulation software package will be outlined.

2.3.1 Simulation type

Two types of simulation can be distinguished, continuous and discrete simulation. In a continuous system the state variable can change continuously over time, in a discrete system state variables change only at a discrete set of points (Banks and Carson, 1984). Continuous systems are described by differential equations. The process described prior, the waterway network with vessels sailing through it, cannot completely be described by differential equations, therefore the used type of simulation is discrete simulation. As discussed by Evans (1988), in a discrete simulation one can choose between discrete-time and continuous time systems. In discrete-time systems the value of time is being incremented by a constant amount, which results in a problem when an event takes place between two successive points in time. In continuous-time systems, the magnitude of the time increment is not fixed, but variable, so this problem will not occur. The simulation algorithm can either search each time step for events or can be event driven. The disadvantage of the former is that at every time step all entities which can have an event must be checked, which is inefficient if the number of events on a time step in much smaller than the number of entities in the system. A discrete, continuous-time simulation which concentrates on processing events is called a discrete event simulation (Zeigler et al., 2000).

2.3.2 Software

The used software package must support discrete event simulation. The package TOMAS (Veeke and Ottjes, 2000) is a discrete event simulation software package for Delphi/Lazarus, developed by employees of the section Maritime and Transport Technology of the Technical University of Delft. The Matlab toolbox SimEvents also supports discrete event simulation. The disadvantage is that the entities which flow through a network constructed with SimEvents cannot perform tasks by themselves (MathWorks Inc., 2012). Vessels will be the entities which flow through the waterway network. As discussed earlier in this chapter, a vessel has its own controller and is able to perform the task of sailing. Therefore the Matlab toolbox SimEvents is not suitable for this purpose and the package TOMAS for Delphi/Lazarus will be used.

SIVAK, written in Prosim is a inland shipping simulation tool, which is developed commissioned by Rijkswaterstaat (Rijkswaterstaat, 1981). During this research the possibilities of implementing the concept of intelligent waterways in SIVAK have been investigated. Due to the absence of the source code of SIVAK this investigation was quickly finished.

Chapter 3

Implementation

The comprehensive model specification discussed in the preceding chapter will be turned into an implementable model in this chapter. In section 3.1 the defined classes and their interrelationship is presented, together with the attributes and methods of the classes. The process description of classes which own a process are described in section 3.2. The implemented optimization is explained in section 3.3. Finally, in section 3.4, the structure of the model is exemplified.

3.1 Class descriptions

From the specifications discussed in Chapter 2 classes are defined. The important characteristics of objects in object-oriented programming (OOP), encapsulation, inheritance and polymorphism, are used during the design of the classes, as discussed by Schildt (2011), Saleh (2009) and Raphael and Smith (2003). The methodology of OOP is designed to make data structures related close to reality (Kerman, 2004), therefore the designed classes are chosen to be close to the reality. For

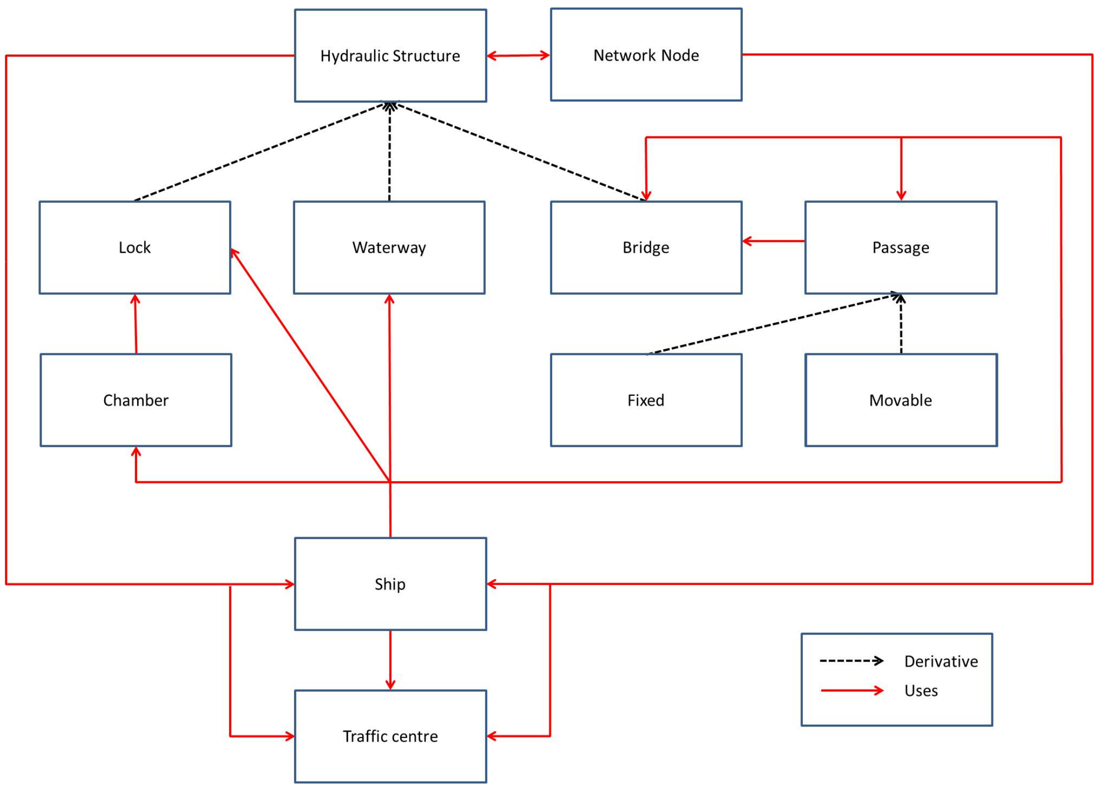


Figure 3.1: A class diagram of the model

example, a ship is an instance of the ship class, with properties (also called attributes) like length and draught. Some of the classes and attributes are more abstract, these are mostly implemented for programming purposes. The ship generator is an example of an abstract class, it does not exist in reality, but its needed in the program to create ships during the simulation. An overview of the interrelated classes is depicted in figure 3.1. In the next sections all relevant classes, in order to understand the simulation model, are explained. At first is the TomasElement class discussed, this class is predefined in the TOMAS package and all classes described in this section are descendants of this class.

3.1.1 TOMAS Element

Because all the classes defined in the model are descendants of this class, they own the attributes and methods of this class. Veeke and Ottjes (2000) describes the TOMAS package, in the user manual of TOMAS (Veeke, 2000) all attributes and methods are extensively described. The most important ones are listed below. The TOMAS Element class it self is a descendant of a standard class in Lazarus, TPersistent, more information about lazarus can be found at *http://wiki.freepascal.org*.

TomasElement(TPersistent)		
Attribute	**Type**	**Description**
Name	String	Name of object
Method	**Type**	**Description**
Create	Constructor	Creates an instance
Destroy	Destructor	Destroys an instance
Process	Procedure	A virtual procedure, must be overwritten
Start	Procedure	Starts the process
Stop	Procedure	Stops the process
Resume	Procedure	Resumes the process

3.1.2 Hydraulic Structure

For the creation of a waterway network, components of the inland waterway are identified. These components are waterways, locks and bridges. By connecting instances of these components (called segments), each with different attributes, each part of the inland waterway network can be modelled. Ports (not depicted in Figure 3.1) could be a fourth component, but these can also be seen as a special type of waterways. In this research, bridges and ports are left out of consideration. This abstract class is the parent class for the waterway and lock classes. All segments in the waterway network are descendants of this class.

THydraulicStructure(TomasElement)		
Attribute	**Type**	**Description**
ID	Integer	Unique number
Nodes	Array of TNetworkNode	The nodes of the hydraulic structure
Method	**Type**	**Description**

3.1.3 Network Node

The waterway network is constructed with segments as discussed in preceding section, the connection between segments is made by instances of this abstract class; nodes. A node is connected with at least one object of the class hydraulic structure (or a descendant of this class). A waterway network segment (an instance of the hydraulic structure class is connected with two nodes).

TNetworkNode(TomasElement)		
Attribute	**Type**	**Description**
ID	Integer	Unique number
NofConnections	Integer	Amount of hydraulic structures connected to the node
Connections	Array of THydraulicStructure	All connected hydraulic structures
Method	**Type**	**Description**

3.1.4 Waterway

A waterway represents a piece of the waterway network. The simulation is a simplification of the reality, therefore parameters like width and depth are omitted from this consideration but could be appended when necessary.

