NTNU
Norwegian University of
Science and Technology

Ship Traffic Scheduling and Disruption Management for the Kiel Canal

A Simulation-Optimization Approach

Knut Viljar Hilstad
Guro Skjæveland

Industrial Economics and Technology Management
Submission date: June 2018
Supervisor: Kjetil Fagerholt, IØT
Co-supervisor: Xin Wang, IØT
Frank Meisel, Kiel University

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

Description of the Thesis

This thesis studies the Kiel Canal Ship Traffic Optimization Problem (KCSTOP). The problem involves finding conflict-free schedules while minimizing total travelling time for all ships traversing the Kiel Canal. The decisions to be made are which sequence the ships sail in, at which velocity each ship traverses at, and where and for how long each ship waits in order to mitigate potential conflicts.

The KCSTOP is modelled as a static and deterministic problem. However, a real-world planning situation is highly dynamic and stochastic. The Kiel Canal operates day and night, with ships continually arriving. In addition, unforeseen events may occur which disrupt the schedule in use, causing the need for replanning. Thus, the Dynamic Kiel Canal Ship Traffic Optimization Problem under Uncertainty (DKCSTOP) is established. The DKCSTOP is solved with a rolling horizon scheduling procedure, which is developed in a simulation-optimization framework. Different disruption management strategies are implemented in the framework in order to mitigate disruptions that take place in a dynamic and stochastic environment.

Preface

This Master's thesis is the concluding part of our Master of Science at the Norwegian University of Science and Technology, Department of Industrial Economics and Technology Management. The work is done in the field of Managerial Economics and Operation Research. The thesis is written during the spring semester of 2018 as a continuation of the project report written during the fall semester of 2017.

We would like to thank our academic supervisors Professor Kjetil Fagerholt and Postdoctoral Fellow Xin Wang, as well as our collaborator Prof. Dr. Frank Meisel at Kiel University, for their constructive feedback, valuable guidance and interesting discussions. We greatly appreciate your support and involvement throughout the thesis period.

We also wish to give an extra thanks to Prof. Dr. Frank Meisel for his hospitality during our visit to Kiel, October 2017.

Abstract

The Kiel Canal is a nearly 100 km long artificial waterway that connects the North Sea and the Baltic Sea through the north of Germany. The canal alternates between wide and narrow segments, and on the narrow segments the passing of large ships is considered unsafe. Therefore, a traffic management system is needed. The purpose of the system is to decide which ships that have to wait on the wider segments in order to ensure safe passage of all ships. The decisions affect the transit times of ships, and thus, have a major impact on the attractiveness for shipping companies to send their vessels through the canal.

In this thesis, the *Kiel Canal Ship Traffic Optimization Problem* (KCSTOP) is defined, and an optimization model is developed for the problem. The model captures the relevant traffic rules and safety requirements, with the goal of generating conflict-free schedules while minimizing total travelling time for all ships. The decisions to be made are which sequence the ships sail in, at which velocity each ship traverses at, and where and for how long each ship waits in order to mitigate potential conflicts.

The KCSTOP is modelled as a static and deterministic problem. However, the daily scheduling task is in reality exposed to both dynamism and stochasticity. Ships arrive at the canal continually, and events may occur that disrupts the schedule. Thus, the *Dynamic Kiel Canal Ship Traffic Optimization Problem under Uncertainty* (DKCSTOP) is established. The complexity of the problem makes it hard to model mathematically, and therefore a simulation-optimization framework is developed. The framework solves the problem by generating a composite schedule which is composed of consecutive solutions to constrained versions of the KCSTOP, found by using a rolling horizon scheduling procedure.

The traffic flow and possible disruptive events are simulated. The two disruptive events incorporated are ships arriving at the canal later than planned (delay), and ships sailing at a lower velocity than planned (slowness). In order to mitigate the impact of disruptions, appropriate strategies are implemented. *Real-time replanning* is one strategy, while another is to postpone waiting of ships. The latter is implemented by altering the mathematical formulation of the KCSTOP, and is referred to as the *weighted approach*.

How the different disruption management strategies perform in a dynamic and stochastic planning environment is tested in the developed simulation-optimization framework. The DKCSTOP is solved in the framework over 63 different scenarios. Results show that real-time replanning manages to nullify disruptions caused by delay. In the scenarios where slowness occurs, the solution is at most compromised by 3.8% compared to the disruption-free scenarios. When using the weighted approach, the solution quality is on average improved in all scenarios. In scenarios where both disruptive events occur frequently, the approach reduces average traversing time with nearly three minutes. This is roughly equivalent to 300 000 dollars saved for shipping companies yearly.

In addition to evaluating the performance of disruption management in the developed framework, several other features are examined: The limits on waiting times for ships, and the requirement that ship captains report the vessel's position prior arrival, to name a few.

Sammendrag

Kielkanalen er en nesten 100 km lang kunstig vannvei som forbinder Nordsjøen og Østersjøen gjennom Nord-Tyskland. Kanalen alternerer mellom brede og smale segmenter, og på de smale segmentene anses det som farlig at store skip passerer hverandre. Derfor er et trafikkhåndteringssystem nødvendig. Formålet med systemet er å avgjøre hvilke skip som må vente på de brede segmentene for å garantere sikker passasje for alle skip. Disse beslutningene påvirker transittidene til skipene som benytter kanalen, og har dermed stor innvirkning på om shippingselskap velger å sende sine fartøy gjennom kanalen.

I denne masteroppgaven defineres og modelleres optimeringsproblemet for skipstrafikken i Kielkanalen (Kiel Canal Ship Traffic Optimization Problem, forkortet KCSTOP). Modellen tar høyde for relevante trafikkregler og sikkerhetskrav i kanalen, og har som mål å generere konfliktfrie timeplaner samtidig som den totale transittiden for alle skip minimeres. Timeplanene skal inneholde informasjon om hvilken rekkefølge skip seiler i, hvilke segment hvert skip må vente på for å unngå potensielle konflikter og hvor lenge de må vente der. De skal også ha informajson om hastigheten til skipene.

KCSTOP er modellert som et statisk og deterministisk problem. I realiteten er den daglige planleggingen av skipstrafikk utsatt for både dynamikk og stokastisitet. Skip ankommer kanalen kontinuerlig, og hendelser kan forekomme som forstyrrer timeplanene. For å ta høyde for disse egenskapene defineres det dynamiske og stokastiske optimeringsproblemet for skipstrafikken i Kielkanalen (Dynamic Kiel Canal Ship Traffic Optimization Problem under Uncertainty, forkortet DKCSTOP). Problemet er vanskelig å modellere matematisk grunnet dets kompleksitet, og derfor blir et simulerings-optimeringsrammeverk utviklet. Rammeverket konstruerer en løsning til DKCSTOP ved å sette sammen deler fra gjentatte løsninger til KCSTOP. Løsningen blir funnet ved bruk av en rullende horisont.

Trafikkstrømmen og potensielle disruptive hendelser simuleres i rammeverket. De to disruptive hendelsene som omfattes er skip som kommer til kanalen senere enn planlagt (forsinkelse), og skip som seiler med lavere hastighet enn planlagt (langsomhet). Diverse strategier er implementert for å redusere negative virkninger av disse hendelsene. Replanlegging er en strategi, en annen er å utsette ventetidene til skip. Den sistnevnte strategien er implementert ved å endre den matematiske formuleringen av KCSTOP, og refereres til som den vektede tilnærmingen.

Hvordan de forskjellige strategiene presterer i dynamiske og stokastiske omgivelser er testet i simulerings-optimeringsrammeverket, hvor DKCSTOP er løst for 63 forskjellige scenarier. Resultatene viser at replanlegging annulerer forstyrrelser forårsaket av forsinkelser. I scenariene hvor langsomhet forekommer, blir løsningen forverret på det meste med 3,8% sammenlignet med scenarier hvor ingen disruptive hendelser forekommer. Ved å bruke den vektede tilnærmingen blir løsningskvaliteten i gjennomsnitt forbedret i alle scenarier. I scenarier hvor både forsinkelser og langsomhet forekommer hyppig, reduserer tilnærmingen gjennomsnittlig transittid med nesten tre minutter. Dette tilsvarer en årlig besparelse på ca. 300 000 dollar i året for shippingselskaper.

I tillegg til å evaluere hvordan de ulike strategiene presterer benyttes også rammeverket til å undersøke andre egenskaper ved problemet. Blant annet grensene for skips ventetider, samt kravet om at kapteiner må rapportere på fartøyets posisjon før ankomst til kanalen.

Contents

List of Tables

List of Figures

1 Introduction

In 2016, total volumes of international seaborne trade reached 10.3 billion tons, an increase by 2.6% from 2015 (UNCTAD (2017)). Furthermore, the United Nations Conference on Trade and Development projects a compounded annual growth rate of 3.2% in the sector between 2017 and 2022. This motivates the ongoing development in the broad study field of maritime routing and scheduling. The 98.7 km long Kiel Canal is an important piece in the world-spanning maritime transportation puzzle. By connecting the North Sea and the Baltic Sea through the north of Germany as shown in Figure 1, the canal is the basis for trade between the countries of the Baltic area and the rest of the world. Ships travelling between these two seas may save up to 250 nautical miles by travelling through the canal instead of using the alternative of sailing around the Jutland Peninsula (Kiel Canal Official Website (2017)). According to the Kiel Canal's official website, the canal is the most heavily used artificial waterway in the world. 30 000 vessels traversed the canal in 2017, an average of 80 ships a day, when disregarding pleasure crafts and other small boats. A total of 84 million tons of cargo was carried through which makes up 0.8% of the total international trade transported at sea. When considering these numbers, it is obvious that a well-functioning traffic management system in the canal is key.

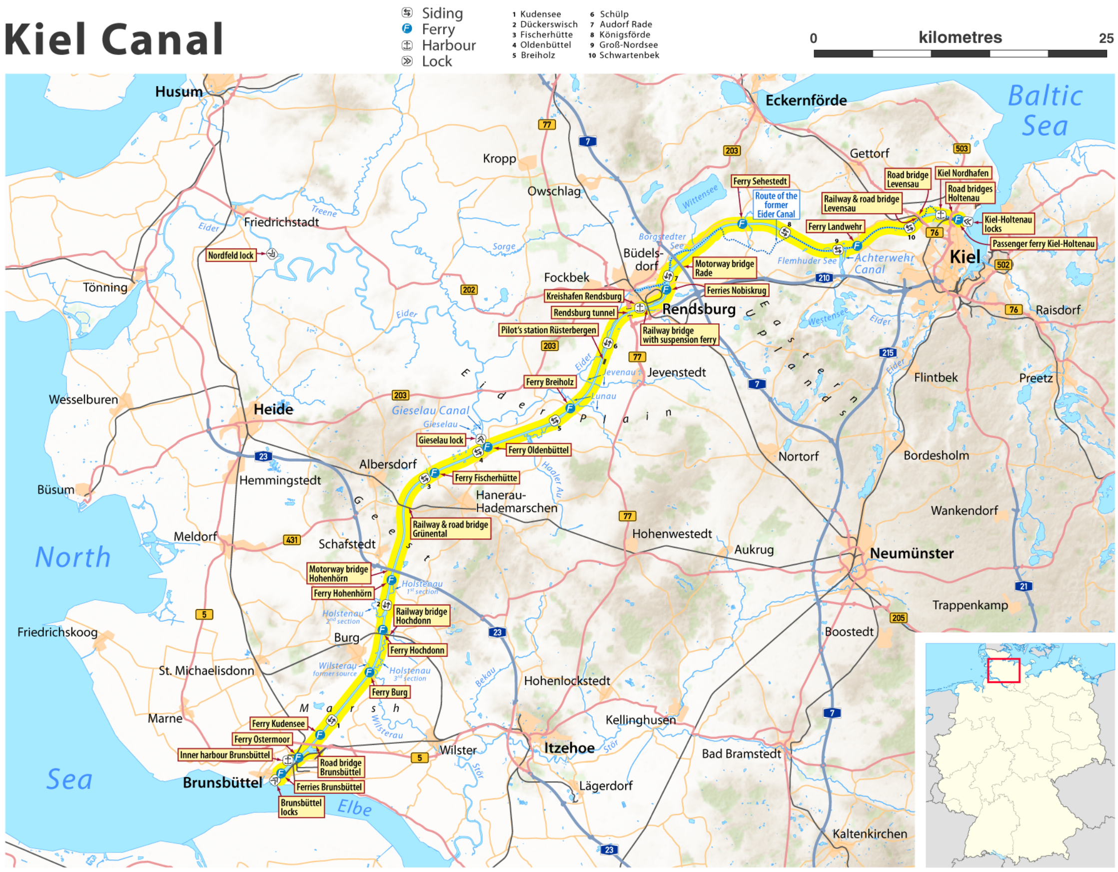


Figure 1: Map of the Canal. Wikipedia (2017)

1.1 Kiel Canal Ship Traffic Optimization Problem

The traffic in the Kiel Canal is managed by operators designated to the daily task of scheduling ships through the canal. The scheduling of ships is not a straightforward task to solve due to the infrastructure of the canal. It consists of alternating wide and narrow segments and the passing of two large ships might be impossible on the latter. A schedule should include the decision on which order ships travel in on different stretches of the canal; the time it takes for each ship to traverse the different stretches; if, where and for how long each ship must wait; and if and where each ship meets and passes another ship.

Throughout the canal there is no significant variation in the water level. Because of this, there are no locks located on the waterway. There are only locks at the entrances from the North Sea and the Baltic Sea, separating the canal from the two oceans. In other waterways, locks are typically the physical obstacle that limits the throughput and are used as the main tool for traffic management. Due to the lack of locks in the Kiel Canal, it is rather traffic lights that are used as the main tool to manage the traffic. Traffic lights are controlled by the operators and placed at each wide segment, signalling if a ship can enter the following narrow segment, or if it must wait.

Taking into account the heavy traffic through the Kiel Canal each day as well as the narrow segments limiting two-way traffic, it is probable for congestions to arise and long waiting times to occur. Some shipping companies choose to send their ships around the Jutland Peninsula to avoid the risk of delays and congestions. In order for the Kiel Canal operators to maintain their market shares, they are interested in minimizing waiting times for traversing ships and mitigating congestions. Shipping companies are also interested in reducing fuel consumption and sailing costs while delivering goods on time to their customers. Thus, there are possibilities for great economic savings by having an efficiently operated canal. Furthermore, in times of increased focus on the environmental aspect of global trading, the social surplus of reducing ship's CO_2-footprint is increased by having as goal to minimize ships travelling distance. This goal is achieved every time a shipping company sees a route through the Kiel Canal as more attractive than the detour around Jutland.

Based on the above, the desire to create schedules for traversing ships which minimizes their total travelling time is apparent. The problem of minimizing the travelling time for a certain amount of ships traversing the Kiel Canal under these complex circumstances is called the *Kiel Canal Ship Traffic Optimization Problem*, abbreviated KCSTOP. A solution to the problem is a schedule for a static horizon which is utterly conflict-free, and where all ships passes through the canal in the shortest possible time.

1.2 Dynamism and Disruption Management

The KCSTOP is a static and deterministic problem. However, the daily operation of the Kiel Canal is highly dynamic as ships enter and exit the canal continually. The times at which ships arrive are unknown for the operators up until some time in advance of their arrivals. Thus, dynamism in itself entails stochasticity. Furthermore, disruptions may occur during the execution of a schedule, making the current plan subject to changes. Two common stochastic events leading to disruptions are ships being delayed to the canal, and ships travelling at a lower velocity than planned. In other words, the planning situation is stochastic in nature. The problem of minimizing the total travelling time for ships traversing the Kiel Canal in a dynamic and stochastic environment is named the *Dynamic Kiel Canal Ship Traffic Optimization Problem under Uncertainty*, denoted DKCSTOP.

In contrast to the KCSTOP, the DKCSTOP is a complex system to model mathematically and solve analytically. A simulation-optimization framework is therefore developed in order to implement and solve the DKCSTOP using a rolling horizon approach. The approach involves iteratively replanning by solving constrained versions of the KCSTOP. The consecutive solutions make up the solution to the DKCSTOP. The framework mimics the real-life planning situation in the Kiel Canal by modelling both the dynamism of the problem and the disruptive events.

Real-time replanning is a strategy for managing disruptions. Disruptions may lead to the current schedule becoming infeasible or of poor quality, and replanning must be considered. Important decisions to be made when solving the DKCSTOP is: 1) when a complete regeneration of the schedule is needed, and 2) when parts of the current schedule can be re-used. The former is referred to as replanning without recycling, and the latter as replanning with recycling. Both of these replanning procedures are disruption management strategies implemented in the developed framework.

The KCSTOP is a time-continuous problem, giving rise to the problem of degeneracy. Several solutions appear equally good in terms of the objective value when regarding the KCSTOP isolated. However, these solutions may not be equivalent when included in the composite solution to the DKCSTOP. Thus, another disruption management strategy is developed in order to pick the degenerate solutions that are the most viable in the composite solution. This strategy involves postponement of waiting times for ships, and is referred to as the weighted approach.

1.3 Contributions

The motivation for our thesis is to provide the operators of the Kiel Canal with a decision support tool in their daily scheduling task. The main contributions are thus:

- An optimization model that solves the KCSTOP deterministically over a static horizon
- A simulation-optimization framework which implements and solves the DKCSTOP using a rolling horizon approach
- An evaluation of how the disruptive management strategies perform in a real-life planning situation, simulated in the framework

To our knowledge these contributions represents improvements of the available models and algorithms for ship traffic optimization in waterways. Furthermore, the simulation-optimization framework developed is tested on a data set based on real-life traffic data provided by the operators in the Kiel Canal. This, together with our contributions and analyses made throughout this thesis, may give valuable insight to the canal operators.

1.4 Remaining Outline

In Chapter 2, a precise description of the problem studied is given, followed by a brief review of relevant literature in Chapter 3. A mathematical formulation of the KCSTOP is presented in Chapter 4, and in addition, the weighted approach is proposed. In Chapter 5, a thorough explanation of the developed simulation-optimization framework is given. A computational study that evaluates the different disruption management strategies is conducted in Chapter 6. Finally, concluding remarks and suggestions for further research are presented in Chapter 7 and 8, respectively.

2 Problem Description

In this chapter, a possible way of designing both the Kiel Canal Ship Traffic Optimization Problem (KCSTOP) and the Dynamic Kiel Canal Ship Traffic Optimization under Uncertainty (DKCSTOP) is presented. A problem description of the KCSTOP is presented in Section 2.1, while the DKCSTOP is described in Section 2.2. How solutions to both the KCSTOP and DKCSTOP can be vizualized is shown in Section 2.3, followed by a brief summary of the problem in Section 2.4

2.1 Kiel Canal Ship Traffic Optimization Problem

The objective of the KCSTOP is to create a collision-free schedule for all ships planning to traverse the Kiel Canal, while minimizing total travelling time. Total travelling time includes the sailing time each ship uses to traverse the canal in addition to any waiting. A schedule tells the optimal sequence of ships travelling through the canal, the time it takes for each ship to traverse different segments of the canal, where and for how long each ship must wait, and where each ship meets and passes other ships. The following Subsections 2.1.1 - 2.1.3 contain necessary definitions and descriptions of different aspects of the problem which must be considered when solving the KCSTOP.

2.1.1 Canal Infrastructure

The infrastructure of the canal must be specified when formulating the KCSTOP. The canal is divided into several segments of which some are wide and referred to as *sidings*, and other narrow and referred to as *transits*. The canal alternates between sidings and transits. The transition from one segment to a next is referred to as a *border*. Each segment is given a *passage number* (PN) which is a proxy for the relative width of the respective segment. A larger passage number indicates a wider segment. The first and last segment represents the entrance and/or exit from the North Sea and the Baltic Sea, respectively. Along the canal there are ports where ships may want to dock. Thus, ships can enter and/or exit the canal on other segments along the canal than the first and last segment.

2.1.2 Nature of Ships

As with the canal infrastructure, the nature of traversing ships must be defined. The canal is operated bidirectionally. Ships travelling in the same direction are called *aligned*, otherwise *opposed*. A ship is referred to as *eastbound* if it travels from west to east, and *westbound* if it travels in the opposite direction. Ships intending to traverse the canal signal to the canal operator an *estimated time of arrival* (ETA), and which segment they will enter at, referred to as *entering segment*. The segment at which a ships exits is termed *exiting segment*. The operator assigns a *traffic group number* (TGN) to each ship depending on its size and draft. Direction, ETA, entering segment, exiting segment and TGN are throughout this thesis referred to as *ship parameters*.

A five-segment canal is illustrated In Figure 2. There are two sidings and three transits, with PNs of 12 and 8, respectively. Observe that the large, eastbound ship has received a TGN of 6, while the small, westbound ship has a TGN of 3.

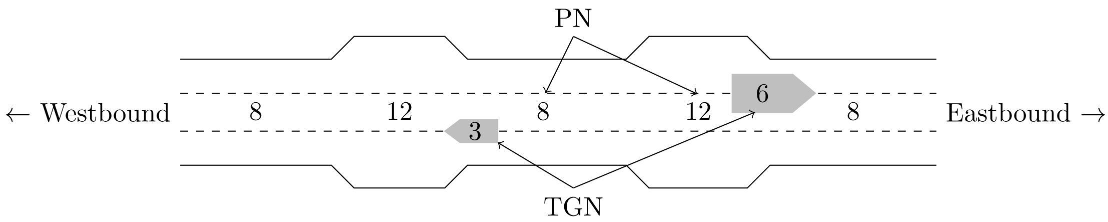


Figure 2: A simple sketch of a canal with two ships traversing

In the KCSTOP ship velocities are considered variable. This means that a ship can speed up and slow down, but it can never exceed its speed limit. The speed limit for a particular ship is based on its TGN. It is assumed that a ship can decelerate from its velocity to a full stop, and accelerate back, instantaneously. Note that variable velocity makes it possible for ships to change velocity from one segment to another, but not within a segment. I.e., on a particular segment the velocity is assumed to be constant.

2.1.3 Conflicts and Conflict Resolution

A central issue of the problem that must be defined is a *conflict* between two ships. A conflict represents the situation where overtaking or passing of two ships is presumed unsafe. Ships of any size can pass (when opposed) and overtake (when aligned) each other on sidings, while on transits there are physical limitations defining the circumstances under which ships are allowed to meet. Two opposed ships may meet and pass on a transit if the sum of their TGNs does not exceed the PN of that particular transit, i.e. the transit must be wide enough. If the ships do not meet this criteria, they are in conflict. On the other hand, aligned ships may under no circumstances overtake each other on transits. This is due to the maneuver of overtaking moving ships is considered dangerous on narrow segments. All pairs of aligned ships are therefore defined as being in conflict on all transits. All pairs of ships that are defined to be in conflict are contained in a set of *conflicting pairs of ships.*

A safety measure imposed by the canal operators is the so called *safety distance.* It is the minimum distance required between two aligned ships. The safety distance is dependent on the size of the rearmost ship and hence its TGN. The distance is translated into time via the velocity of the ship in front, and when doing so the safety measure is called *safety time.*

A conflict-free schedule implies that all conflicts must be resolved. To resolve conflicts, ships must take corrective maneuvers of which there exists two of. A ship may wait on sidings for ship(s) it is in conflict with for safe passing and overtaking, or it can vary its velocity. Thus, for how long and where a ship waits in addition to its traversing time on each segment, are decisions that must be made when solving the KCSTOP. Furthermore, the sequence of ships travelling through each segment must be decided.

Figure 3 is a simple illustration of conflicting versus non-conflicting ships. The aligned, eastbound ships of TGN 4 and 3 are by definition in conflict on the transit, and the rearmost ship waits on the siding prior to the transit. The waiting ship is also in conflict with the westbound ship of

TGN 4 on the transit ($4 + 4 > 7$). The waiting ship is dispatched when both the required safety time between itself and the other eastbound ship is satisfied, and the opposed, westbound ship has entered the siding. The two opposed ships passing each other on the transit are not in conflict ($4 + 3 \leq 7$).

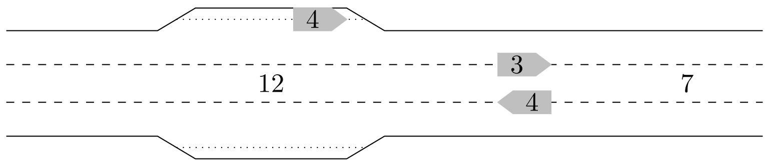


Figure 3: Illustration of conflicting and non-conflicting ships

2.2 Dynamic Kiel Canal Ship Traffic Optimization Problem under Uncertainty

The KCSTOP is a static problem as it seeks to compose an optimal schedule for a specific number of ships, or for a given planning horizon. Furthermore, it is deterministic as there are no uncertainty in the parameters. However, a real-world planning situation is highly dynamic and stochastic. The DKCSTOP denotes the Dynamic Kiel Canal Ship Traffic Optimization Problem under Uncertainty. What the DKCSTOP entails is explained in this section.

The Kiel Canal operates day and night, every day of the year, and ships are continually arriving at the canal. Thus, the schedule must be updated frequently to account for new ships arriving. In order to deal with dynamism, a replanning procedure is needed which replans whenever required. Replanning is done by resolving a constrained version of the KCSTOP, and hence obtaining a new schedule that replaces the current one in use.

Some managerial decisions exists which, to some extent, structures when a new schedule should be made and displace the current one in use. Firstly, the canal operators need information regarding ships that intend to traverse the canal some time prior to their arrivals. Hence, they require all ship captains to call a minimum time in advance of their arrival and announce their ship parameters. The point at which this announcement takes place is called the *announcement time* of a ship. Announcements of ships are related to the dynamism of the problem due to them occurring continually in the same manner as ships arrive continually. The announcement time has an underlying uncertainty, as it is unknown exactly when it occurs. Secondly, the operators of the canal define a *replanning threshold* (RT). The RT is the number of ships which may call the operators before they replan. Although not strictly necessary, it can be advantageous to have a RT of more than one ship in order to avoid new schedules being made too frequently.

It must be emphasized that rescheduling does not only take place when the number of announced ships exceed the RT. Other events may occur and disrupt the current schedule, requiring immediate attention. When disruptive events happen the replanning procedure serves as real-time disruption management.

The first event related to disruptions is delay. Due to various unforeseen incidents, such as unfavorable weather or technical issues, ships might be unable to arrive at the canal at the time specified when calling. This is especially true when considering that the time units used in the modelling of the KCSTOP are minutes, so a lateness of just a few minutes is considered a delay. If a ship is delayed, the ETA used by the planners in the current schedule is false. The delay may cause the plan to be incorrect, or even infeasible, both when the planners become aware of it, and when the delayed ship eventually arrives. Another common event is that a ship fails to keep its planned velocity when traversing a segment. Such an event may occur due to several reasons: Acute technical issues may force a ship to sail with a lower velocity; a captain may choose to sail slower on narrow segments if the ship is large, or if the ship is carrying fragile cargo; or a captain may opt to slow down for a safer passing of another ship, to name a few. When a ship traverses a segment slower than intended in the current schedule, it arrives at the border to the next segment later than planned. The event may cause disruptions in the current plan, making it subject to changes.

When events occur that make the current plan unusable, disruption management ensures new schedules to be made. Thus, the schedule might be updated several times during ships' journeys through the canal. The path each ship actually followed when their journey ends is a composition of repeated solutions to constrained versions of the KCSTOP. This *composite schedule* is a solution to the DKCSTOP. The process of finding the composite schedule is referred to as the rolling horizon scheduling procedure.

2.3 Visualization of Solutions

By utilizing a so called *time-distance diagram* one can visualize a schedule. Such diagrams assist the canal operators in their planning and scheduling process. An example of such a diagram is given in Figure 4. Time in minutes is measured along the y-axis and increases downwards, the length of the canal in kilometers is measured along the upper x-axis, and the segment number is found along the lower x-axis. Segments are always counted from left to right. The passage number of each segment is also shown in the diagrams, positioned in the bottom of each segment. Sidings are shaded grey, while transits are white. A ship's position is measured as segments from its entry point. Thus, an eastbound ship traversing the full length of the canal has a starting segment at 1, and moves from left to right, while a westbound ship traversing the full length of the canal starts at segment 23 and moves from right to left. Note that the canal topology in Figures 4-5 is for illustrative purposes, and does not represent the actual Kiel Canal.

2.3.1 Static Horizon

In Figure 4 a time-distance diagram is drawn for a canal consisting of seven segments with two opposed ships, an eastbound ship i (red) and a westbound ship j (blue). The eastbound and westbound ships enter at segments 1 and 7, respectively. The diagram sheds light on several aspects. First off, it illustrates ships' velocities. The TGN of the two ships differ, and hence their speed limits are different. This is apparent from the slight difference in slope for the two lines; the blue ship of higher speed limit has a flatter slope and is covering a larger distance of the canal per unit of time. Secondly, the diagram shows a conflict, and how it is resolved. The dashed line indicates how the blue ship would proceed through the canal if it travelled straight through. However, since the the sum of the TGNs exceeds the passage number of the transit in the middle

$(4+6>8)$, the two ships are defined as being in conflict on Segment 4. One possible resolution is to let ship j wait on Segment 5 until ship i has passed, depicted by the blue solid line. Note that several solutions exists that are equally good. For instance, a solution could be that ship j waits in Segment 7 instead of Segment 5.

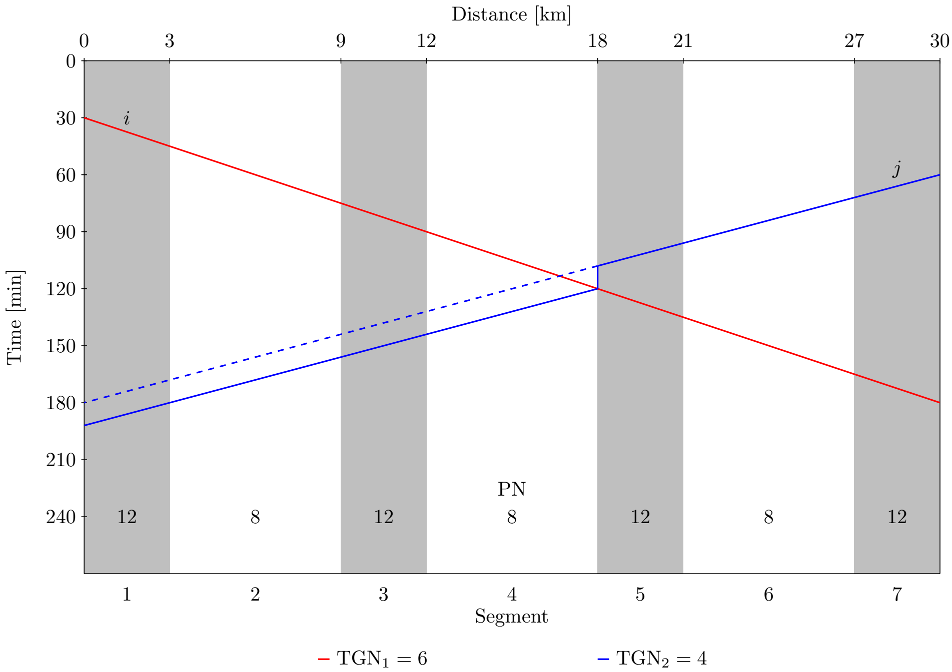
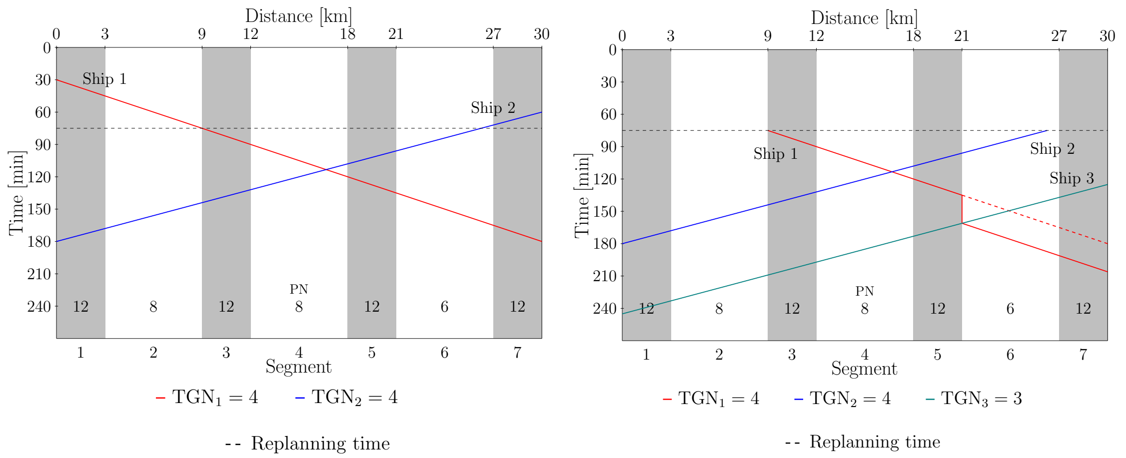


Figure 4: A time-distance diagram of a solution to the KCSTOP

2.3.2 Rolling Horizon

When solving the DKCSTOP with a rolling horizon scheduling procedure, a time-distance diagram is produced after each replanning, visualizing the new schedule obtained. An initial schedule is depicted in Figure 5a, and the new one after replanning is depicted in Figure 5b. The schedule changes for ship 1 when going from the current, or initial schedule, to the new schedule. The dashed red line in the latter represents how the path of ship 1 would be if no changes were made in the schedule. However, since ship 1 and the newly arrived ship 3 are in conflict on Segment 6 $(4+3>6)$, ship 1 now waits on Segment 5, shown in Figure 5b. Once again, several equally good solutions exists, e.g. ship 1 could wait on Segment 3 instead of Segment 5. The new schedule starts from the time at which replanning occurs, here denoted replanning time, and onwards. This is shown in Figure 5b.