TWaterway(THydraulicStructure)		
Attribute	**Type**	**Description**
Length	Double	Length of the waterway
Ships	Array of TShip	Ships present in the waterway
Method	**Type**	**Description**

3.1.5 Lock

A lock is simulated as a waterway network segment with zero length and which is able to consume time in the simulation while holding ships. It can be compared with a server in a queue-server system. In the waiting queue in front of the lock, ships are waiting until they are served by the lock. Serving ships takes time.

TLock(THydraulicStructure)		
Attribute	**Type**	**Description**
NofChambers	integer	Number of chambers
Chambers	Array of TChamber	Chambers which belong to the lock
Waiting ships	Array of TShip	Unassigned ships
All ships	Array of TShip	All ships present in the lock
Method	**Type**	**Description**

The capacity of the cycle is in reality dependent on the sizes of the ships, normally the size of ships is varying so the capacity cannot be expressed in a single number. In the simulation the capacity is assumed to be constant, which implies that all ships have the same dimensions.

TChamber(TomasElement)		
Attribute	**Type**	**Description**
Capacity	Integer	Maximum number of ships
Ships	Array of TShip	All ships in the chamber
Cycles	Array of TCycle	All scheduled cycles
Method	**Type**	**Description**

A chamber performs cycles to serve ships from one side of the lock to the other side. The maximum number of ships in a cycle is determined by the chamber capacity.

TCycle(TomasElement)		
Attribute	**Type**	**Description**
Start Time	integer	Time at which the cycle begins
Duration	integer	The duration of the cycle
Chamber	TChamber	The chamber used in this cycle
Ships	Array of TShip	Ships assigned to this cycle
Method	**Type**	**Description**

3.1.6 Ship

In this research main particular like length and width of a ship are not considered, therefore they cannot be found in the class description.

TShip(TomasElement)		
Attribute	**Type**	**Description**
ID	Integer	Unique identification number
Velocity	TVelocity	Min, max and current velocity
Engine	TEngine	The installed propulsion engine
Current Hydraulic Structure	THydraulicStructure	Segment location
Route	TRoute	The voyage route
Voyage log	Array of TPosition	Information about the trajectory
Statistics	TStatistics	Voyage statistics
Method	**Type**	**Description**
Process	Procedure	See process description
Update Position	Procedure	Update ship's position

A ship uses the following classes and records. A record is a data structure used to assemble related data, it does not have any methods.

TVelocity(Record)		
Field	**Type**	**Description**
Min	Double	The minimum velocity
Max	Double	The maximum velocity
Current	Double	The current velocity

The record TEngine contains only one field, c, which is the constant of fuel consumption. The relation between fuel costs per time unit and the velocity of the ship V is approximated by the function:

$$\text{Fuel costs per time unit} = c \cdot V^3 \tag{3.1}$$

This relation is described by Klein Woud and Stapersma (2003). The route is saved in the record TRoute which contains two fields, an array of TNodes and an array of THydraulicStructure. These are both sorted in order of encounter. The attribute costs in the class description above is a record with fields fixed and fuel, to make a distinction between these two parts of the total costs. The fixed and fuel costs made by a ship are recorded in these fields. In the record TStatistics all relevant information of a ship in the simulation is recorded as can be seen below.

TStatistics(Record)		
Field	**Type**	**Description**
Time In	Double	The time a ship entered the system
Time Out	Double	The time a ship leaved the system
Time Sail	Double	Time of sailing to the lock
Time Wait	Double	Time of waiting at the lock
Time Lock	Double	Time of passing the lock
Costs Fixed	Double	Total fixed costs
Costs Fuel	Double	Total fuel costs
Max	Double	The maximum velocity

3.1.7 Ship generator

During the simulation an instance of this class produces instances of the ship class. A ship generator is connected to a node and can only produce ships which depart from that node and follow the same (predefined) route. The inter-arrival time is dependent on the moment of the day and is determined with a sample from a distribution. These distributions are included in the TOMAS package. The entrance velocity of the ship is also sampled from a distribution.

TShipGenerator(TomasElement)		
Attribute	**Type**	**Description**
ID	Integer	Unique identification number
Node	TNetworkNode	Ships will enter in this node
Route	TRoute	The route of the ships
Arrival distributions	Array of TDistribution	All inter-arrival time distributions
Velocity distribution	TDistribution	The entrance velocity of the ship
ETA	Double	Estimated Time of Arrival at lock
Velocity	Double	The current velocity
Costs	TCosts	Total costs until now
Method	**Type**	**Description**
Process	Procedure	Creation of vessels

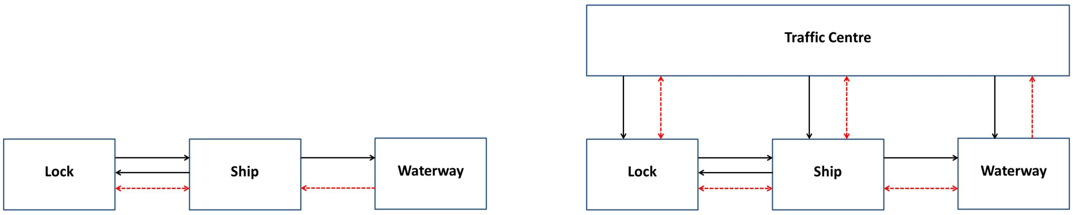
3.1.8 Traffic control

An instance of this class is used to apply the designed controller, which is implemented in the process of this class. It is a traffic centre without the supervising function. When a simulation is made without a global controller, the process of this class will do nothing.

TTraficControl(TomasElement)		
Attribute	**Type**	**Description**
Including ships	Array of TShip	All ships included in the optimization
Including cycles	Array of TCycle	All cycles included in the optimization
Method	**Type**	**Description**
Process	Procedure	Control of the waterway network

3.2 Process descriptions

The control part of the classes, as discussed in Section 2.2.3, is implemented in the process, before these are described, the interaction between the objects (instances of a class) will be clarified. Two situations can be distinguished, the globally uncontrolled situation and the globally controlled. In the globally uncontrolled situation, as it is nowadays, the traffic centre has only a supervising function, as discussed in section 2.2.1. This supervising function is not taken into account in the model, therefore in the globally uncontrolled situation the traffic centre is not present in the model. The interactions between objects in the two situations are depicted in Figure 3.2. The black arrows indicate process interaction, the red (and dotted) arrows indicate



(a) Uncontrolled situation

(b) Controlled situation

Figure 3.2: Interaction between objects, the black arrows indicate process interaction, red (dotted) arrows indicate information flow, the direction is indicated by the arrows.

information flow. To illustrate, in the globally uncontrolled situation (Figure 3.2a) a ship retrieves information from waterways and locks, sends information to locks and uses waterways and locks in its process. In the globally controlled situation (Figure 3.2b), the intelligent waterway can be distinguished, because waterways also retrieves information from ships, so a waterway segment knows which ships are present. All the processes described in the remaining part of this section will be first briefly clarified in text, subsequently the way the processes are implemented in the model is explained in words.

Ship

The process of a ship is repeating until it arrives at its destination. When then a ship is on a waterway, it sails to the end of the waterway. If a ship approaches a lock, it will notified him self and will sail to the lock. At the lock it will wait until its allowed to enter the lock. In the lock the ship will wait until its allowed to leave te lock, after leaving the lock the ship will continuous its journey.

Process of TShip

Repeat the following actions
    Define the last (known) node
    Define the next node to sail to
    If the next node is the last node of the route
        Update ships position
        Write all statistics to log file
    If the last node is a lock notification node
        The ship notifies at lock
    Define the current hydraulic structure

    Case current hydraulic structure is a waterway
        Calculate sailing time from current position to next node
        Wait sailing time

    Case current hydraulic structure is a lock
        Enter the waiting queue of the lock
        Suspend the process
        Sail into the chamber to which the ships is assigned
        Suspend the process
        Sail out the chamber
        Update ships position

    Remove next node from node route
    Remove current hydraulic structure from segment route
    Last node = next node
    Start process ship from the beginning

Besides the process, a ship has also the *update position* method. This method can be called any time and will determine the position of the ship, expressed in current hydraulic structure (waterway or lock) and its position along the hydraulic structure. In case of a lock, the latter is currently 0, because the length of a lock is currently neglected. The length of a chamber is also not defined, only the capacity in number of ships.