When one planning horizon is over, the time-distance diagram produced is for the composite schedule which is the solution to the DKCSTOP. The composite solution of the two schedules depicted in Figures 5a and 5b is shown in Figure 6. The dashed and solid lines originates from the initial and the new schedule, respectively.



(a) Initial schedule (b) New schedule after replanning

Figure 5: Time-distance diagram for two consecutive solutions to KCSTOP

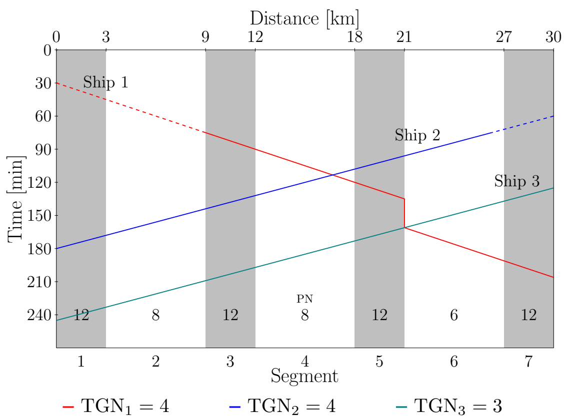


Figure 6: Composite schedule, solution to the DKCSTOP

2.4 Summary

The KCSTOP entails the problem of scheduling ships through the Kiel Canal. The objective when solving the KCSTOP is to minimize total travelling time for all ships traversing the canal while obtaining conflict-free schedules for these ships. The decisions to be made are which sequence the ships sail in, the time it takes for each ship to traverse different segments of the canal, and at which sidings and for how long each ship waits. The daily operation of the Kiel Canal is both dynamic and stochastic in nature as ships enter and exit the canal continually, and moreover, disruptive events may occur. The DKCSTOP entails the problem of scheduling ships through the canal in a dynamic and stochastic environment. A solution to the problem is obtained by iteratively solving constrained versions of the KCSTOP.

3 Literature Review

In this chapter, relevant literature is presented and discussed. Research regarding optimization of ship traffic in waterways is presented in Section 3.1. Train timetabling and scheduling problems for single-track networks share many of the same properties as the KCSTOP and the DKCSTOP, and publications regarding train scheduling are therefore discussed in Section 3.2. Section 3.3 covers disruption management, reviewing publications on the topic. Finally, in Section 3.4 literature relevant for simulation in combination with optimization is discussed, both in general and with specific examples from the field of maritime shipping.

3.1 Ship Traffic Optimization

The PhD-thesis written by Lübbecke et al. (2014) study the Ship Traffic Control Problem (STCP), a problem regarding ship traffic optimization in the Kiel Canal. The authors aim to create a support tool for finding bottlenecks in the canal. By finding these bottlenecks it is possible to evaluate different enlargement options and pick the one that gives the most added value per dollar spent. The problem is solved by utilizing a fast, online heuristic. The work done is highly relevant for the problem studied in this thesis, and have been of great inspiration for the development of the KCSTOP. However, capacity constraints are relaxed in our thesis, and furthermore we extend the STCP model by enabling ships to vary their velocity in the KCSTOP. Lübbecke et al. (2014) consider a static problem. That is, the schedule created is for a given list of ship, and what happens after the time span of this list has passed is not considered. It is also assumed that all parameters are deterministic, and that no unforeseen events occur. The DKCSTOP studied in this thesis is on the other hand dynamic and, moreover, stochastic due to several parameters having an underlying uncertainty.

Rulusçu et al. (2009) formulate a mathematical model that mimics the current scheduling procedure in the Strait of Istanbul. Furthermore, a vessel scheduling algorithm is constructed in order to test how closely the model resembles the procedure. The model is able to recreate the actual traffic management system in 90% of the cases studied. Unlike the problem studied in this thesis, ships are given priorities which are used to create a conflict-free schedule. Such priority weights are not easy to implement in the KCSTOP, as weights should be as unbiased as possible and justifiable for the ship captains. Thus, how to set ship priorities is worthy of a study in its own right, and this approach is not considered any further.

The Waterway Ship Scheduling Problem (WSSP) is studied by Lalla-Ruiz et al. (2016) and is modelled as a Mixed Integer Linear Problem (MILP). The goal of the WSSP is to schedule incoming and outgoing ships of a waterway while minimizing waiting times, and consequently reducing fuel consumption and emissions while the ships are waiting at the anchorage. Several parts of their mathematical formulation are similar to the mathematical model of the KCSTOP. However, the KCSTOP entails waiting along the canal and not just at entry and exit points. Commercial optimization software used to solve the WSSP for the Shanghai Port in China falls through when the test instances grow large, and the authors turn to greedy heuristics and simulated annealing (SA). Results show that the SA approach provides high-quality solutions in a short period of time. The solution to the DKCSTOP is a composite of consecutive solutions to constrained versions of the KCSTOP. The instances which the KCSTOP is solved over in this thesis are small enough for commercial optimization software to solve with satisfying optimality gaps. It could nevertheless be interesting to develop heuristics for solving the KCSTOP to see if better solutions to the DKCSTOP can be found.

Similar to the KCSTOP, Li and Lam (2017) study how to construct conflict-free schedules for vessel arrivals within a seaport while minimizing ship delays. They start with an initial schedule containing several conflicts due to each ship being scheduled straight through the seaport traffic system without any waiting. An unconventional algorithm to generate conflict-free schedules is developed, which iteratively forces ships in conflict in the initial schedule to wait based on some vessel priority list. Thereafter, delay is minimized by searching among the generated, conflict-free schedules for a local optimum. A simulator is developed with the purpose of testing the scheduling algorithm. The results show that the algorithm is generates conflict-free schedules efficiently in a real-world environment. Once again, the lack of vessel priorities in the KCSTOP distinguishes the core of the solution approach used in this thesis and the one used in the study by Li and Lam (2017). Furthermore, the simulation-optimization framework developed in this thesis does not only have as purpose of testing solution approaches, but moreover it is used as a tool to solve the DKCSTOP.

3.2 Train Scheduling

While the STCP studied by Lübbecke et al. (2014) is closely related to the KCSTOP, other papers on traffic optimization in waterways either share a small set of properties with the KCSTOP, or there are fundamental differences in the modelling approach. To find literature with more similarities it is necessary to look further than the field of maritime scheduling. In the field of train scheduling, there exists research where both the nature of the problem and the modelling approach are similar to what we study in this thesis. The obvious difference is that papers on train scheduling consider railways and trains as transportation modes, but this is more or less analogue to waterways and ships. Especially relevant for both the KCSTOP and DKCSTOP are papers concerning the *train dispatching problem*. The train dispatching problem is defined by Cordeau et al. (1998) as the problem of creating a possible schedule for a set of trains, so that no train overrules a system of constraints that decides the operation of the trains.

Zhou and Zhong (2007) study a single-track train timetabling problem formulated as a a generalized resource-constrained project scheduling problem, minimizing total train travelling time. Precedence between conflicting trains are added iteratively and chronologically in a branch and bound algorithm to eliminate conflicts. The nature of the problem is similar to the KCSTOP with single-tracks and stations being analogue to transits and sidings, respectively (although the KCSTOP is less strict regarding passing rules on transits where opposed ships may pass each other if the transit is wide enough). The main difference lies in the solution approach. When solving the KCSTOP, all conflicts are mitigated at once, and not in an iterative manner as the authors of the reviewed paper do. Having said that, the iterative solution approach is somewhat analogue to the solution approach of the DKCSTOP. Conflicts are mitigated iteratively in batches, each batch being a solution to a constrained version of the KCSTOP.

Castillo et al. (2009) build on the work done by Zhou and Zhong (2007), but propose several extensions to the model: 1) Traversing speed on each segment is considered variable, 2) Several trains can be on the same segment at the same time if they travel in the same direction, and 3) several objective functions are considered. Extension 1 and 2 makes the mathematical model even more similar to the KCSTOP. The first extension does however make solutions to the problem degenerate due to the equivalency between waiting and slow traversing. In order to find a unique solution, three objective functions are proposed and the problem is solved iteratively, each time searching only among the solutions found in the previous iteration. The problem with degenerate solutions is directly transferable to both the KCSTOP and DKCSTOP. A ship can wait on several segments, and on which it waits is equivalent with respect to the objective of minimizing total travelling time. Furthermore, waiting and slow traversing are also equivalent with respect to the objective. Thus, the solution to both the KCSTOP and the DKCSTOP is often degenerate.

Higgins et al. (1996) study single track railways, and develop an on-line scheduling model for trains traversing these railways. The scheduling task is similar to the one in the DKCSTOP, but the model presented uses priority weights just as Li and Lam (2017) and Ulusçu et al. (2009). In the follow-up paper by Higgins and Kozan (1998), the expected delay of individual trains and train networks are modelled. The model is shown to be able to replicate actual train delays to a relative high precision.

As mentioned in Section 2.2, ships being delayed to the canal is a disruptive event in the DKCSTOP. Thus, it is of interest to study how these delays are distributed. Both Higgins et al. (1996) and Higgins and Kozan (1998) assume exponentially distributed delays. This assumption is also used by Yuan et al. (2002) and Krüger et al. (2013). Krüger et al. (2013) also develop several more sophisticated distributions. Harris (2006) studies both an exponential distribution and a q-exponential model, and finds that the latter is more accurate in modelling train delays.

3.3 Disruption Management

According to Yu and Qi (2004) the concept of disruption management refers to *the real time dynamic revision of an operational plan when disruptions occur*. Furthermore, it is stated that *this (disruption management) is especially important in situations where an operational plan has to be published in advance, and its execution is subject to severe random disruptions.* Due to such disruptions, the original plan may not remain optimal, or even feasible. The act of revising the plan consists of making a new one subject to constraints and objectives of the evolved environment while minimizing the negative impact of the disruption.

The authors highlight that great efforts have been made during the past several decades to cope with uncertainty in planning situations. The approaches to deal with uncertainty are classified into in-advance planning and real-time replanning. In the former an initial plan accounting for future uncertainties is created, while in the latter the current plan in use is revised whenever needed while it is executed. In this thesis, real-time replanning is the main focus.

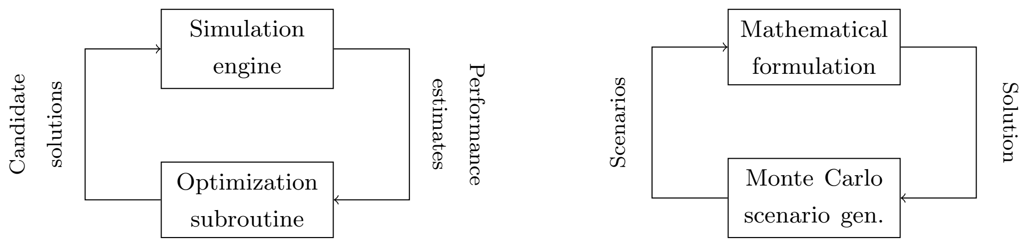
Visentini et al. (2014) present a comprehensive review on methods for real-time vehicle schedule recovery in transportation services reported during 2005-2013. Common for all, the rescheduling process starts with an analysis of the disruption and how it affects the initial, off-line schedule.

The next step is to create a recovery plan if necessary. The new schedule must take into account the current schedule in use, and the current state of the problem when the disruption triggering replanning occurred. The survey regards both buses, trucks, airplanes and trains as transportation modes. Rescheduling problems for the latter is the most relevant for this thesis, as train transportation is often analogue to the infrastructure and ship traffic in the Kiel Canal. A solution method frequently used in the articles reviewed is to solve a relaxation of the problem, and to find a feasible solution to the actual problem by using some sort of heuristic.

Two papers in the survey are of special interest. The first paper is by Törnquist and Persson (2007), who study N-tracked railway traffic rescheduling during disturbances. They formulate the problem as a MILP, develop and test four different strategies for solving the rescheduling problem. The second paper is by Acuna-Agost et al. (2011), who extend the mathematical formulation presented by Törnquist and Persson (2007). Several solution methods are proposed for solving the problem. One method is a so-called right-shift rescheduling, where both the order in which trains traverse segments and stations, and the track assignments are maintained by fixing integer variables from the current schedule. Another method is a complete regeneration in which a full rescheduling is conducted, the drawback being the computational time. These two method are analogue to the replanning procedures utilized in this thesis when solving the DKCSTOP.

3.4 Simulation Optimization

Amaran et al. (2014) define simulation optimization as "*the optimization of an objective function subject to constraints, both of which can be evaluated through a stochastic simulation*". Stand-alone optimization models may not suffice in accounting for uncertainty, whilst simulation models cannot make decisions. During the last decades there has been an increased focus on combining simulation and optimization. Fu (2002) discusses how this is combined, usually by letting optimization be subservient to the simulation routine or vice verca. Figure 7 shows the two different approaches. Figure 7a depicts how optimization is a subroutine to the simulation engine. Figure 7b depicts the opposite, where a Monte Carlo simulation is the add-on used to generate scenarios for the optimization engine which solves the mathematical formulated problem. In this thesis we refer to *simulation-optimization* as the approach illustrated in Figure 7a; the simulation engine models a real-world situation which requires optimization of some kind, and calls an optimization procedure within the simulation. Amaran et al. (2014) provide an extensive review of research literature proposing algorithms and applications combining the two study fields, and the reader is referred to this survey for further in-depth studies.



(a) Optimization for simulation (b) Simulation for optimization

Figure 7: Approaches of combining simulation and optimization (Fu (2002))

Literature on modelling vessel traffic in waterways by simulation is growing. During the last decade, several ports and canals have been used as subject of one or more research papers on simulation.

Traffic simulation in the Panama Canal is the topic of a study done by Golkar et al. (1998). An extensive and complex simulation model is developed and used as a tool for scenario analysis. Thiers and Janssens (1998) examine the port of Antwerp. A simulation model is created and is used to simulate how the construction of a future container quay will affect the traffic flow. The model simulates actual navigational rules, lock operations and tide flows that exist in the port. The presumption from the simulation results is that the quay only increases the hindrance of ships with a few percent. Cortés et al. (2007) focus on simulating freight transport in the Guadalquivir River, Spain. By simulating both the maritime transportation and the logistics activities in the berths, it is shown how access infrastructure is key in order to increase the traffic volumes. Specifically, increasing the depth of the estuary and the dimensions of the locks should be prioritized. Similarly to Cortés et al. (2007), Smith et al. (2009) utilize simulation in order to evaluate different infrastructural improvements in the Upper Mississippi River. In addition, alternative decision rules are investigated. Results show that infrastructural improvements have a larger positive impact on the system than improved decision rules, but the costs associated with such infrastructural changes are considerably larger. The Delaware River is studied by Almaz and Altiok (2012), who use simulation in order to measure the effect of increasing the depth of the river. By building different scenarios and testing them in the simulator, the conclusion is that no significant efficiency is gained for bulk and general cargo vessels if a deepening is conducted.

Several papers regard the Strait of Istanbul and investigate the traffic within it by simulation. Köse et al. (2003) develop a model simulating the traffic under different scenarios, in order to discuss the effects of increased marine traffic due to new oil pipelines being built in the strait. Results of the simulation show that both waiting times and the probability of accidents are presumed to increase. Almaz et al. (2006) and Özbaş and Or (2007) simulate the traffic in the Strait of Istanbul with the purpose of analyzing how various rules and regulations in the strait affect the traffic. Results from a scenario analysis indicate that parameters such as availability of piloting and tugboat services; arrival rate of vessels; vessel profiles; current and visibility; and overtaking rules are of importance for the traffic in terms of number of traversed vessels, average transit and waiting times.

Within the larger field of maritime transportation, simulation in combination with optimization have become more evident. For instance Fagerholt et al. (2010) study strategic planning in tramp and industrial shipping, using a Monte Carlo scenario generation as in Figure 7b, in addition to using simulation to evaluate solutions to the optimization problem as in Figure 7a. The simulation engine uses a rolling horizon principle, revealing information as time elapses. Halvorsen-Weare and Fagerholt (2011) study the supply vessel planning problem, with the goal of determining optimal fleet size and mix of supply vessels and their weekly voyages and schedules which should be robust. Robustness is tested and evaluated by a simulation model. Vessel routing and scheduling in the liquefied natural gas business is studied by Halvorsen-Weare et al. (2013). Routes and schedules created should be robust with respect to unpredictable sailing times and daily LNG production rates. Robustness strategies are added to an optimization model that solves the ship routing and scheduling problem, in addition to developing a simulation-optimization framework with the purpose of evaluating the strategies. The framework imitates a real-life planning situation and allows for replanning if certain conditions are met. Fischer et al. (2016) study the roll-on-roll-of fleet deployment problem. Similarly to Halvorsen-Weare et al. (2013), a simulation-optimization framework is used to mimic the real world by adding disruptive events to the problem, and a replanning procedure based on a rolling horizon heuristic is called whenever needed. Furthermore, robustness strategies are proposed and evaluated in this simulation-optimization framework.

To the best of our knowledge, there is no literature regarding the use of simulation-optimization in the specific study field of ship traffic management and scheduling in waterways. The only research found on combining simulation and optimization to some extent is done by Li and Lam (2017) (see Section 3.1 for the review of the paper). They develop a simulator with the sole purpose of testing their developed ship scheduling optimization algorithm. The authors do not utilize simulation-optimization, which in this thesis is defined as calling an optimization procedure within the simulation. However, for future research the authors do state that the next step would be to integrate real-time corrective measures, i.e. replanning, in the simulations.

The study of the KCSTOP and the DKCSTOP falls within the field of traffic management and scheduling in inland waterways. As a final remark on the reviewed literature, there is a lack of research in the field that combines simulation and optimization with the purpose of implementing a disruption management system that replans whenever necessary. Thus, the work done in this thesis will contribute to fill some of the gap in the existing literature.

4 Mathematical Formulation

In this chapter, the mathematical formulation of the KCSTOP is presented. In Section 4.1, a brief listing of modelling assumptions is given. Notation is introduced in Section 4.2 followed by Section 4.3, where the mathematical formulation is established. Lastly, an enhancement strategy is developed in Section 4.4.

4.1 Modelling Assumptions

A set of assumptions are needed when modelling the problem. Firstly, it is assumed that a ship's entering segment, exiting segment and thus direction, along with its estimated time of arrival (ETA) at the canal, is signalled by the ship captains to the canal operators prior to their arrival. The signalled ETA is in reality the time at which a ship arrives at the entrance gates to the lock, but for modelling purposes an ETA is the time at which a ship arrives at its entering segment. Secondly, it is assumed that a ship's traffic group number (TGN) is a known parameter prior to its arrival. Thirdly, capacity of sidings are ignored, implying that an unlimited number of ships may potentially be situated in the same siding at the same time. Next, it is assumed that it is more convenient for a canal operator to impose waiting on ships rather than specific velocities. The former can easily be done in practice by signalling with the traffic lights where and for how long a ship must wait. Finally, the velocity of each ship can vary between a full stop and its speed limit, but the velocity is constant on each segment. It is assumed that no time is lost due to a ship's retardation and acceleration.

4.2 Notation

In the mathematical description of the problem, let $\mathcal{V}$ be the set of all ships, indexed by i and j. Let $\mathcal{V}^E$ and $\mathcal{V}^W$ represent the set of eastbound ships and westbound ships, respectively. Thus, $\mathcal{V} = \mathcal{V}^E \cup \mathcal{V}^W$.

The set of all transits and sidings are denoted by $\mathcal{T}$ and $\mathcal{S}$, respectively. Let $\mathcal{P}$ be the set of all segments indexed by p such that $\mathcal{P} = \mathcal{T} \cup \mathcal{S}$. Let $\underline{P}_i$ and $\overline{P}_i$ denote the segment p at which ship i starts and ends its journey at, respectively. Furthermore, let $\mathcal{P}_i$ represent the set of all segments that i will traverse during its journey. $\mathcal{P}_i$ is constructed as follows:

$$\mathcal{P}_i = \{\underline{P}_i, \underline{P}_i + 1, \underline{P}_i + 2, ..., \overline{P}_i - 1, \overline{P}_i\}, \qquad i \in \mathcal{V}^E$$
$$\mathcal{P}_i = \{\underline{P}_i, \underline{P}_i - 1, \underline{P}_i - 2, ..., \overline{P}_i + 1, \overline{P}_i\}, \qquad i \in \mathcal{V}^W$$

Further, the set of transits and sidings which ship i traverses are denoted by $\mathcal{T}_i$ and $\mathcal{S}_i$, respectively, so that $\mathcal{P}_i = \mathcal{T}_i \cup \mathcal{S}_i$. If a ship enters the canal at a siding, the set $\mathcal{T}_i$ starts at $\underline{P}_i + 1$, while $\mathcal{S}_i$ starts at $\underline{P}_i$. Contrary, if it enters at a transit, the set $\mathcal{T}_i$ starts at $\underline{P}_i$, while $\mathcal{S}_i$ starts at $\underline{P}_i + 1$.

Two input parameters are the traffic group number, $TGN_i \in [1, 6]$ for ship $i \in \mathcal{V}$, and the passage number, P_p of segment p. The passage number of the transits are given by $P_p \in [6, 8]$ for $p \in \mathcal{T}$, and for sidings, $P_p = 12$ for $p \in \mathcal{S}$. To determine if two opposed ships i and j are in conflict on segment p, the sum of their TGNs are checked against the passage number of the segment. Recall from Subsection 2.1.3, that *all* aligned ships are in conflict. Let $\mathcal{C}_p^O$ be the set of all conflicting pairs of opposed ships on segment p, defined according to Equation (1), and $\mathcal{C}_p^A$ be the set of all conflicting pairs of aligned ships on segment p, defined according to Equation (2):

$$\mathcal{C}_p^O = \{(i,j) \mid i \in \mathcal{V}^E, j \in \mathcal{V}^W, TGN_i + TGN_j > P_p\} \tag{1}$$

$$\mathcal{C}_p^A = \{(i,j) \mid i,j \in \mathcal{V}^E, i \neq j\} \cup \{(i,j) \mid i,j \in \mathcal{V}^W, i \neq j\} \tag{2}$$

$\mathcal{C}_p$ then represents the set of all conflicting pairs of ships on segment p such that $\mathcal{C}_p = \mathcal{C}_p^A \cup \mathcal{C}_p^O$.

The minimum time required for ship i to traverse a segment p is denoted $\underline{S}_{ip}$. Let L_p be the cumulative length of the canal up until segment p measured from west, and V_i be the maximum speed limit for ship i. S_{ip} is then calculated by:

$$\underline{S}_{ip} = \frac{L_{p+1} - L_p}{V_i}, \qquad i \in \mathcal{V}, \quad p \in \mathcal{P}_i \tag{3}$$

In the KCSTOP, a ship i with $TGN_i < 5$ have a speed limit of $V_i = 15$ km/h $= 250$ m/min, while a ship j with $TGN_j \geq 5$ have a speed limit of $V_j = 12$ km/h $= 200$ m/min.

All ships have an ETA denoted $T_{ETA,i}$ for ship i. The earliest possible time a ship i can enter a segment p is denoted $\underline{T}_{ip}$, and is calculated recursively by:

$$\underline{T}_{i\underline{P}_i} = T_{ETA,i}, \qquad i \in \mathcal{V} \tag{4}$$

$$\underline{T}_{ip} = \underline{T}_{ip-1} + \underline{S}_{ip-1}, \qquad i \in \mathcal{V}^E, \quad p \in \mathcal{P}_i \setminus \{\underline{P}_i\} \tag{5}$$

$$\underline{T}_{ip} = \underline{T}_{ip+1} + \underline{S}_{ip+1}, \qquad i \in \mathcal{V}^W, \quad p \in \mathcal{P}_i \setminus \{\underline{P}_i\} \tag{6}$$

Assume that ship j follows behind ship i on a given segment. The required safety distance between the two ships is denoted D_{ij}. The canal operators have set the safety distance to be 600 meters if ship j's traffic group number is $TGN_j \leq 3$ and 1000 meters if $TGN_j > 3$. This distance is translated into a required safety time by:

$$H_{ij} = \frac{D_{ij}}{V_i}, \qquad \{i,j\} \in \mathcal{V} \tag{7}$$

Since the safety distance between two aligned ships depends on the size of the ship following behind, $H_{ij} \neq H_{ji}$ is possible.

The next parameter to be defined is big M. M is set to be the number of ships, $|\mathcal{V}|$, multiplied by the time it takes for the slowest ship to traverse the full canal with no conflicts, i.e. sailing at its speed limit:

$$M = |\mathcal{V}| \cdot \max_{i \in \mathcal{V}} \left\{ \sum_{p \in \mathcal{P}_i} \underline{S}_{ip} \right\} \tag{8}$$

The following decision variables are defined: w_{ip} represents the waiting time for ship i on segment p (waiting time variable); t_{ip} is the time at which ship i enters a segment p and starts to traverse it (entering segment time variable); s_{ip} is the time ship i uses to traverse a segment p (traversing time variable); the binary variable z_{ijp} which equals 1 if ship i is scheduled before ship j on segment p, and 0 if ship j is scheduled before ship i on segment p (ship sequence variable). An overview of all notation can be found in Appendix A.1.

4.3 Mathematical Model

In this section, the mathematical model of the KCSTOP is formulated.

The objective function (9) minimizes the total travelling time of all ships sailing through the canal. The first component represents the total waiting time for all ships in the canal, while the second represents the total traversing time for all ships. The third component represents the time ships have to wait outside the canal. Note that the waiting time and traversing time variables are equivalent with respect to the objective. According to the assumption that waiting is preferred to prolonged traversing, traversing time is penalized by a weight, $(1+\varepsilon)$, where ε is a small, positive number. The penalty is introduced in order to allocate the majority of the time needed for conflict resolution to the waiting time variables instead of traversing time variables.

$$\min \qquad \sum_{i\in\mathcal{V}}\sum_{p\in\mathcal{S}_i} w_{ip} + \sum_{i\in\mathcal{V}}\sum_{p\in\mathcal{P}_i}(1+\varepsilon)s_{ip} + \sum_{i\in\mathcal{V}}\left(t_{i\underline{P}_i} - T_{ETA,i}\right) \tag{9}$$

Constraints (10) ensure that the time at which eastbound ship i arrives at a siding must equal the time it arrived at the transit prior to this particular siding, plus the time required to traverse that transit. Similarly, constraints (11) ensure that the time at which eastbound ship i arrives at a transit must equal the time it arrived at the siding prior to this particular transit, plus the time required to traverse the siding in addition to any potential waiting on the siding. Note that t_{ip} is the time eastbound ship i arrives at segment p, and t_{ip+1} is the time the ship arrives at the next segment which is equivalent to when it leaves segment p. The same logic holds for constraints (12) and (13), but for westbound ships. Note that for these constraints t_{ip-1} replaces t_{ip+1}.

$$t_{ip} + s_{ip} - t_{ip+1} = 0, \qquad i \in \mathcal{V}^E, \quad p \in \mathcal{T}_i \tag{10}$$

$$t_{ip} + s_{ip} + w_{ip} - t_{ip+1} = 0, \qquad i \in \mathcal{V}^E, \quad p \in \mathcal{S}_i\backslash\{|\mathcal{S}|\} \tag{11}$$

$$t_{ip} + s_{ip} - t_{ip-1} = 0, \qquad i \in \mathcal{V}^W, \quad p \in \mathcal{T}_i \tag{12}$$

$$t_{ip} + s_{ip} + w_{ip} - t_{ip-1} = 0, \qquad i \in \mathcal{V}^W, \quad p \in \mathcal{S}_i\backslash\{1\} \tag{13}$$

Constraints (14)-(15) and (16)-(17) establish precedence between aligned and opposed conflicting ships, respectively. Constraints (14) ensure that if ship i traverses segment p before ship j ($z_{ijp} = 1$), then ship j must enter segment p after ship i, in addition to having the required safety time H_{ij} between them. Constraints (15) imply the opposite, i.e. when ship j travels through segment p before ship i ($z_{ijp} = 0$). Constraints (16) and (17) require that if $z_{ijp} = 1$ then ship j cannot enter segment p before ship i has left the segment, and vice versa for $z_{ijp} = 0$.

$$t_{ip} + H_{ij} - t_{jp} \leq M(1 - z_{ijp}), \qquad \{i,j\} \in \mathcal{C}_p^A, \quad p \in \mathcal{P}_i \tag{14}$$

$$t_{jp} + H_{ji} - t_{ip} \leq M z_{ijp}, \qquad \{i,j\} \in \mathcal{C}_p^A, \quad p \in \mathcal{P}_i \tag{15}$$

$$t_{ip} + s_{ip} - t_{jp} \leq M(1 - z_{ijp}), \qquad \{i,j\} \in \mathcal{C}_p^O, \quad p \in \mathcal{P}_i \tag{16}$$

$$t_{jp} + s_{jp} - t_{ip} \leq M z_{ijp}, \qquad \{i,j\} \in \mathcal{C}_p^O, \quad p \in \mathcal{P}_i \tag{17}$$

Constraints (18) and (19) state that the order of aligned ships cannot change in a transit segment.