Lock

A lock can contain more than one chambers, each chamber transfers ships from one side of the lock to the other side. Therefore not the lock owns the process but the chamber. The process of a chamber depends on the type of simulation, whether or not it is globally controlled. In both cases it will allow ships to enter and leave the chamber. As is illustrated in Figure 2.1, the ships will only approach the lock from one side, consequently the chamber is empty when it transfers back.

Process of TChamber

Repeat the following actions
Determine the simulation method

Case simulation method is globally uncontrolled
    Wait while the waiting queue of the lock is empty
    For all ships in the waiting queue repeat
        Enter the ships queue of the chamber
        Resume ships process
    Wait cycle duration

Case simulation method is globally controlled
    Determine the next cycle
    Wait until the next cycle begins
    For all ships in the lock waiting queue and which are assigned to the next cycle
        Enter the chamber ships queue
        Resume ships process
    Wait cycle duration
    For all ships in the chamber
        Exit the chamber ships queue
        Resume ships process

Ship generator

The ship generator is responsible for creating ships with a certain inter-arrival time and let them enter the waterway network. The process is repeating until the end of the simulation. It takes a sample of the inter-arrival time distribution, holds that amount of time and creates a ship. If the simulation method is globally controlled, it will trigger the traffic control to start its process.

Process of TShipGenerator

Repeat the following actions
    Create a new ship
    Give the ship a route
    Update ships position
    Case simulation method is globally controlled
        Determine the distance to the lock
        If needed, create extra cycle(s)
        Resume process of the traffic control
    Determine the inter-arrival time
    Wait inter-arrival time
Stop simulation when desired

Traffic control

The optimization algorithm is performed in the process of this class. It collects all the information for the optimization (ships and chamber cycles) and runs the optimization. The explanation of the optimization will be discussed in the next section.

Process of TTraficControl

Repeat the following actions
    Determine all ships which needs to be taken into consideration
    Determine all cycles which needs to be taken into consideration
    Set optimization variables
    Run optimization
    For all considered ships
        Set velocity to calculated value
        Update ships position
    For all considered cycles
        Add all assigned ships to cycle

    Suspend process

3.3 Optimization

Two different optimization algorithms are used. The first is a non-linear mixed integer programming (NLMIP) algorithm, which will be discussed first. Thereafter the second algorithm, a heuristic one, will be discussed.

3.3.1 Non-linear mixed integer programming algorithm

As discussed in Section 1.2 the goal of this research is to optimize the flow of inland shipping vessels in order to reduce waiting times at locks and lowering the emission without an increase of the voyage costs for the vessels owner. In Section 2.2.2 the costs are divided in two parts, fixed cost and variable cost, both are expressed per time unit [$€/s$]. For ship i the costs to pass network segment j can be written as:

$$C_i = \left(C_i^f + C_i^v\right) \cdot t_i \tag{3.2}$$

in which $C_i[]$ are the costs for ship i passing a network segment, $t_i[s]$ is the time the ship i needs to pass the segment, $C_i^f[€/s]$ are the fixed costs of ship i per time unit and $C_i^v[€/s]$ are the variable costs per time unit. The variables t_i and C_i^v are both dependent on the velocity of the ship $V_i^s[m/s]$, t_i can be calculated by:

$$t_i = \frac{d_i}{V_i^s} \tag{3.3}$$

in which $d_i[m]$ represents the distance from the lock of ship i. The fuel consumption is proportional with the velocity of the ship to the power 3, as discussed by Klein Woud and Stapersma (2003). Using the fuel price ([$€/l$]), one can write:

$$C_i^v \propto V_i^{s3} \tag{3.4}$$

Or,

$$C_i^v = c \cdot V_i^{s3} \tag{3.5}$$

in which c is a non dimensionless constant which is dependent on the ships resistance and engine. This constant can be estimated as is discussed by Holtrop and Mennen (1982) or can be deduced from the vessels engine data and velocity (Klein Woud and Stapersma, 2003). As is depicted in figure 2.1 the considered system consist of a waterway and a lock. Vessels will enter the system and sail into the direction of the lock. Once arrived at the lock they will wait until the next lock cycle starts, if they are allowed to enter the lock, the ships will be transferred to the other side of the lock. A lock cycle j starts at time $t_j^0[s]$. The objective of minimizing the costs for the ship owners can now be written as follows:

$$\text{Minimize} \quad Z = \sum_{i=1}^{n} \left(C_i^f + C_i^v\right) \cdot \frac{d_i}{V_i^s} + \sum_{i=1}^{n}\sum_{j=1}^{m} \left(t_j^0 - \frac{d_i}{V_i^s}\right) \cdot y_{ij} \cdot C_i^f \tag{3.6}$$

The second term in brackets represents the expected waiting time if ship i joins cycle j, its the difference in time between arrival at the lock and start of the lock cycle. This is multiplied by the fixed costs to get the waiting costs. The variable $y_{ij}[-]$ is a binary variable, if its value is 1, ship i will join lock cycle j and if its zero, it will not. The variables n and m represent the number of ships and cycles respectively. The variables in the objective function are subjected to the following constrains:

$$V_i^{s,min} \leqslant V_i^s \leqslant V_i^{s,max} \qquad \forall i \tag{3.7}$$

The minimum and maximum velocity of a ship is restricted.

$$\frac{d_i}{V_i^s} \cdot y_{ij} \leqslant t_{0,j} \qquad \forall i, j \tag{3.8}$$

The arrival time of the ship at the lock has to be before the start of the joining cycle.

$$\sum_{i=1}^{n} y_{ij} \leqslant q_j \qquad \forall j \tag{3.9}$$

The number of ships joining a cycle must be smaller than the chamber capacity q_j.

$$\sum_{j=1}^{m} y_{ij} = 1 \qquad \forall i \tag{3.10}$$

A ship can only join one cycle.

$$y_{ij} = 0, 1 \qquad \forall i, j \tag{3.11}$$

y_{ij} is a binary variable, non negative constraints for all V_i^s are included in equation 3.7 if $0 \leqslant V_i^{s,min}$. The above discussed problem is a non-linear mixed integer programming problem, in

which the integer variables, binary ones are. Matlab has an in build function *fmincon* which is able to solve non-linear programming problems. This function can not handle integer or binary variables, therefore a branch & bound algorithm is written in Matlab, as described by Hillier and Lieberman (2010). This algorithm is described with reference to Figure 3.3. Consider a

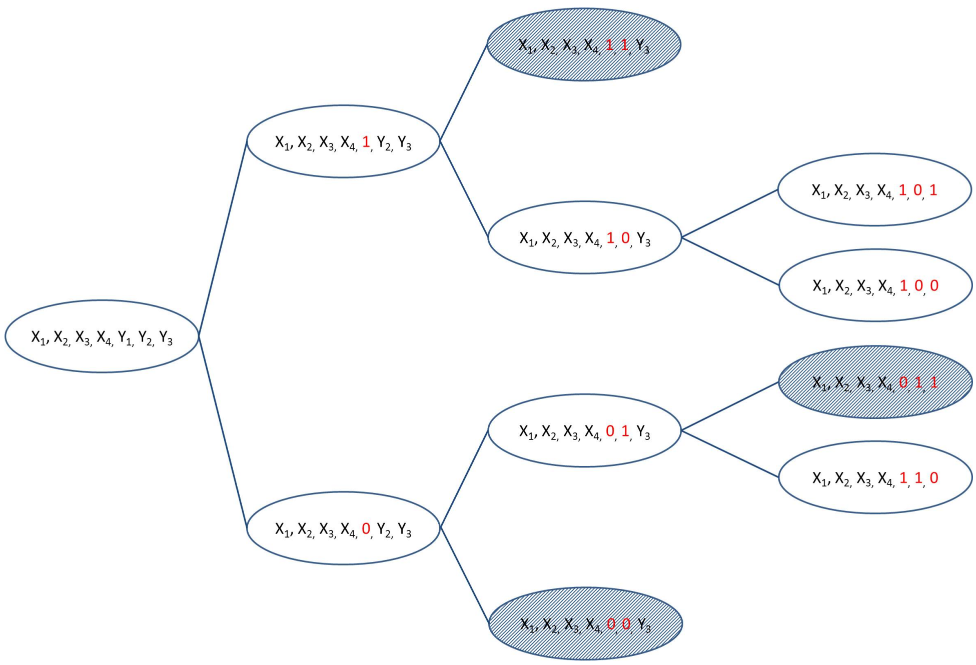


Figure 3.3: The branch and bound algorithm for a problem with 4 non-integer variables and 3 binary ones.