That is, they cannot under any circumstances overtake each other in a transit.

$$z_{ijp+1} - z_{ijp} \leq 0, \qquad \{i,j\} \in \mathcal{C}_p^A \cap \mathcal{V}^E, \quad p \in \mathcal{T}_i \tag{18}$$

$$z_{ijp-1} - z_{ijp} \leq 0, \qquad \{i,j\} \in \mathcal{C}_p^A \cap \mathcal{V}^W, \quad p \in \mathcal{T}_i \tag{19}$$

Constraints (20) enforce that the traversing time for ship i on segment p equals or exceeds the minimum traversing time possible due to speed limits. Constraints (21) enforce that the arrival time of a ship i at a segment p equals or exceeds the earliest possible arrival time.

$$s_{ip} \geq \underline{S}_{ip}, \qquad i \in \mathcal{V}, \quad p \in \mathcal{P}_i \tag{20}$$

$$t_{ip} \geq \underline{T}_{ip}, \qquad i \in \mathcal{V}, \quad p \in \mathcal{P}_i \tag{21}$$

Constraints (22)-(23) express sign and binary constraints on the remaining variables.

$$w_{ip} \geq 0, \qquad i \in \mathcal{V}, \quad p \in \mathcal{S}_i \tag{22}$$

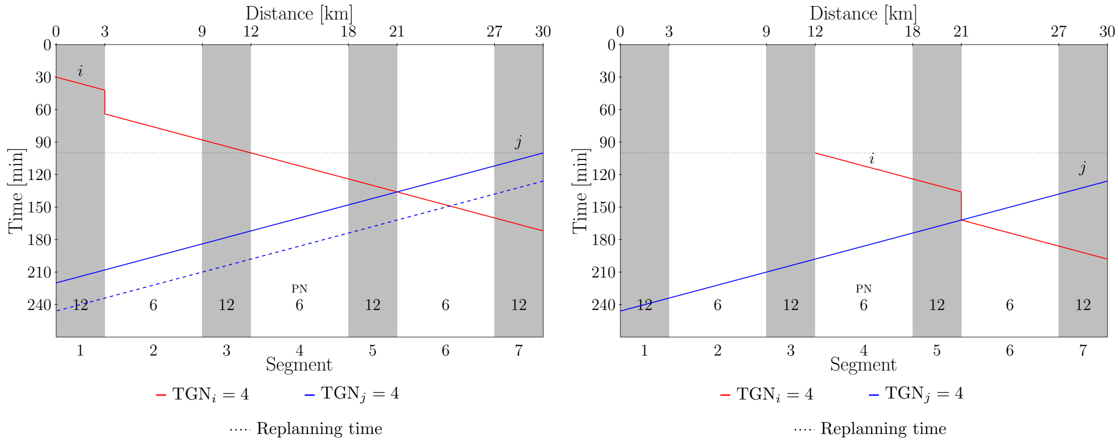
$$z_{ijp} \in \{0,1\}, \qquad \{i,j\} \in \mathcal{C}_p, \quad p \in \mathcal{P}_i \tag{23}$$

4.4 Weighted Approach

In this section, an enhancement strategy that rewards postponement of waiting is modelled. Introducing the strategy has a two-fold purpose. The first purpose is related to the problem of degenerate solutions. Often there are several possibilities for where a ship can wait, and some of these possibilities are equal in terms of the objective value. In fact, since time is continuous there are always a plethora of degenerate solutions. Degeneracy often increases the computational time needed when solving linear programming problems, thus increasing the time spent on searching the branch-and-bound tree. The first purpose of the strategy is therefore to remove some of the degeneracy from the problem, in order to reduce computational time. The second purpose is related to choosing between degenerate solutions to the KCSTOP. Recall that the solution to the DKCSTOP is a composite schedule of repeated solutions to constrained versions of the KCSTOP. Even though the degenerate solutions are equal in terms of the objective function stated in Equation (9), which degenerate solution that is included in the composite schedule may yield different solutions to the DKCSTOP. The second purpose is therefore to make sure that the choice of degenerate solution benefits the composite schedule to the greatest extent possible.

The enhancement strategy rewards solutions where waiting is postponed. By postponing waiting as long as possible for each ship, it is possible to accumulate new information regarding events before waiting is distributed. This strategy may result in better scheduling decisions with respect to the composite schedule. Therefore, the weighted approach is deemed a disruption management strategy.

An example of a situation where this strategy might be beneficial is shown in Figure 8. Note that the canal topology is for illustrative purposes. In Figure 8a the red ship, i, is scheduled to wait at Segment 1 for the blue ship, j. However, as indicated by the blue dashed line, ship j is delayed. The canal operators become aware of the delay at the time indicated by the black dotted line. Re-running the scheduling algorithm at this time produces the schedule shown in Figure 8b and ship i must wait once again at Segment 5. Thus, by utilizing these two solutions, ship j traverses the canal in 120 minutes, while ship i must in fact wait for ship j twice, and uses 168 minutes to traverse it.

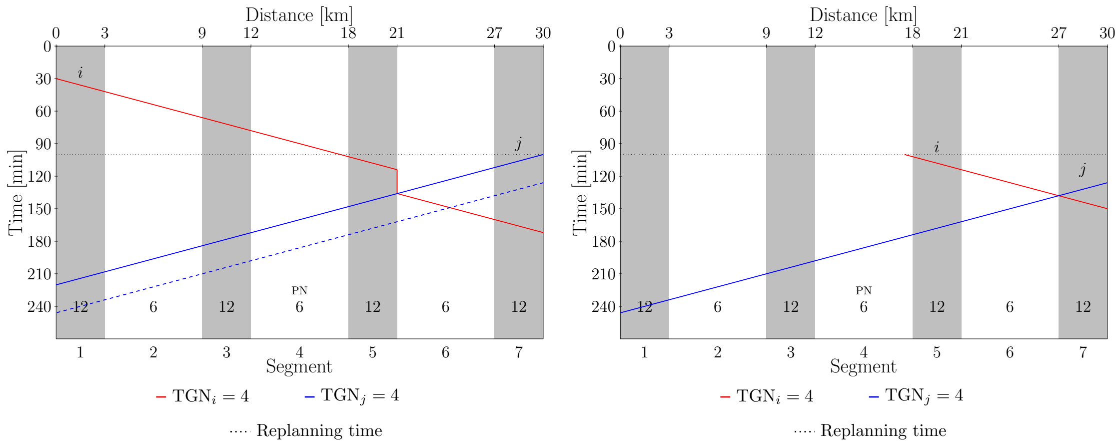


(a) Initial schedule (b) New schedule after replanning

Figure 8: Example of scheduling with early waiting

Contrast this to the situation in Figure 9. In 9a the exact same situation as in Figure 8a is shown, but this time another degenerate solution is picked where ship i waits at Segment 5 instead of Segment 1. When the canal operators are informed of the delay, ship i has not yet waited for ship j. Due to this, i is now able to meet and pass ship j at Segment 7. The outcome is that both ship i and j are able to traverse the canal in 120 minutes, an improvement of 48 minutes.

This example illustrates how it may be beneficial to wait for more information before issuing waiting in a dynamic and stochastic planning situation. Note that the mathematical model presented in Section 4.3 is not able to distinguish the two initial solutions in Figure 8a and Figure 9a without the enhancement strategy developed here.



(a) Initial schedule (b) New schedule after replanning

Figure 9: Example of scheduling with postponed waiting

In order to postpone waiting for all ships, some new notation is introduced. Let W_{ip} denote the weighting coefficient associated with the waiting time variable for each ship $i \in \mathcal{V}$ and for all sidings, $p \in \mathcal{S}$. The weights are created so that $W_{ip} > W_{ip+1}$ for eastbound ships, and vice versa for westbound ships. Lower weights are assigned to sidings towards the end of a ship's journey, and thus, late waiting is rewarded. Furthermore, let $\underline{W}_i$ denote the weighting coefficient associated with the time ship i potentially must wait outside the canal before starting its journey. $\underline{W}_i$ is created so that $\underline{W}_i > W_{ip}$ for all $p \in \mathcal{S}$.

As before, waiting is favored over slow traversing, and the traversing time component of the objective function is assigned a marginally higher weight than the largest waiting weight, $(\underline{W}_i + \varepsilon)$. This is in order to avoid that the solver transfers waiting time to the traversing time variables. With this new notation introduced, the objective function incorporating the enhancement strategy is as follows:

$$\min \quad \sum_{i \in \mathcal{V}} \sum_{p \in \mathcal{S}_i} W_{ip} w_{ip} + \sum_{i \in \mathcal{V}} \sum_{p \in \mathcal{P}_i} \left(\underline{W}_i + \varepsilon\right) s_{ip} + \sum_{i \in \mathcal{V}} \underline{W}_i \left(t_{i\underline{P}_i} - T_{ETA,i}\right) \tag{24}$$

When solving the mathematical model with the above stated objective, it is referred to as solving the KCSTOP with *the weighted approach.* When using the objective stated in Equation (9), it is referred to as solving the problem with *the basic approach.*

5 Simulation-Optimization Framework

As mentioned in Section 2.2, it is necessary to have a rolling horizon scheduling procedure in order to solve the DKCSTOP. Such a procedure is developed in a simulation-optimization framework. The DKCSTOP is implemented in this framework, where ships are continually entering and leaving the canal, and events related to both dynamism and disruptions are occurring. In addition, the framework makes replanning possible whenever required. As the name suggests, the simulation-optimization framework consists of two parts, one simulator and one optimizer. These are related since the simulator needs an up-to-date schedule to function at all times, while the optimizer needs input parameters produced by the simulator in order to create a schedule. Figure 10 shows the interaction between these two.

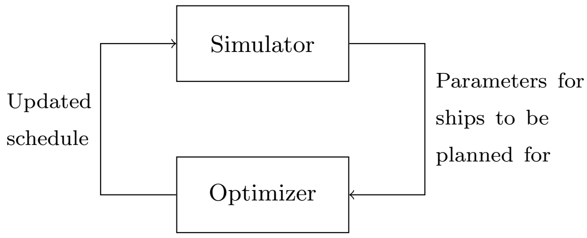


Figure 10: Simulation-optimization framework

In addition to an initial schedule, the simulator is initialized with a list of events and a data set based on real-life traffic data provided by the operators in the Kiel Canal. This data set contains a list of ships and their respective parameters. Note that real-life traffic data only exists for ships that fully traverse the canal, implying that the entering and exiting segments of ships are the first or last segment of the canal, depending on their direction.

The remainder of this chapter is dedicated to show how the simulator interacts with the optimizer in order to solve the DKCSTOP. Firstly, a set of simulation assumptions is presented in Section 5.1, before some new notation is introduced in Section 5.2. The modelling of events is described in the following section, Section 5.3. Section 5.4 contains a detailed explanation of the replanning procedure which links the simulator and optimizer, before a description of how a composite solution to the DKCSTOP is obtained is given in Section 5.5. Finally, Section 5.6 holds a thorough explanation of the flow control of the entire simulation-optimization framework.

5.1 Simulation Assumptions

Some assumptions are needed in order to implement the DKCSTOP efficiently, and these are presented in this section.

There are three assumptions related to a ship's announcement time. Firstly, it is assumed that all captains announce their estimated time of arrival (ETA) at some point. In other words, all ships have an announcement time. Secondly, it is assumed that the operators require captains to call a minimum time in advance, so that it is impossible for a ship to arrive immediately after calling. Finally, as the operators require captains to call a minimum time in advance, it is assumable that they call in the vicinity of this minimum time, and not too long before. Thus, ships' announcement times will lie within an interval, referred to as an *announcement time interval*, with the lower bound being the minimum required time in advance of arrival. In the remainder of this thesis, when referring to increasing or decreasing the announcement time interval, it means shifting the whole interval. It should however be understood that in reality this is equivalent to the operators changing the required minimum time, that is, the lower bound.

There exists four underlying modelling assumptions when considering ships being slow. First off, it is assumed that there is no correlation between a ship's slowness and on which segment(s) it is slow on. That is, a ship is neither more, nor less, likely to be slow on the following segment if it slow on the current segment. Next, it is assumed that when a ship sails slower than planned on a segment, it manages to instantaneously slow down and speed up on the slow segment and the following segment, respectively. I.e., no time is lost in acceleration or retardation. Thirdly, it is assumed that the canal operators have no knowledge of the actual velocity a ship sails with on a segment before it reaches the next border: Either it arrived as planned, or it did not. If a ship is slow and the latter is the case, it is assumed that canal operators instantly know where it is positioned at this point in time. Finally, it is assumed that large ships more frequently slow down their velocity than small ships. The logic behind is that large ships often carry heavy cargo, and due to both size and cargo, these ships require more caution when traversing narrow segments. Hence, a ship's slowness is correlated with the ship's TGN.

A last modelling assumption is that all schedules are produced instantaneously. This means that the simulator pauses the simulation time while the optimizer solves the KCSTOP.

5.2 Notation

In this section, some important terms and mathematical notation is introduced. The notation introduced in Section 4.2 is re-used wherever possible. Note that the terms plan, schedule and solution are used interchangeably.

Every time the simulator interacts with the optimizer, a schedule is made. Such an interaction is defined as a *roll*. At all times there exists an up-to-date schedule in the simulator, referred to as the *current* schedule. The schedule made in the next roll is named the *new* schedule. Let superscript C and N indicate whether a parameter or variable belongs to the current or the new schedule, respectively. The ships in the current schedule are contained in a set denoted $\mathcal{V}^C$, while the ships that should be planned for in the new schedule are contained in a set denoted $\mathcal{V}^N$. Furthermore, let $\mathcal{V}^A \subseteq \mathcal{V}^N$ represent the set of all ships that have called the operators and announced their arrival, but not yet been included in a schedule. I.e. all *announced*, unscheduled ships.

Recall from Section 4.2 the decision variables waiting time, entering segment time, traversing time and ship sequence w_{ip}, t_{ip}, s_{ip}, and z_{ijp}, respectively, and $\mathcal{P}_i$ being the set of all segments that ship i traverses during its journey. Let $\hat{w}_{ip}^C$, $\hat{t}_{ip}^C$, $\hat{s}_{ip}^C$ and $\hat{z}_{ijp}^C$ for all $i, j \in \mathcal{V}^C$ and all $p \in \mathcal{P}_i^C$ be the values assigned to the decision variables in the current schedule. Let w_{ip}^N, t_{ip}^N, s_{ip}^N and z_{ijp}^N for all $i, j \in \mathcal{V}^N$ and all $p \in \mathcal{P}_i^N$ be the decision variables in the new schedule.

The current simulation time is represented by τ, and CS_i denotes the current segment of ship $i \in \mathcal{V}^C$, i.e. the segment p ship i is positioned on at time τ. Recall from Section 4.2 that $\overline{P}_i$ is the exiting segment of ship i. The current segment of all ships in the canal can be defined as:

$$CS_i = \left\{ p \in \mathcal{P}_i^C \backslash \left\{ \overline{P}_i^C \right\} \mid \hat{t}_{ip}^C \leq \tau < \hat{t}_{ip+1}^C \right\} \qquad i \in \mathcal{V}^E$$
$$CS_i = \left\{ p \in \mathcal{P}_i^C \backslash \left\{ \overline{P}_i^C \right\} \mid \hat{t}_{ip}^C \leq \tau < \hat{t}_{ip-1}^C \right\} \qquad i \in \mathcal{V}^W$$

When $\tau \geq \hat{t}_{i\overline{P}_i}^C$, i.e. ship i has entered its exiting segment $\overline{P}_i$, the ship is considered as fully traversed, and is no longer of concern. The set of all fully traversed ships is denoted $\mathcal{V}^T$. As stated in Section 4.1, it is assumed that the exiting segment for each ship is the first or the last segment of the canal, depending on its direction. As both of these are sidings, conflicts are never the issue for a ship positioned on its exiting segment, and hence no scheduling is needed for this ship - it is categorized as fully traversed.

$\mathcal{V}^N$ is constructed as follows:

$$\mathcal{V}^N = \mathcal{V}^A \cup \left(\mathcal{V}^C \backslash \mathcal{V}^T \right)$$

In other words, the schedule made in the next roll includes all ships in the current plan, excluding the ones that are fully traversed, in addition to all ships that have announced their arrival since the last schedule was made.

As outlined above, the simulation-optimization framework requires a data set with a list of ships and a list of events. The list of events is initialized in the start of each simulation run, and needs some user input in order to be created. The term *simulator parameter* is used in the remainder of this thesis to represent these user inputs.

Recall that $T_{ETA,i}$ is the notation for ship i's estimated time of arrival. Let $T_{A,i}$ denote the announcement time of ship i. The framework requires an announcement time interval, $\left[\underline{\mathrm{T}}_{A,i}, \overline{\mathrm{T}}_{A,i}\right]$, of which $T_{A,i}$ must lie within. The interval is defined by two simulator parameters, denoted $\Delta\underline{\mathrm{T}}_A$ and $\Delta\overline{\mathrm{T}}_A$, where $\underline{\mathrm{T}}_{A,i}$ and $\overline{\mathrm{T}}_{A,i}$ both are functions of these simulator parameters.

Furthermore, let $T_{d,i}$ represent the time ship i is delayed with to the canal, and $T_{L,i}$ be the time at which it eventually arrives at the canal. In order to pre-generate the event of ships being delayed, the simulator must beforehand be given an amount of ships that should be affected by delay. This amount is determined as a fraction of the total number of ships in the problem instance, denoted f_d. Furthermore, a mean delay, represented by β, must be given as input in order to generate $T_{d,i}$ for all delayed ships. Both f_d and β are simulator parameters.

When a ship is slow on a segment, it arrives at the border to the next segment later than planned. The time at which it should have arrived at the border according to the current schedule is denoted $T_{S,ip}$, often referred to as slowness time. In the new plan, slow ships have a lower velocity on their respective slow segments. How much a ship's intended velocity is reduced by is bounded by a maximum velocity reduction, denoted $\overline{V}_{max}$. The low velocity a slow ship sails with in the new schedule is denoted $R_{i,p}^N$. Similarily to delay, a fraction f_s determines the amount of ships that should be affected by slowness. The segment(s) a ship is slow on is termed *slow segments*. The amount of slow segments ranges within an interval defined by two endpoints, denoted $\underline{SS}$ and $\overline{SS}$, where $\underline{SS} \leq \overline{SS}$. f_d, $\underline{SS}$ and $\overline{SS}$ are all simulator parameters.

A last simulator parameter needed is the simulation horizon, and is symbolized by T_{end}. An overview of all new introduced notation can be found in Appendix B.

5.3 Modelling of Events

A discrete-event simulation is utilized in order to model the DKCSTOP. When regarding events, a distinction is made between events related to the dynamism of the problem and events related to disruptions. In the following two subsections a description of the events incorporated in the simulator is given and, moreover, how they are modelled is explained.

5.3.1 Events Related to Dynamism

Two events related to the dynamism of the problem can be defined as:

1. *Arrival*: A new ship arrives at the canal
2. *Announcement*: A new ship calls the canal operators

Dynamic event 1, which is referred to as arrival, is the situation where a ship i arrives at the canal. Arrival takes place at the ship's estimated time of arrival, $T_{ETA,i}$. $T_{ETA,i}$ for each ship is provided by the initial data set.

Dynamic event 2, which is referred to as announcement, is equivalent to the real-world situation where the captain of ship i calls the canal operators and announces that they plan to traverse the canal. Announcement takes place at the ship's announcement time, $T_{A,i}$. $T_{A,i}$ needs to be calculated in a specific way in order to comply with the assumptions stated in Section 4.1. Specifically, it is contained in an interval, $T_{A,i} \in \left[\underline{\mathrm{T}}_{A,i}, \overline{\mathrm{T}}_{A,i}\right]$, so that the upper bound is smaller than the ship's arrival time. Using the simulator parameters $\Delta\underline{\mathrm{T}}_A$ and $\Delta\overline{\mathrm{T}}_A$ the interval is constructed as:

$$\left[\underline{\mathrm{T}}_{A,i}, \overline{\mathrm{T}}_{A,i}\right] = \left[T_{ETA,i} - \Delta\overline{\mathrm{T}}_A, T_{ETA,i} - \Delta\underline{\mathrm{T}}_A\right] \tag{25}$$

The announcement time is then drawn from a continuous uniform distribution with this interval as support.

$$T_{A,i} \sim \mathcal{U}\left(\underline{\mathrm{T}}_{A,i}, \overline{\mathrm{T}}_{A,i}\right) \tag{26}$$

$T_{A,i}$ is drawn for all ships in the data set in advance of the simulation.

5.3.2 Events Related to Disruptions

Two common events related to disruptions in the current schedule are:

1. *Delay*: A ship arrive at the canal later than planned
2. *Slowness*: A ship traverses a segment at a lower velocity than planned

Disruptive event 1, takes place due to unforeseen incidents prior to a ship i's arrival at the canal, causing it to arrive at a later point in time than the announced ETA. A delay becomes evident when the time τ is equal to $T_{ETA,i}$: Ship i either arrives as planned, or it does not show up. The time at which the delayed ship eventually arrives, $T_{L,i}$, represents the event of a delayed ship i. Just as $T_{A,i}$, $T_{L,i}$ is determined in advance of the simulation. Firstly, the fraction f_d determines the *amount* of ships in problem instance that are delayed. Secondly, *which* ships in the problem instance that are delayed must be determined. A number of ship equal to the amount of delayed ships are drawn from a discrete uniform distribution, where each ship has equal probability of being drawn. Recall from Section 3.2 that train delays can be assumed exponentially distributed. Therefore, $T_{d,i}$ is drawn from an exponential distribution with a mean delay β for all delayed ships. The remaining ships get a delay of zero, $T_{d,i} = 0$. Thus, $T_{L,i}$ is calculated as follows for all ships in the data set prior to simulation start:

$$T_{L,i} = T_{ETA,i} + T_{d,i}, \qquad T_{d,i} \sim \text{Exp}[\beta] \tag{27}$$

Figure 11 illustrates delay in a time-distance diagram.

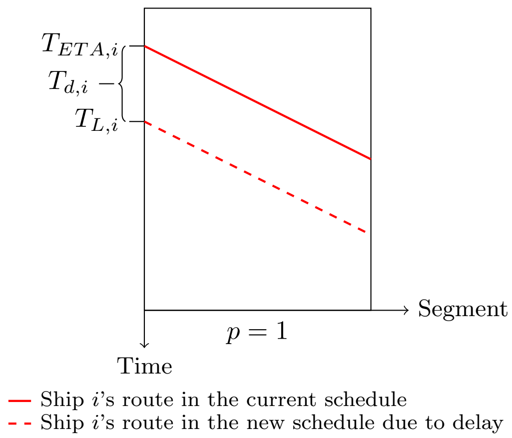


Figure 11: Disruptive event 1 (delay)

Disruptive event 2, slowness, occurs when a ship fails to keep the planned velocity on a segment and hence arrives at the following segment at a later point in time than what the current plan indicates. Just as for the delayed ships, the simulation-optimization framework requires a fraction, f_s, that determines the *amount* of ships that are slow. Thereafter, *which* ships in the problem instance that are slow must be determined. This is done by drawing from a particular distribution. This distribution is constructed as to incorporate the assumption that ships with large TGNs are more often slow than ships with small TGNs. The probabilities are created as follows:

$$\text{Prob}\,(i \text{ is slow}) = \frac{TGN_i}{\sum_{j \in \text{data set}} TGN_j} \tag{28}$$

Using these probabilities as a distribution, the appropriate number of ships are drawn from the data set. Furthermore, the slow segments for each slow ship must be determined. These are drawn as random variables from a discrete uniform distribution, with support $[SS_l, SS_u]$

It is assumed that the maximum velocity reduction for slow ships is $\overline{V}_{\max} = 6$ km/h $= 100$ m/min. This is equivalent to a 50% decrease in maximum velocity for ships with $TGN \geq 5$ and a 40% decrease for ships with $TGN < 5$. Recall that V_i is the speed limit of ship i, and let R^N_{ip} denote is the reduced velocity of i on segment p. The low traversing velocity for a ship i which is slow on a segment p is calculated by:

$$R^N_{ip} = V_i - \frac{X_{ip}}{10} \cdot \overline{V}_{\max}, \qquad X_{ip} \sim \mathcal{U}(1, 10) \tag{29}$$

Where X_{ip} is a random unitless number, drawn from a discrete uniform distribution. This calculation is repeated until all ships affected by slowness have received a reduced velocity on all their slow segments. Which ships that are slow, their respective slow segments and the reduction in velocity are all generated in advance of the simulation.

Slowness for a ship i on a segment p becomes evident when the simulation time τ is equal to the planned arrival at the border of the following segment for ship i, denoted $T_{S,ip}$. $T_{S,ip}$ represents the event of a ship i being slow on segment p. Unlike the other parameters related to slowness, it is not possible to generate $T_{S,ip}$ prior to the simulation, since it requires certain decision variables to hold values, and these values only exist for ships in the current schedule. Specifically:

$$T_{S,ip} = \hat{t}^C_{ip+1} - \hat{w}^C_{ip}, \qquad i \in \mathcal{V}^E \tag{30}$$

$$T_{S,ip} = \hat{t}^C_{ip-1} - \hat{w}^C_{ip}, \qquad i \in \mathcal{V}^W \tag{31}$$

I.e., $T_{S,ip}$ is the entering segment time of the following segment for a slow ship i, less any potential waiting on p. The next slowness event is determined by finding the minimum among the existing $T_{S,ip}$:

$$T_S = \min_{\substack{i \in \mathcal{V}^C \\ p \in \mathcal{P}_i}} \{T_{S,ip}\} \tag{32}$$

This calculation is done whenever replanning is triggered, regardless of event. This is due to the values assigned to the decision variables in the current schedule used in Equations (30)-(31) may be altered in the new schedule, implying that which particular combination of slow ship and segment that will happen first may change after replanning. Figure 12 illustrates slowness and when it occurs, in addition to how a ship's scheduled path changes when the velocity is reduced.

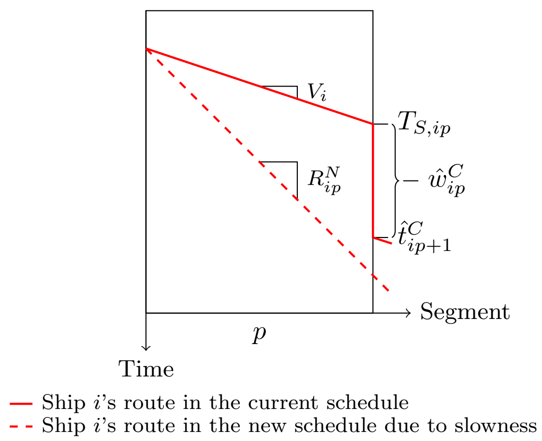


Figure 12: Disruptive event 2 (slowness)

5.4 Replanning Procedure

In this section, the replanning procedure is described in detail. When events presented in the previous section occur, the current schedule may become sub-optimal or infeasible. Thus, replanning is in some cases required. Replanning is a disruption management strategy, allowing for a new, optimal (or near-optimal) schedule to be made and displace the current one. Every time replanning is triggered, the decision maker can choose to re-use parts of the current schedule, or to create a completely new one. The former has a lower computational time compared to the latter, but the latter may yield a higher solution quality. In the framework developed, re-optimizing without recycling parts of the current schedule is done in every replanning procedure, except when slowness is the trigger. Common for all events is that the replanning procedure involves solving a constrained version of the KCSTOP.

When replanning is triggered by announcement, arrival or delay, the ship in question has not yet entered the canal. Therefore, it is assumed that the operators can afford to wait some minutes for a full regeneration of the schedule. Slowness is distinctive from the other events as it is the only event that may occur several times for the same ship. Furthermore, slowness occurs while the ship is located within the canal, contrary to all other events that occur prior to a ship's arrival. Both of these facts enlighten the benefit of obtaining a new schedule quickly when slowness is the replanning trigger, and thus justifies the use of recycling.

How replanning with and without recycling is implemented is explained in Sections 5.4.1-5.4.2. The reader is referred to Sections 5.2 and 4.2 for all new and old notation, respectively.

5.4.1 Replanning Without Recycling

A replanning procedure without recycling is triggered by delay, and in some cases by announcement and arrival. This version of replanning re-uses no parts of the current schedule. When the procedure is triggered by delay, it serves as a real-time disruption management strategy. When announcement and arrival are the triggers, the procedure acts as a method for handling dynamism. This subsection explains how replanning without recycling is implemented in the framework.

Let replanning be triggered at time τ. Before a new schedule is created, it is crucial that certain fixations are added to the mathematical model from Section 4.3. This to ensure continuity between the current and the new schedule. Continuity in this context means that a ship should be at the exact same position at time τ in both schedules. Furthermore, the fixations ensure that no decisions made before time τ are altered.

The procedure starts by finding the current segment CS_i^C for all ships in current plan, $i \in \mathcal{V}^C$. Recall $\underline{P}_i$ being the notation for the segment a ship i starts its route at. The starting segment for each ship in the current schedule which also is planned for in the new one is set to its current segment:

$$\underline{P}_i^N = CS_i^C, \qquad i \in \mathcal{V}^C \cap \mathcal{V}^N \tag{33}$$

For ships that have announced their arrival and thus should be planned for, but which are not in the current plan, i.e. ships in $\mathcal{V}^A$, their starting segment $\underline{P}_i^N$ is set to their entering segment according to the data set. In the making of a new schedule at time τ, no decisions made before this time should be altered, and the decisions to be made for each ship should only yield for the segments they not yet have traversed. The fixing of starting segment ensures this, since the optimizer only considers segments from each ship's starting segment and outwards when solving the KCSTOP.

The following constraints for entering segment times are added to the mathematical model:

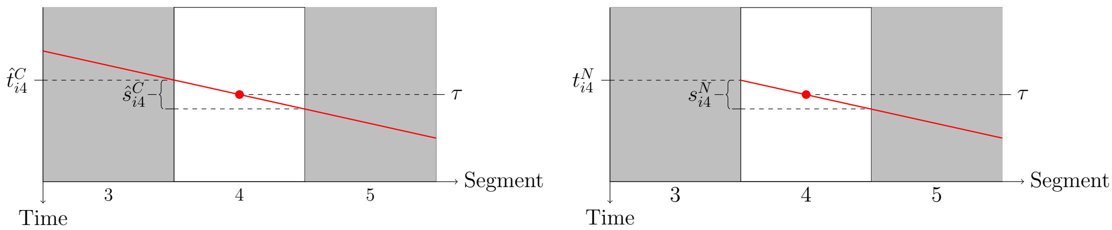
$$t^N_{i\underline{P}^N_i} = \hat{t}^C_{i\underline{P}^N_i}, \qquad i \in \mathcal{V}^C \cap \mathcal{V}^N \tag{34}$$

The same is done for traversing times:

$$s^N_{i\underline{P}^N_i} = \hat{s}^C_{i\underline{P}^N_i}, \qquad i \in \mathcal{V}^C \cap \mathcal{V}^N \tag{35}$$

The fixations in (34) and (35) ensure that every single ship is at the exact same position in the new schedule as they are in the current schedule at time τ.

All fixations can be visualized in a time-distance diagram, shown in Figure 13. Note that only a section of the canal is illustrated. Let ship i sail according to the current schedule, represented by the red line in Figure 13a. When replanning is triggered at time τ, the ship is positioned in the middle of segment 4, represented by the dot, and hence its current segment is four, $CS^C_i = 4$. According to (33), its starting segment in the new schedule is then $\underline{P}^N_i = 4$. When resolving the constrained version of the KCSTOP, only the following segments will be considered for ship i, i.e. $p = \left\{4, ..., \overline{P}^N_i\right\}$. The ship's position must be at the exact same spot in the new schedule in order to ensure continuity. In other words, the slope of the line representing the ship's route must be fixed so that it intersects the dot in both schedules at time τ. This is done by fixing the entering segment time and traversing time variables in the new schedule for this particular ship on this particular segment to the values they hold in the current schedule, i.e. $t^N_{i4} = \hat{t}^C_{i4}$ and $s^N_{i4} = \hat{s}^C_{i4}$. This is shown in Figure 13b.