NLMIP problem with 4 non-integer variables (these are allowed to be integer, but not restricted) indicated by X_i ($i = 1..4$) and 3 binary variables (these are restricted to be either 0 or 1) indicated by Y_j ($j = 1..3$) (This is a general example, the optimization problem discussed above is not considered). At every branch the next steps are performed:

- Solve the reduced problem as if its is an non-linear programming problem, by using the *fmincon* function in Matlab.
- If the solution does not exist, stop the current branch.
- If the solution is worse than the best currently known solution, stop the current branch.

- If the solution is better than the best currently known solution and all binary variables are binary, stop the current branch.
- If the solution is better than the best currently know solution and not all binary variables are binary, create a new branch.

When a new branch is made, the problem is reduced by bounding one of the binary variables, either to zero or to one. In the first branch, the *reduced problem* is the original problem and the best currently known solution is infinitely large (because it is a minimization problem). The shaded branches in Figure 3.3 represent stopped branches. In Section 3.4 the communication between the model (written in Lazarus) and Matlab is discussed.

3.3.2 Heuristic algorithm

The second implemented algorithm is the heuristic algorithm. The algorithm searches for each ship which enters the system the optimal velocity, the velocities of other ships will not be adapted in this algorithm. Therefore the discussed process of the traffic control in Section 3.2 can be replaced by the following process.

Process of TTraficControl (Heuristic algoritm)

```
Repeat the following actions
        Calculate the ships ETA to the lock with maximum speed
        Determine the first possible cycle the ship can join
        Calculate the corresponding velocity in order to have zero waiting time
        Calculate the costs of joining this cycle
        Set this solution to be the best one
        Determine the next possible cycle the ship can join
        Calculate the corresponding velocity in order to have zero waiting time
        While the corresponding velocity is bigger than the ships minimum do
                Calculate the costs of joining this cycle
                If this solution is better than the currently best one
                   Set this solution to be the best one
                Determine the next possible cycle the ship can join
                Calculate the corresponding velocity in order to have zero waiting time
        Set velocity to the corresponding velocity of the best solution
        Add the ship to the corresponding cycle of the best solution

        Suspend process
```

3.4 Program structure

The simulation is written with Lazarus because the used simulation package TOMAS is available for Lazarus. Also the graphical user interface is created in Lazarus. For the non-linear mixed integer programming algorithm is Matlab used. During runtime the Lazarus application establishes a connection with Matlab. First the variables are loaded into the workspace of Matlab, then Lazarus let Matlab run Matlab script files in which the optimization algorithm is implemented. Finally the results are written into text files, which can be read by Lazarus. To run the model, Matlab is required. Instructions on how to use the model, are written in the read me file, included with the application.

Chapter 4

Simulation Experiments

This chapter contains the results of the simulation runs. In Section 4.1 the settings for the experiments are discussed. The results of experiments with varying setting are illustrated in Section 4.2.

4.1 Simulation Settings

4.1.1 System

The Volkerak lock complex is used as model for the simulation. This lock complex has three chambers, the dimensions of two of these are 308.9×24.1 $[m]$ and one is 331.5×24.1 $[m]$ (Nieuwkamer and Rouwette, 2012). As discussed in Section 2.2 the ships will enter the lock at one side. Therefore, only the waterway on that side need to be modelled. The length of the waterway is varied to see the influence, used lengths are 20 and 30 $[km]$. In the simulation only one chamber of the Volkerak lock complex will be considered.

4.1.2 Control

As discussed in preceding chapters, two situations are distinguished. The globally uncontrolled situation represents the current situation, the globally controlled situation is the situation as it would be in the future. Both situations are split into two sub-situations. The globally uncontrolled situation is not completely uncontrolled because the skipper of each ship has a certain intelligence. The skipper sets the velocity of the ship, often (as experienced during spending time on an inland vessel) the velocity is not the maximum velocity. The skipper knows, either by experience or supported by a board computer, that the fuel consumption at maximum velocity (maximum engine power) is much higher than at a velocity of around 70-80% of the maximum velocity (see next section). To see the influence of the decision of a skipper to sail not at maximum velocity the globally uncontrolled situation is split into a sub-situation in which all ships sail at maximum velocity and a sub-situation in which ships sail on average at 80% of their maximum velocity. For this purpose, use is made of a normal distribution. The globally controlled situation is spit into a sub-situation with the non-linear mixed integer programming algorithm and a sub-situation with the heuristic algorithm. For clarity, the resulting 4 situations are listed below, between the brackets the name is stated as used in the simulation results.

- Globally uncontrolled, all ships at maximum velocity (No control)
- Globally uncontrolled, ships at 80% of the maximum velocity on average(Skipper control)
- Globally controlled, non-linear mixed integer programming (NLMIP) algorithm (NLP control)
- Globally controlled, heuristic algorithm (Heuristic control)

4.1.3 Ships

The average loading capacity of ships passing the Volkerak lock complex is around 2000 ton (Rijkswaterstaat, 2011). This corresponds to the CEMT class Va (a European classification for inland shipping vessels) or to the AVV class M8 (a Dutch classification for inland shipping vessels which is an expansion of the CEMT class), both classes can be found in Brolsma and Roelse (2011). The main dimensions of such a ship are 110×11.4 $[m]$, in the simulation all ships will have these dimensions. In Hove (2010) can be found that a ship with these dimensions and a loading capacity of around 2000 ton, has an engine with a maximum output of 1100 to 1200 $[kW]$. The maximum velocity of this kind of ship in loaded condition is 15 $[km/h]$ (Hengeveld, 2012), in the simulation all ships are assumed to be loaded and thus will all have the same maximum velocity. The fixed costs are assumed to be €250 which corresponds with the work of Hengeveld (2012).

Arrival pattern

Sickinghe (2009) gives the maximum number of ships passing the Volkerak lock complex in south direction on a day, which is 180. Because only one of the three chambers is considered in the simulation, 60 ships will pass the lock in 24 hours. In Borst et al. (2012) it can be seen that during a day, the most ships pass the Volkerak lock complex between 10:00 and 22:00 hours. From this, a arrival distribution is established. The hours of the day are divided in slices of 3 hours, starting at midnight. For each slices a mean inter-arrival time is assumed, the standard deviation of the corresponding normal distribution is taken to be 10% of the mean value, as can be seen in the table below.

Periods $[hours]$	00:00-03:00 03:00-06:00	06:00-09:00 21:00-00:00	09:00-12:00 18:00-21:00	12:00-15:00 15:00-18:00
Mean $[minutes]$	60	30	20	15
σ $[minutes]$	6	3	2	1.5
Ships $[number/hour]$	1	2	3	4

The used inter-arrival time distributions result in a peak arrival of vessels between 12:00 and 18:00 hours. The average total number of ships passing one chamber of the Volkerak lock complex is 60.

Engine

The Caterpillar 3512B Marine Propulsion engine has been used for the fuel consumption calculations (Caterpillar, 2013). This engine is installed on the *M.S. Carrera*, which dimensions are 110×11.4 $[m]$. The engine properties are listed in the table below.

Engine speed $[rpm]$	Engine Power $[kW]$	Fuel Rate $[l/h]$
1600	1249	287
1400	836.7	196.3
1200	526.9	125.6
1100	405.9	97.7
900	222.3	53.9
650	83.7	23

It is assumed that the ships have a fixed gearbox between the engine and the propeller. Klein Woud and Stapersma (2003) discusses that it is allowed to assume that the velocity is proportional with the speed of the propeller. Using a fuel price of €0.75 per litre, which is the average price over 2011 and 2012 (Backer van Ommeren, 2013), the coefficient c in Equation 3.1 is determined.

Emissions

The emissions can be calculated according to Gon and Hulskotte (2010), the average emission factors for diesel engines used in inland shipping (in $[g/kWh]$) are stated in the table below.

Substance	NO_x	PM	CO	VOC	SO_2	CO_2
$[g/kWh]$	9.4	0.4	2.0	0.4	0.004	662

From the simulation, the velocities, and thus the used engine power, are known. Also the sailing time between entering the system and arrival at the lock is known. Together with the emission factors, the total emissions can be calculated.