(a) Initial schedule

(b) New schedule after replanning

Figure 13: Fixation of entering segment time and traversing time in the re-planning procedure

Additionally, waiting times must be fixed. This is done by checking if ships are currently waiting at a siding at time τ. If this is the case, the following constraints are added:

$$w^N_{i\underline{P}^N_i} \geq \tau - \hat{t}^C_{i\underline{P}^N_i} - \hat{s}^C_{i\underline{P}^N_i} \qquad i \in \mathcal{V}^C \cap \mathcal{V}^N \tag{36}$$

Note that one cannot simply fix the waiting time in the new schedule to the waiting time in the current schedule. The time τ may be in the middle of ship i's scheduled waiting in the current plan, and the optimal waiting in the new plan might be of different size than $\hat{w}^C_{iCS_i}$. However, it must at least be equal to, or greater than, what the ship has waited up until τ. See Figure 14 for a visualization of constraint (36). The red line indicates the route of ship i in the current schedule, and the dot is its position when replanning is triggered at time τ.

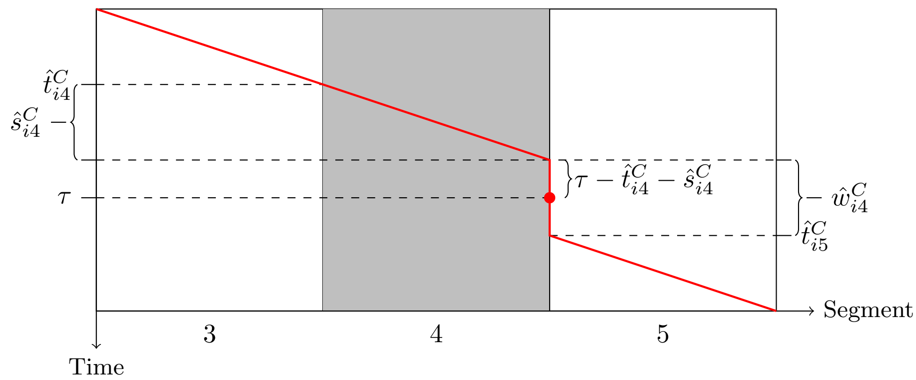


Figure 14: Fixation of waiting time in the re-planning procedure

The fixations stated in constraints (33)-(36) ensure continuity between the two schedules, and that no earlier decisions can be altered. With these constraints added to the model presented in Section 4.3, the problem is solved for all ships $i \in \mathcal{V}^N$, yielding a new schedule, and the current plan is now the new one found. The last activity in the replanning procedure is to empty the set $\mathcal{V}^A$ and set the set of ships in current plan to the set of ships in new plan, $\mathcal{V}^C = \mathcal{V}^N$.

5.4.2 Replanning with Recycling

Replanning with recycling is the strategy implemented in the framework in order to handle disruptions caused by slowness. The procedure involves re-using the ship sequence variables from the current schedule.

Let ship i be slow on one or several segments. A new plan is made by resolving the KCSTOP with all ship sequence variables, z_{ijp}, fixed to the values assigned to them in the current schedule, and moreover, modifying the velocity of ship i on the segment(s) it is slow at. The former fixation is implemented by:

$$z_{ijp}^N = \hat{z}_{ijp}^C, \qquad \{i,j\} \in \mathcal{V}^C \cap \mathcal{V}^N, \quad p \in \mathcal{P}_i \tag{37}$$

The latter modification is implemented by fixing the traversing time variable in the new plan to the slow traversing time imposed on ship i on segment p:

$$s_{ip}^N = \frac{L_{p+1} - L_p}{R_{ip}^N} \tag{38}$$

Where L_p is the cumulative length of the canal up until segment p, and R_{ip}^N is the slow traversing velocity the ship should use on segment p according to equation (29). The continuity fixations stated in Constraints (33)-(36) are also added to the model.

There exists two special cases of slowness where exceptions to the fixation rules are made. These are explained in the following.

Slowness for Aligned Ships

The first special case is associated with overtaking and safety distance between aligned ships. If the foremost of two consecutive, aligned ships is slow, the rearmost may catch up with it, and must slow down in order to comply with the safety distance rule. Figure 15 illustrates the case. Let eastbound ship i be slow on transit p so that its arrival at the next segment in the new schedule is $t^N_{ip+1} = \hat{t}^C_{ip} + s^N_{ip}$, where s^N_{ip} is fixed according to equation (38). Further, let ships i and j be aligned. The special case takes place if: 1) ship j starts to traverse segment p at a later point in time than i in the current schedule: $\hat{t}^C_{jp} > \hat{t}^C_{ip}$ and 2) the safety time constraints (14)-(15) are not satisfied for the two ships on segment $p+1$ if all fixations are implemented according to (33)-(38), i.e., $t^N_{ip+1} > \hat{t}^C_{jp+1} - H_{ij}$. The dashed red line in Figure 15 represents the slow traversal route for ship i, and *is* fixed according to Equation (38). This line crosses the blue solid line representing the current scheduled route for ship j, which *should* be fixed according to Constraint (35). A fixation of both these lines do not comply with the safety time constraints. Thus, the fixation for j on p according to Constraint (35) is not implemented. In the new schedule ship j rather follows the dashed blue line.

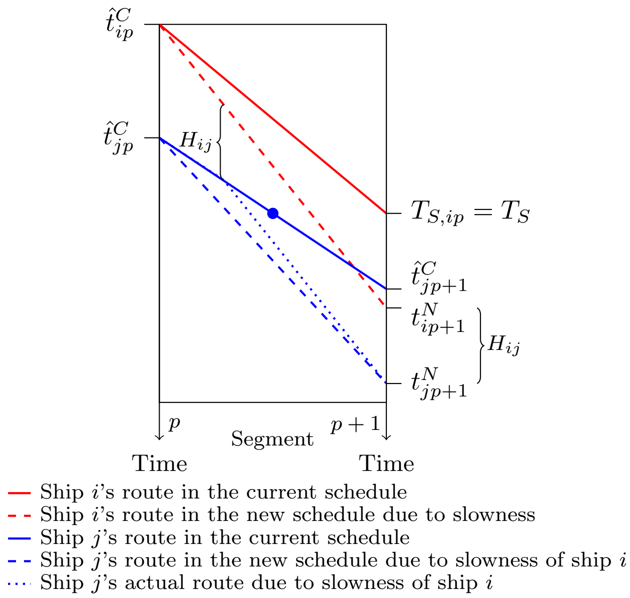


Figure 15: Special slowness case 1: Slowness for aligned ships

From a modelling point of view, continuity in ship j's position is not maintained when s^N_{jp} is free. According to continuity fixations, the blue dashed line representing ship j's route in the new schedule should have intersected the blue dot representing ship j's position at time $\tau = T_S$ in the current schedule. However, this discontinuity can be justified due to the modelling of slowness is by backtracking. The event does not become evident before the time reaches $\tau = T_S$, but the slow ship has slow ever since $\hat{t}^C_{ip}$, and hence been following the red dashed line up until time τ. Furthermore, ship j has been following the blue solid line until the safety distance to ship i is reached. Then, it slows down so that it reaches the next segment at the same time of which the blue dashed line does. This route is illustrated by the blue dotted line. Since the framework operates with traversing times for whole segments at a time, the blue dotted line and the blue dashed line are equally good solutions with respect to the objective function, since they start and end at the same time (determined by the safety distance to i).

This special case does not only apply to ships sailing directly behind slow ships. If the canal is particularly congested, it might happen that several ships are able to catch up with the slow ship. Since overtaking is not allowed on transits, these ships will then queue up and follow each other to the next border. Thus, the fixations according to Constraints (35) are not implemented for any of the ships that would defy the safety time constraints (14)-(15) if they were implemented.

Scheduled Velocity vs. Slowness Velocity
Another special case of slowness where exceptions to the fixation rules are made, is associated with a ship that is already sailing slower than its speed limit. If the ship is already sailing slower than the pre-determined velocity reduction, equation (38) is changed to:

$$s_{ip}^{N} = \frac{L_{p+1} - L_p}{V_{ip}^{C}} \tag{39}$$

Figure 16 illustrates this special case of slowness. Ship i is scheduled to sail with velocity V_{ip}^{C}, where $V_{ip}^{C} < R_{ip}^{N}$. Thus, at time $\tau = T_S$, the velocity of ship is not updated to R_{ip}^{N}. Instead, the velocity is unchanged and all the other fixations are applied as normal.

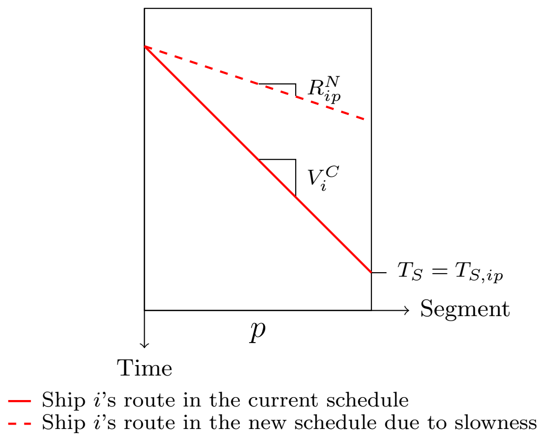


Figure 16: Special Slowness case 2: Scheduled velocity vs. slowness velocity

5.4.3 Summary

Replanning is triggered due to some event happening at time τ. In the event of delay, and in some cases of announcement and arrival, replanning without recycling is conducted. This means that a full regeneration of the schedule is done, by solving a constrained version of the KCSTOP. The constrained version contains fixations that ensure continuity between the current and new schedule, and avoid decisions made before time τ to be altered. When replanning is triggered by slowness, the ship sequence variables from the current schedule are recycled. Thus, the KCSTOP, with all continuity fixations and integer variables being fixed, is re-optimized, and the current order of ships is retained.

5.5 Composite Solution to the DKCSTOP

When solving the DKCSTOP with a rolling horizon scheduling procedure, the solution from each roll must be saved in order to compose the composite solution after the simulation horizon has elapsed. In every roll after replanning, the decisions made between the last roll and time τ are saved for each ship in the current plan. I.e., the values for entering segment time, traversing time and waiting time, $\hat{t}_{ip}^C$, $\hat{s}_{ip}^C$ and $\hat{w}_{ip}^C$, respectively. The segments these are saved for are all segments between and including the segment a ship started its journey at in the current plan, i.e. its starting segment, and its current segment. The process of saving decisions can be illustrated by time-distance diagrams. Figure 17 shows time-distance diagrams for an example simulation run. The first schedule is the initial schedule, obtained at time $\tau = 0$. Then replanning is triggered at time $\tau = 154$, and the second schedule is obtained. The second schedule considers the ships as if they started their voyage from their current segment at time $\tau = 154$. It is therefore crucial that all decision variables belonging to the first four ships are saved for the segments these ships have traversed between $\tau = 0$ and $\tau = 154$. The process of saving variable is repeated at time $\tau = 324$. Finally, at time $\tau = 850$ the simulation has ended and the decision variables from the last schedule is saved. Using this procedure, it is possible to draw the path each ship followed through the canal. This is constitutes a solution to the DKCSTOP, and is shown in Figure 18. The decisions made in each temporary schedule do not consider the events that are included in the next roll. Thus, it is unlikely that the saved variables are optimal in the composite schedule. Nevertheless, it is throughout this thesis assumed that good composite solutions are created from optimal or near optimal temporary schedules.

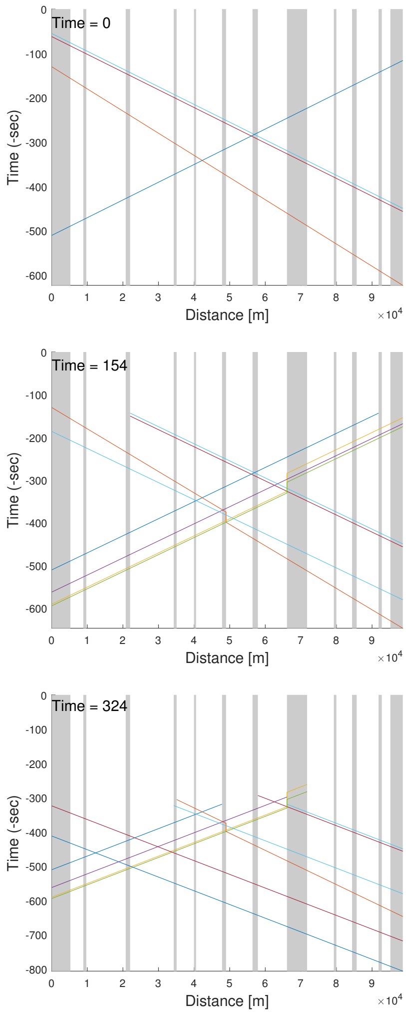


Figure 17: Example of solutions to constrained versions of the KCSTOP

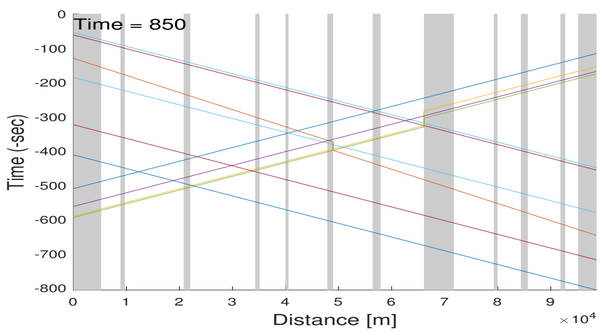


Figure 18: Example of a composite solution to the DKCSTOP

5.6 Flow Control of the Simulation-Optimization Framework

In this section, a thorough explanation of the entire simulation-optimization framework is given. This involves how the simulator implements the DKCSTOP and how it interacts with the optimizer, the link being the replanning procedure.

Before the simulation starts at τ_0, all events are generated as presented in Section 5.3 and the simulation horizon T_{end} is set, defining the event of termination. In addition, a data set containing ships and their belonging parameters and an initial schedule are required. The initial schedule is created by solving the KCSTOP for all ships where the announcement time is less than the starting simulation time, i.e., for all ships where $T_{A,i} < \tau_0$. When the simulation starts, the simulation time τ is incremented to the first event, and after the necessary computations have been made, the simulation time increments to the next event. This incrementation is done according to the following equation:

$$\tau = \min \{T_{A,i}, T_{ETA,i}, T_{L,i}, T_S, T_{end}\} \tag{40}$$

Whenever referring to an indexed event time, it is understood that there is only one ship i associated with the event. E.g., the $T_{L,i}$ included in the above equation is associated with the first delayed ship. Note that although the same subscript is used for several event times, the different events do not necessarily belong to the same ship. How the incrementation is carried out, and the implications of the different events, are depicted in the flow chart in Figure 19.

A review of the different events and their implications are given below. Unless otherwise stated, replanning refers to replanning without recycling.

1. Announcement is the next event, $\tau = T_{A,i}$. Ship i is added to the set of unscheduled, announced ships $\mathcal{V}^A$. There are two outcomes of this event.
   (a) If the number of ships in $\mathcal{V}^A$ exceeds the replanning threshold, $|\mathcal{V}^A| \geq \text{RT}$: Replanning is triggered, and the next $T_{A,i}$ and T_S are determined.
   (b) If the number of ships in $\mathcal{V}^A$ does not exceed the replanning threshold, $|\mathcal{V}^A| < \text{RT}$: The current plan remains, and the next $T_{A,i}$ is determined.
2. Arrival is the next event, $\tau = T_{ETA,i}$. There are four outcomes of this event.
   (a) If ship i is in the current plan, $i \in \mathcal{V}^C$, and arrives as announced: The current plan remains, and the next $T_{ETA,i}$ is determined.
   (b) If ship i is in the current plan, $i \in \mathcal{V}^C$, but does not arrive at $T_{ETA,i}$, i.e., it is late and will arrive at $T_{L,i}$: A new plan is made where ship i is excluded. This is done by removing i from $\mathcal{V}^C$, and triggering replanning. The next $T_{ETA,i}$ and T_S are then determined.
   (c) If ship i is not in the current plan, $i \notin \mathcal{V}^C$, and arrives at the canal as announced: Replanning is triggered, and the next $T_{ETA,i}$ and T_S are determined.
   (d) If ship i is not in the current plan, $i \notin \mathcal{V}^C$, and does not arrive at $T_{ETA,i}$, i.e., it is late: i should not be planned for before it eventually arrives at $T_{L,i}$, and is therefore removed from $\mathcal{V}^A$. The current plan remains, and a new $T_{ETA,i}$ is drawn.