4.1.4 Lock

As discussed before, the dimensions of biggest chamber of the Volkerak lock complex is 331.5×24.1 $[m]$ and the ships dimensions are 110×11.4 $[m]$. Consequently, the chamber capacity is 4. However, in reality not all ships are equal and therefore the chamber capacity is varied in the experiments. The used capacities are 4, 5 and 6 ships. The average cycle time of the lock is assumed to be fixed and is 36 minutes as can be found in Sickinghe (2009).

4.2 Results

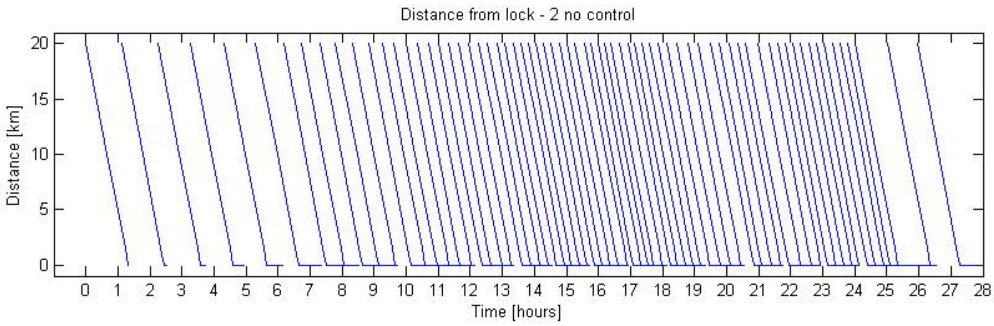
With the simulation settings discussed in the previous section, 3 experiments are defined. In these experiments the effect of the optimization algorithms, the influence of the chamber capacity and the influence of the length of the controlled waterway are investigated. In the next subsections the settings of these experiments and the results are discussed.

4.2.1 Optimization algorithms influence

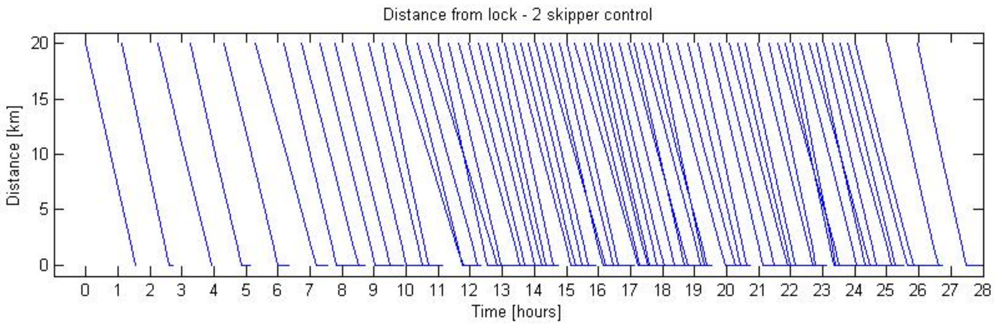
In the first experiment, the influence of the optimization algorithms are investigated with respect to the situations with no control and skipper control. The settings of the first experiment are:

Simulation time	28 hours
Cycle duration	36 minutes
Chamber capacity	5 ships
Waterway length	20 km

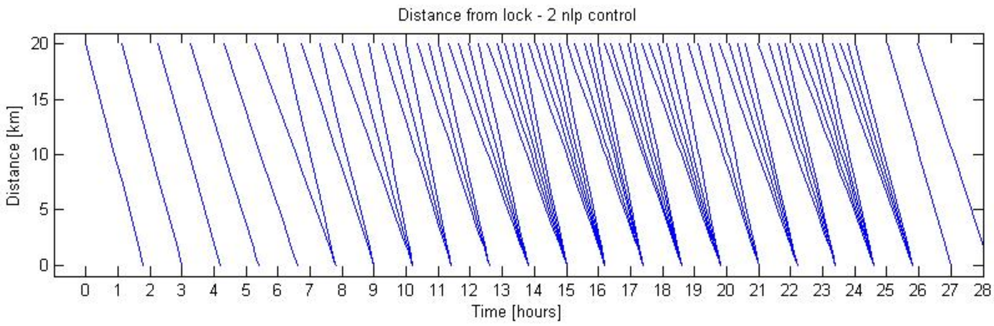
The results of these 4 simulations are depicted in Figure 4.1 to 4.5.



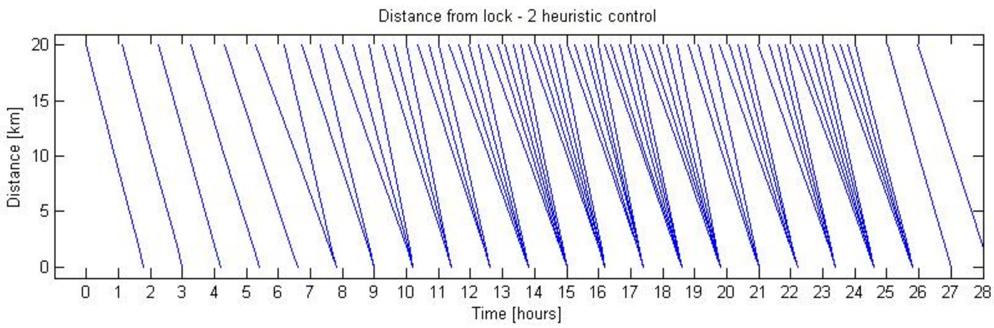
(a) No control



(b) Skipper control



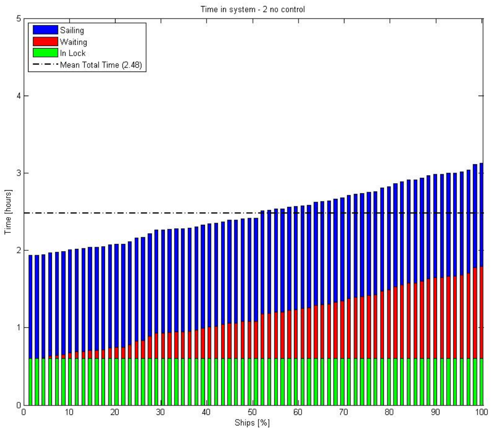
(c) NLP control



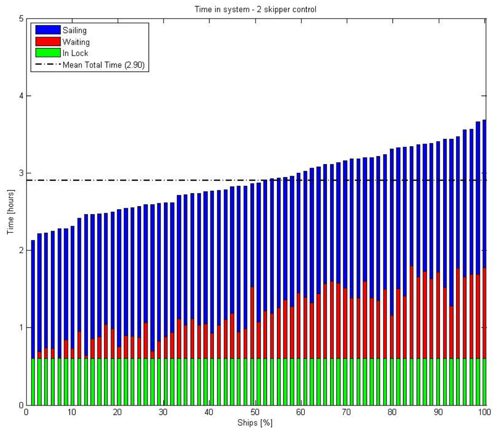
(d) Heuristic control

Figure 4.1: Influence optimization algorithms - Distance from lock

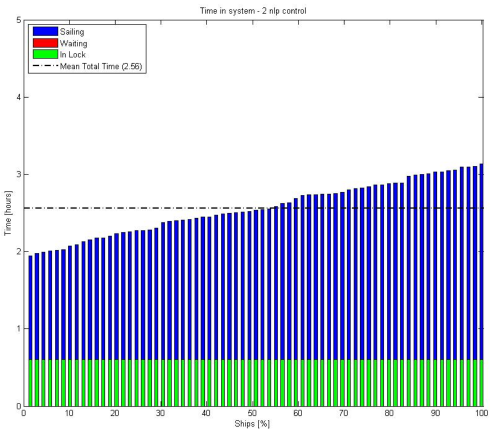
In Figure 4.1 each line represents a ship, the slope of the line indicates the velocity, the steeper the line, the higher the velocity. In the globally controlled situations (the lower two) all ships, which join the same lock cycle, arrive at the same time at the lock (at the beginning of the cycle), so the waiting times are reduced to zero. In the upper two graphs, the horizontal part of a line represent waiting at the lock.



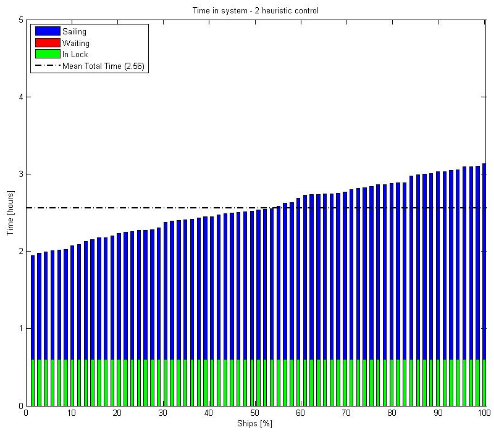
(a) No control



(b) Skipper control



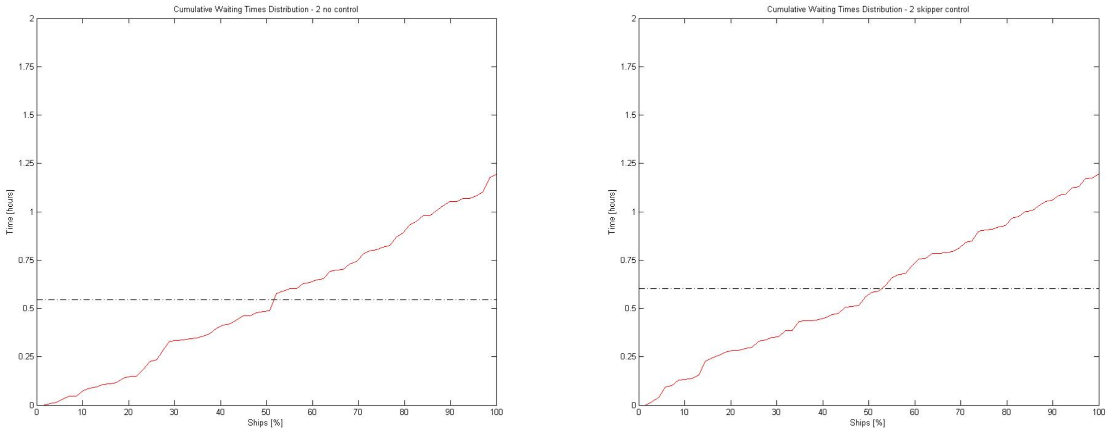
(c) NLP control



(d) Heuristic control

Figure 4.2: Influence optimization algorithms - Time in system

Despite of the waiting times, the mean total time in system is the shortest when all ships sail at their maximum velocity. However the difference with the optimized situations is small. The mean values of the two optimized situations are equal. The situation with skipper control is the worst.

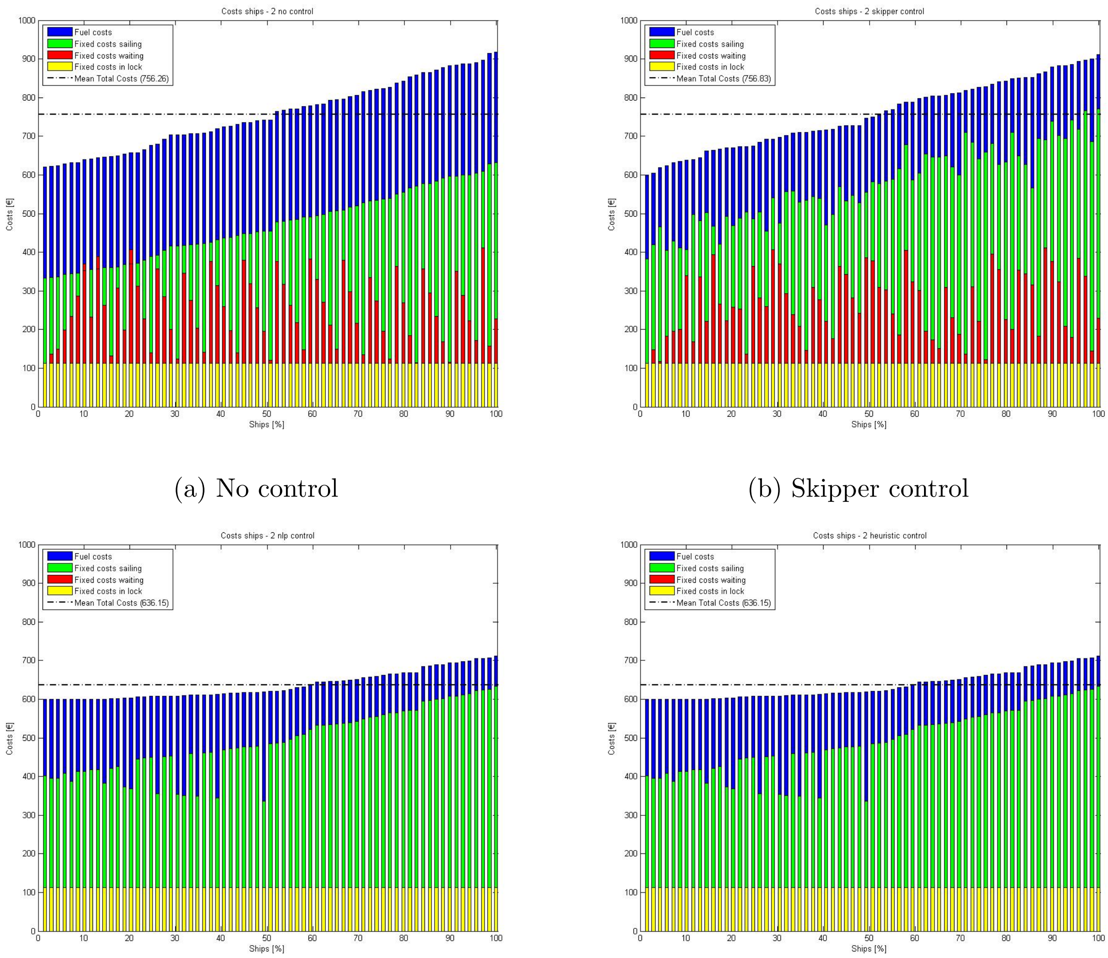


(a) No control

(b) Skipper control

Figure 4.3: Influence optimization algorithms - Cumulative waiting times distribution

The mean value of the waiting times is the highest for the situation without control, as can be seen in Figure 4.3. However, the maximum waiting time is in both situation approximately equal.



(a) No control

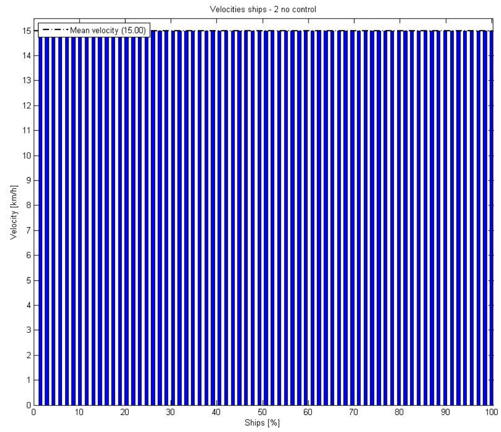
(b) Skipper control

(c) NLP control

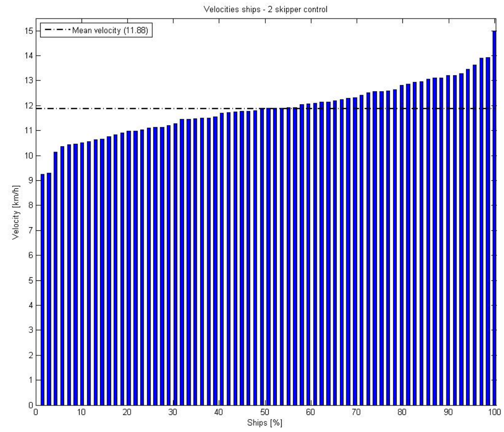
(d) Heuristic control

Figure 4.4: Influence optimization algorithms - Costs Ships

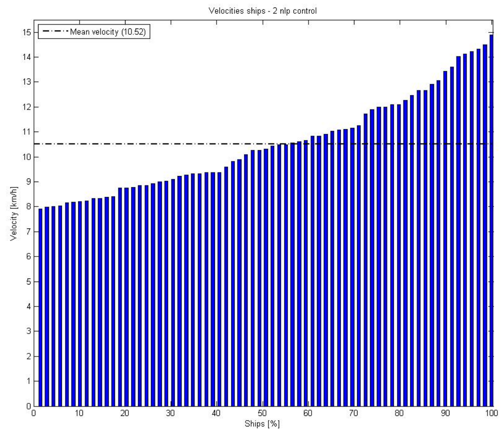
The costs reduction using the two optimization algorithms is considerably, as is depicted in Figure 4.4. Also the difference between the minimum costs and maximum costs has become much smaller in the globally controlled situations. The difference between the mean costs between no control and skipper control is negligible, the results for the NLP control and the heuristic control are equal.



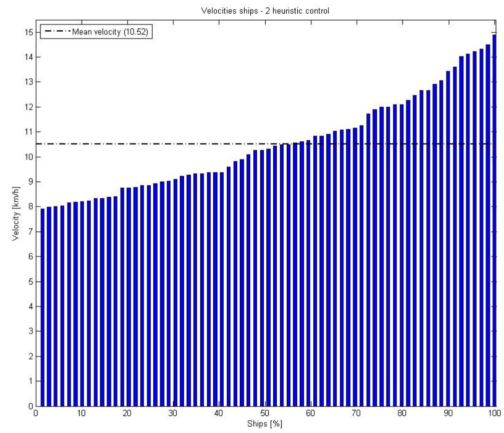
(a) No control



(b) Skipper control



(c) NLP control



(d) Heuristic control

Figure 4.5: Influence optimization algorithms - Velocities Ships

The velocity graphs, as depicted in Figure 4.5, confirm the agreement between the two optimized situations, the heuristic algorithm gives the same results as the non-linear (mixed integer) programming algorithm. Only the NLP control simulation takes about 15 minutes to run this experiment and compared to the less than a minute the model needs for the heuristic algorithm. The emissions have been calculated with the data presented in section 4.1.3, the results are given in kilograms and can be seen in the table below.

	NO_x	PM	CO	VOC	SO_2	CO_2
No control	15.63	0.66	3.32	0.66	0.007	1100.57
Skipper control	9.92	0.42	2.11	0.42	0.004	698.86
NLP control	7.94	0.34	1.69	0.34	0.003	559.50
Heuristic control	7.94	0.34	1.69	0.34	0.003	559.50

The results above show clearly the influence of the skipper control on the emissions with respect to the no control situation. From the way the fuel costs are related to the ships velocity and the emissions are related to used motor power (these are expressed in $[g/kWh]$), the emissions are proportional to the fuel costs. Therefore, when the fuel costs are decreased, the emissions are reduced as well. The NLP and heuristic control situations are even better. The preceding results are also used in the next two sections, together with results of other experiments.

4.2.2 Chamber capacity

The second experiment determines the influence of the chamber capacity on the optimization algorithms. The chamber capacity is varied between 4 and 6. The settings of the second experiment are listed below

Simulation time	28 hours
Cycle duration	36 minutes
Chamber capacity	4, 5 and 6 ships
Waterway length	20 km

Because visualizing all these results in graphs as in the previous section could be come unclear by the amount of graphs, the results of these simulations (mean, minimum and maximum values) are given in tables. In the first table the minimum, maximum and mean times in the system are shown, split into sailing time, waiting time and total time (the time in the lock is for all ships equal).

Capacity		4 ships			5 ships			6 ships		
Time		Sail	Wait	Total	Sail	Wait	Total	Sail	Wait	Total
No	Mean	1.33	0.56	2.49	1.33	0.54	2.48	1.33	0.54	2.48
Control	Min	1.33	0	1.93	1.33	0	1.93	1.33	0	1.93
	Max	1.33	1.79	3.11	1.33	1.19	3.13	1.33	1.19	3.13
Skipper	Mean	1.70	0.60	2.89	1.70	0.60	2.90	1.70	0.60	2.90
Control	Min	1.33	0	2.13	1.33	0	2.13	1.33	0	2.13
	Max	2.16	1.18	3.69	2.16	1.20	3.69	2.16	1.20	3.69
NLP	Mean	2.45	0	3.05	1.96	0	2.56	1.96	0	2.56
Control	Min	1.38	0	1.98	1.34	0	1.94	1.34	0	1.94
	Max	3.74	0	4.34	2.53	0	3.13	2.53	0	3.13
Heuristic	Mean	2.45	0	3.05	1.96	0	2.56	1.96	0	2.56
Control	Min	1.38	0	1.98	1.34	0	1.94	1.34	0	1.94
	Max	3.74	0	4.34	2.53	0	3.13	2.53	0	3.13

The results show that there is no difference between the simulations with a chamber capacity of 5 or 6 ships. In the NLP control and heuristic control the waiting times are always reduced to zero. The total time increases in the globally controlled situations. In the table below the minimum, maximum and mean costs are listed, split into fixed costs, fuel costs and total costs.

Capacity		4 ships			5 ships			6 ships		
Costs		Fixed	Fuel	Total	Fixed	Fuel	Total	Fixed	Fuel	Total
No	Mean	473.2	287.0	760.2	469.3	287.0	756.3	469.3	287.0	756.3
Control	Min	333.3	287.0	620.3	333.3	287.0	620.3	333.3	287.0	620.3
	Max	628.0	287.0	915.0	631.4	287.0	918.4	631.4	287.0	918.4
Skipper	Mean	573.4	181.6	755.1	575.3	181.6	756.8	575.3	181.6	759.8
Control	Min	382.8	109.2	600.4	382.8	109.2	600.4	382.8	109.2	600.4
	Max	771.5	287.0	911.2	771.5	287.0	911.2	771.5	287.0	911.2
NLP	Mean	612.9	103.2	716.1	490.3	145.9	636.1	490.3	145.9	636.1
Control	Min	345.0	36.4	599.4	335.9	79.6	599.3	335.9	79.6	599.3
	Max	935.9	268.0	972.3	632.8	282.6	712.4	632.8	282.6	712.4
Heuristic	Mean	612.9	103.2	716.1	490.3	145.9	636.1	490.3	145.9	636.1
Control	Min	345.0	36.4	599.4	335.9	79.6	599.3	335.9	79.6	599.3
	Max	935.9	268.0	972.3	632.8	282.6	712.4	632.8	282.6	712.4

In these results there is also no difference between a chamber capacity of 5 or 6 ships. In the case of a capacity of 4 ships, the mean total costs are reduced with approximately 6% in the case of NLP or heuristic control, however, the maximum total costs are much higher. In the case of a chamber capacity of 5 or 6 ships, the mean total costs are reduced with approximately 16% in the globally controlled cases. The distribution of the total costs in these cases are smaller than in the globally uncontrolled cases. The mean fuel costs are lower in the globally controlled situations than in the globally uncontrolled situation. This implies, as stated in Section 4.2.1, that the emissions are reduced as well.

4.2.3 Length of the controlled waterway

In the last experiment the influence of the length of the controlled waterway on the optimization algorithm is determined. Two lengths are tested, 20 and 30 km. The settings of the third experiment are listed below.

Simulation time	28 hours
Cycle duration	36 minutes
Chamber capacity	5 ships
Waterway length	20/30 km

First the times in the system are considered, split in sailing, waiting and total time (the cycle duration is constant). These results can be seen in the table below.

Length		20 km			30 km		
Time		Sail	Wait	Total	Sail	Wait	Total
No Control	Mean	1.33	0.54	2.48	2	0.56	3.16
	Min	1.33	0	1.93	2	0	2.60
	Max	1.33	1.19	3.13	2	1.17	3.78
Skipper Control	Mean	1.70	0.60	2.90	2.54	0.60	3.74
	Min	1.33	0	2.13	2	0	2.90
	Max	2.16	1.20	3.69	3.24	1.17	4.61
NLP Control	Mean	1.96	0	2.56	2.62	0	3.22
	Min	1.34	0	1.94	2.04	0	2.64
	Max	2.53	0	3.13	3.15	0	3.75
Heuristic Control	Mean	1.96	0	2.56	2.62	0	3.22
	Min	1.34	0	1.94	2.04	0	2.64
	Max	2.53	0	3.13	3.15	0	3.75

The time differences between the no control situation and the globally controlled (the NLP and heuristic control) situations are small, about 3% and 2% for the 20 an 30 km cases respectively. Compared to the skipper control situation, the differences are bigger, about 11% and 14% for the 20 and 30 km cases respectively. Next the costs are considered, the fixed, fuel and total costs are presented in the table below.

Length		20 km			30 km		
Costs		Fixed	Fuel	Total	Fixed	Fuel	Total
No	Mean	469.3	287.0	756.3	640.1	430.5	1070.6
Control	Min	333.3	287.0	620.3	500	430.5	930.5
	Max	631.4	287.0	918.4	794.7	430.5	1225.2
Skipper	Mean	575.3	181.6	756.8	784.3	273.3	1057.7
Control	Min	382.8	109.2	600.4	574.2	163.8	900.3
	Max	771.5	287.0	911.2	1001.7	430.5	1215.0
NLP	Mean	490.3	145.9	636.1	654.5	265.2	919.6
Control	Min	335.9	79.6	599.3	510.8	173.6	899.0
	Max	632.8	282.6	712.4	787.3	412.5	961.0
Heuristic	Mean	490.3	145.9	636.1	654.5	265.2	919.6
Control	Min	335.9	79.6	599.3	510.8	173.6	899.0
	Max	632.8	282.6	712.4	787.3	412.5	961.0

Regarding the costs, in the globally uncontrolled situation, the skipper control is slightly better then the no control situation. The costs differences in the 20 km case have already been discussed in the previous section. The total mean costs in the 30 km case for the globally controlled situations are approximately 13% smaller than in the skipper control case, and just as in the 20 km, the distribution of the total costs in the NLP (and heuristic) control situation is much smaller than in the skipper control situation.

Chapter 5

Conclusion and recommendations

In the concluding chapter of this research the obtained data will be used to formulate answers to the research questions, this is discussed in Section 5.1. Based on the conclusion, recommendations for further research are made in Section 5.2.

5.1 Conclusion

From the results presented in Section 4.2 it can be concluded that with the use of an global controller, the waiting times can be reduced to zero. In all cases this results in a longer mean time in system, however, the mean total costs are decreased in all cases and so are the emissions. The performance of the two optimization algorithms is equal, only the calculation time of the non-linear (mixed integer) programming (NLP) algorithm is, especially with many ships to consider, much higher than the heuristic one. In the 30 km simulation case, the NLP algorithm needed sometimes more than a hour to compute one optimization calculation, where the heuristic control simulation was finished within minutes.

The proposed research question:

Is it possible to develop intelligent waterways for the optimization of the flow of inland shipping vessels in order to reduce waiting times at locks and lowering the emission without an increase of the voyage costs for the vessels owner?

has been answered in this research. A simulation model using intelligent waterways is used to prove the possibility of optimizing the flow of inland shipping vessels. The results of the experiments have shown that the waiting times in the simulated cases, are reduced to zero simultaneous with a reduction of the emission and the total voyage costs. The disadvantage of the optimization is the increase of the mean total time in system.

The intelligent waterways are represented by waterway segments with knowledge. A segment knowns the location of each ship in its domain. This knowledge is shared with the global controller when an optimization calculation starts. Also the starting times of the lock cycles are known, together with the position of the ship the exact velocity can be determined in order to arrive, exactly at the beginning of the cycle, at the lock. Therefore an intelligent waterway network must have access to all relevant information of the waterway network at any time. Using this knowledge an optimization algorithm determines the best solution of the system at that moment regarding an objective.

5.2 Recommendations

In order to use the concept of intelligent waterways in the future in (parts) of the inland waterway network further research is recommended. In the proposed model, the duration of each cycle is equal, the does not agree with the reality because the cycle time is dependent on the water levels on both sides of the lock, the number of vessels in the chamber and the dimensions of the vessels in the chamber. Therefore the influence of varying cycle times should be investigated. Subsequently the capacity of the locks was fixed in each simulation run, however this depends on the dimensions of the vessels. The dimensions of the vessels were also not incorporated, implementing different vessels with different dimensions, engines, loading capacities and filling degrees is expected to have an influence on the optimization. When the above proposed changes would be made, the heuristic control algorithm would probably not compete any more with the non-linear (mixed integer) programming (NLP) algorithm, unless its possible to adapt the heuristic algorithm. Research into the affect of disturbances on the waterway should be performed, for example current and curved waterways will influence the mean velocity of the ship, so these factors should also be taken into account in the optimization algorithm. Also a system with more than one lock should be investigated, especially when the camber dimensions differ. Finally the NLP algorithm is currently too slow for usage in real time, the possibilities of speeding up the algorithm by adapting it needs further research.

Bibliography

E. Backer van Ommeren. Rapportage gasolieprijzen die relevant zijn voor de binnenvaart t/m december 2012. Memo, January 2013.

J. Banks and J.S. Carson. *Discrete-event system simulation.* Prentice-Hall, 1984. ISBN 0132155826.

I. Borst, F. Bongers, and R. van Ratingen. Evaluatie idvv programma. Technical report, Logica & Dialogic, January 2012.

J.U. Brolsma and K. Roelse. Waterway guidelines 2011. Technical report, Rijkswaterstaat, Centre for Transport and Navigation, December 2011.

E. Buckmann, J. Harmsen, T. Sendar, D. Goffin, and T. Scheltjens. Capaciteitsanalyse binnenvaart scheldegebied. Technical report, ECORYS Nederland BV and Resource Analysis, November 2009.

Caterpillar. Specification sheet 3512b marine propulsion, January 2013. URL `http://marine.cat.com`.

J.B. Evans. *Structures of discrete event simulation.* Ellis Horwood Limited & Halsted Press, 1988. ISBN 0470210974.

H.D. van der Gon and J. Hulskotte. Methodologies for estimating shipping emissions in the netherlands. Technical report, TNO Built Environment and Geosciences, 2010.

R. Groenveld, H.J. Verheij, and C. Stolker. Capacities of inland waterways. TU Delft, lecture notes, 2006.

J.J.S. Hengeveld. Optimization to reduce waiting times at locks. Master's thesis, Delft University of Technology, Mekelweg 2, Delft, Oktober 2012.

F.S. Hillier and G.J. Lieberman. *Introduction to operations research.* McGraw Hill, 9 edition, 2010. ISBN 978-007-126767-0.

J. Holtrop and G.G.J. Mennen. Approximate power prediction method. In *International Shipbuilding Progress*, volume 29, pages 166–170, 1982.

D. ten Hove. Scheepskarakteristieken van nieuwe grote schepen. Technical Report 24032.600/2, Marin and Rijkswaterstaat, Wageningen, February 2010.

M.C. Kerman. *Programmeren in Delphi.* Pearson Education Benelux, 2004. ISBN 90-430-0887-7.

H. Klein Woud and D. Stapersma. *Design of propulsion and electric power generation systems.* IMarEST, 2 edition, 2003. ISBN 1902536479.

The MathWorks Inc. Simevents: Getting started guide (r2012b), December 2012. URL `http://www.mathworks.com`.

R.L.J. Nieuwkamer and A.M. Rouwette. Mirt-verkenning capaciteitsuitbreiding volkeraksluizen. Technical report, Witteveen & Bos, September 2012.

M. Rahman and G. Rideout. Using the lead vehicle as preview sensor in convoy vehicle active suspension control. *Vehicle System Dynamics*, 50(12):1923–1948, 2012. ISSN 00423114.

B. Raphael and I.F.C. Smith. *Fundamentals of computer-aided engineering*. Wiley, 2003. ISBN 0-471-48709-0.

Rijkswaterstaat. Dokumentatie van een simulatiemodel voor schutsluizen en bruggen geschreven in prosim, Oktober 1981.

Rijkswaterstaat. Volkeraksluizen - achtergrondinformatie. Presentation, March 2011.

K.A. Saleh. *Software Engineering*. J. Ross Publishing, Inc., 2009. ISBN 978-1-932159-94-3.

H. Schildt. *Java: The complete reference*. McGraw-Hill, 8 edition, 2011. ISBN 978-0-07-160630-1.

D. Sickinghe. Functionele specificatie applicatie voor livra praktijkproeven. Technical report, Logica, December 2009.

H.P.M. Veeke. *Tomas: Tool for object-oriented modeling and simulation - User manual*, 2000. URL `http://tomasweb.com`.

H.P.M. Veeke and J.A. Ottjes. Tomas: Tool for object-oriented modeling and simulation. *In proceedings of Advanced Simulation Technology Conference (ASTC2000)*, pages 76–81, April 2000.

B.P. Zeigler, H. Praehofer, and T.G. Kim. *Theory of modeling and simulation*. Academic Press, 2 edition, 2000. ISBN 0127784551.