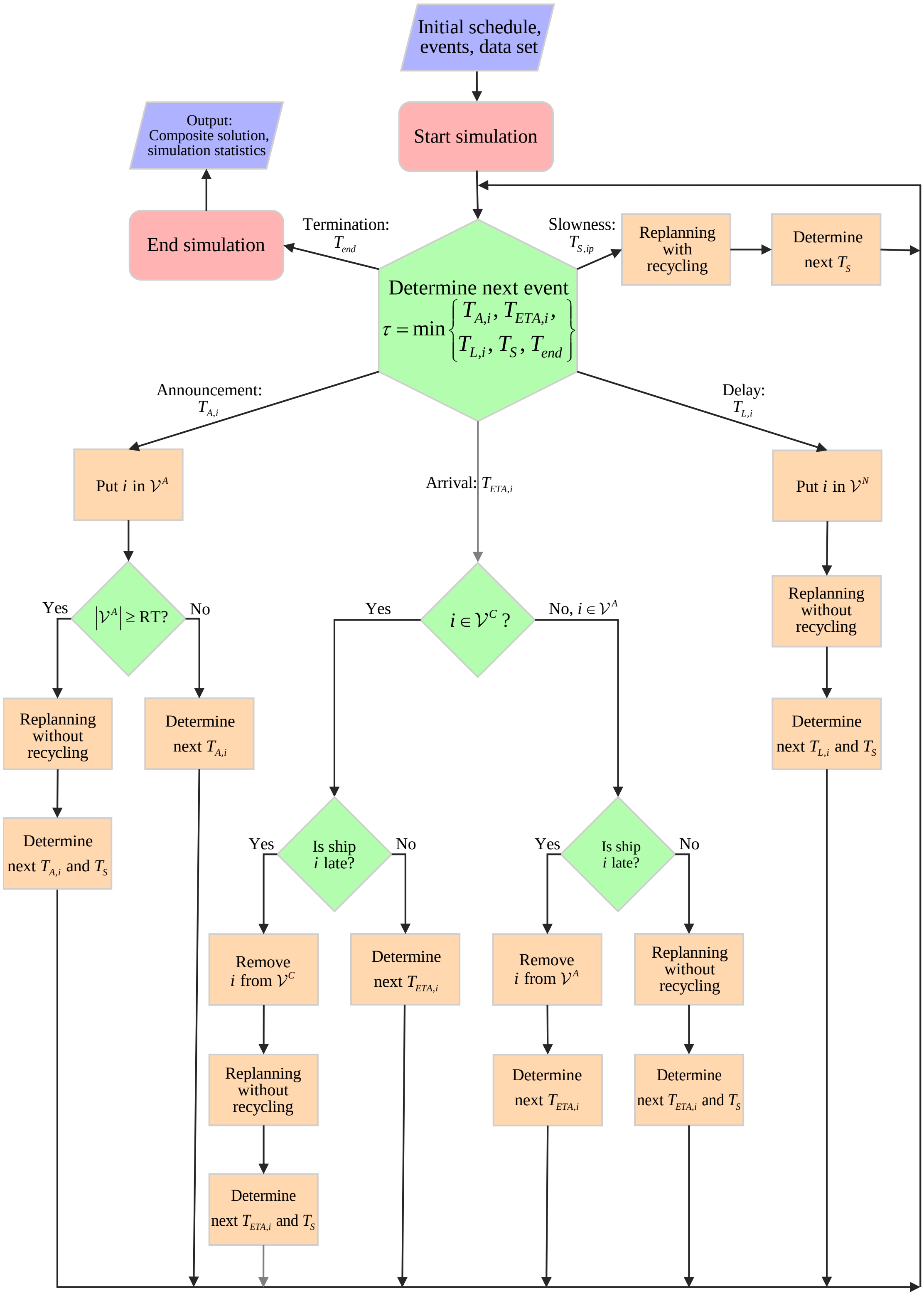


Figure 19: Flow chart of the simulation program

3. Delay is the next event, $\tau = T_{L,i}$. Ship i must now be accounted for and is added to $\mathcal{V}^N$. Replanning is triggered, and the next $T_{L,i}$ and T_S are determined.

4. Slowness is the next event, $\tau = T_S$. Recall that $T_S = \min_{i,p} \{T_{S,ip}\}$, and that the event implies that ship i is in the current plan, $i \in \mathcal{V}^C$. Replanning with recycling is triggered, and the next T_S is determined.

5. Termination is the next event, $\tau = T_{end}$. The simulation is ended, and the decision variables belonging to the last schedule are saved as described in the previous section. Together with the other decision variables that have been saved throughout the simulation, these make up the solution to the DKCSTOP. Statistics from the simulation-optimization are collected as outputs.

In order to better grasp how the simulator and optimizer interacts, an example of a simulation run with a sequence of events occurring is thoroughly explained in the following. The reader is referred to Figure 19 when reading the example simulation run in order to ease the understanding. Figure 20 shows the timeline for the example simulation run.

Example Simulation Run

0) Let $\mathcal{V}^C = \{1,2,3,4\}$, $\mathcal{V}^A = \{5,6\}$ and RT $= 3$.
Recall that $\mathcal{V}^N = \mathcal{V}^A \cup \left(\mathcal{V}^C \backslash \mathcal{V}^T\right)$

I) $\tau = T_{A,7}$ (event 1a)
Ship 7 calls the operators and announces its arrival. In addition, ship 1 is fully traversed.

$$\left.\begin{aligned} \mathcal{V}^A &= \{5,6,7\} \\ \mathcal{V}^C &= \{1,2,3,4\} \\ \mathcal{V}^T &= \{1\} \end{aligned}\right\} \Rightarrow \mathcal{V}^N = \{5,6,7\} \cup (\{1,2,3,4\} \setminus \{1\}) = \{2,3,4,5,6,7\}$$

Replanning is triggered for the ships in $\mathcal{V}^N$. $T_{A,i}$ is set to $T_{A,8}$, T_S is set to $T_{S,5,2}$, $\mathcal{V}^A$ is emptied and $\mathcal{V}^C$ is set to $\mathcal{V}^N$.

II) $\tau = T_{ETA,5}$ (event 2a)
Ship 5 which is in the current plan arrives at the canal as planned. The current plan remains, and $T_{ETA,i}$ is set to $T_{ETA,6}$.

III) $\tau = T_{ETA,6}$ (event 2b)
Ship 6 which is in the current plan does not arrive as planned, i.e., it is late, and removed from $\mathcal{V}^C$:

$$\mathcal{V}^C = \{2,3,4,5,6,7\} \setminus \{6\} = \{2,3,4,5,7\}$$

Thus,

$$\left.\begin{aligned} \mathcal{V}^A &= \{\} \\ \mathcal{V}^C &= \{2,3,4,5,7\} \\ \mathcal{V}^T &= \{1\} \end{aligned}\right\} \Rightarrow \mathcal{V}^N = \{\} \cup (\{2,3,4,5,7\} \setminus \{1\}) = \{2,3,4,5,7\}$$

Replanning is triggered for the ships in $\mathcal{V}^N$. $T_{ETA,i}$ is set to $T_{ETA,7}$, $\mathcal{V}^C$ is set to $\mathcal{V}^N$ and $T_{S,5,2}$ is updated with the decision variables from the new schedule.

IV) $\tau = T_S = T_{S,5,2}$ (event 4)
Ship 5 traverses segment 2 slower than intended in the current plan. In addition, ship 2 is fully traversed:

$$\left.\begin{aligned} \mathcal{V}^A &= \{\} \\ \mathcal{V}^C &= \{2,3,4,5,7\} \\ \mathcal{V}^T &= \{1,2\} \end{aligned}\right\} \Rightarrow \mathcal{V}^N = \{\} \cup (\{2,3,4,5,7\} \setminus \{1,2\}) = \{3,4,5,7\}$$

Replanning with recycling is triggered. $\mathcal{V}^C$ is set to $\mathcal{V}^N$ and $T_{S,ip}$ is emptied as no ships in the current plan are slow.

V) $\tau = T_{L,6}$ (event 3)
Ship 6 arrives at its delayed time of arrival, and is added to $\mathcal{V}^N$:

$$\left.\begin{aligned} \mathcal{V}^A &= \{\} \\ \mathcal{V}^C &= \{3,4,5,7\} \\ \mathcal{V}^T &= \{1,2\} \end{aligned}\right\} \Rightarrow \mathcal{V}^N = \{\} \cup (\{3,4,5,7\} \setminus \{1,2\}) \cup \{6\} = \{3,4,5,6,7\}$$

Replanning is triggered for all ships in $\mathcal{V}^N$. $T_{L,i}$ is set to $T_{L,9}$, $\mathcal{V}^C$ is set to $\mathcal{V}^N$ and T_S is still empty as no ships in the current plan are slow.

VI) $\tau = T_{A,8}$ (event 1b)
Ship 8 calls the operators and announces its arrival.

$$\mathcal{V}^A = \{8\}$$

The replanning threshold is not exceeded and the current plan remains. $T_{A,i}$ is set to $T_{A,9}$.

VII) $\tau = T_{ETA,7}$ (event 2a)
Ship 7 which is in the current plan arrives at the canal as planned. The current plan remains, and $T_{ETA,i}$ is set to $T_{ETA,8}$.

VIII) $\tau = T_{ETA,8}$ (event 2c)

Ship 8 which is not in the current plan arrives at the canal as planned.

$$\left.\begin{aligned}\mathcal{V}^A &= \{8\}\\ \mathcal{V}^C &= \{3,4,5,6,7\}\\ \mathcal{V}^T &= \{1,2\}\end{aligned}\right\} \Rightarrow \mathcal{V}^N = \{8\} \cup (\{3,4,5,6,7\} \setminus \{1,2\}) = \{3,4,5,6,7,8\}$$

Replanning is triggered for the ship in $\mathcal{V}^N$. $T_{ETA,i}$ is set to $T_{ETA,9}$, $T_{S,ip}$ is set to $T_{S,8,14}$, $\mathcal{V}^A$ is emptied and $\mathcal{V}^C$ is set to $\mathcal{V}^N$.

IX) $\tau = T_{A,9}$ (event 1b)
Ship 9 calls the operators and announces its arrival.

$$\mathcal{V}^A = \{9\}$$

The replanning threshold is not exceeded and the current plan remains. $T_{A,i}$ is set to $T_{A,10}$.

X) $\tau = T_{ETA,9}$ (event 2d)
Ship 9 which is not in the current plan does not arrive as planned, i.e. it is late, and removed from $\mathcal{V}^A$:

$$\mathcal{V}^A = \mathcal{V}^A \setminus \{9\} = \{\}$$

The current plan remains, and $T_{ETA,i}$ is set to $T_{ETA,10}$.

XI) $\tau = T_{end}$ (event 6)
The simulation horizon is reached, and the simulation is terminated.

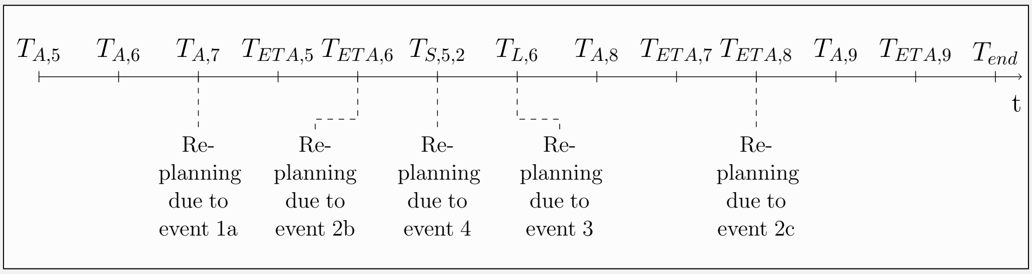


Figure 20: Timeline of events in the example simulation run

6 Computational Study

In this chapter, a computational study is presented. How the experimental environment is designed is described in Section 6.1. The results from the experiments are presented in the next four sections presents. In Section 6.2, the effect delay and slowness have on the solution to the DKCSTOP is examined. Since both disruptions are present in a real-life planning situation, the ramifications of combining disruptive events are studied in Section 6.3. In Section 6.4, the performance of the weighted approach proposed in Section 4.4 is evaluated, while Section 6.5 contains a discussion on different disruption management configurations.

6.1 Experimental Environment

In this section, the experimental environment is described. Firstly, the approach and testing environment is reported in Subsection 6.1.1. Secondly, the problem instance and the different scenarios are described in Subsection 6.1.2. Thirdly, the performance indicators used throughout the computational study are listed in Subsection 6.1.3. Finally, the choice of optimization termination time is justified in Subsection 6.1.4.

6.1.1 Approach and Testing Environment

The data used for the computational study is provided by the canal operators. It includes a list of ships that traversed the canal during one month of operation, with their respective parameters such as direction, ETA, traffic group number and entering segment. The data set only contains ships that fully traversed the canal that month, i.e. the entering segment is 1 for eastbound ships, and 23 for westbound ships. In addition to the data set, the exact topology of the canal is provided. The topology specifies the length and passage number for all segments. Figure 21 shows the canal topology of the Kiel Canal which is used for all experiments.

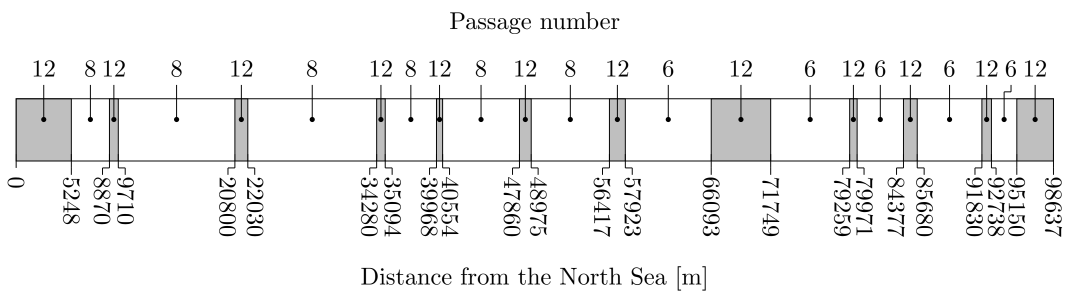


Figure 21: Kiel Canal topology

Each simulation is done with a simulation horizon of 10080 simulation minutes, corresponding to one week of operation, and an optimization termination time of five minutes. The framework is coded in MATLAB. The commercial optimization software Xpress is used to solve the KCSTOP, and an Xpress-MATLAB interface is implemented in order to call the optimizer directly from MATLAB. All specifications of the computer and commercial software used is stated in Table 1.

Processor	Intel Core i7-4790S (3.20GHz) CPU
Ram	16 GB
Operating system	Windows 7 64-bit
MATLAB version	R2018a 64-bit
Xpress-IVE version	1.24.18 64-bit
Xpress Mosel version	4.6.0
Xpress Optimizer version	31.01.09

Table 1: Software and hardware used in the computational study

6.1.2 Problem Instances and Scenarios

With a simulation horizo of one week, the problem instance consists of the ships belonging to the first week of the data set. The problem instance is tested under different scenarios. A scenario is a specific combination of a replanning threshold (RT) and simulation parameters, which are summarized in Table 2.

Event	**Simulator parameter**	**Explanation**
Announcement	$\Delta \underline{T}_A$	Decides upper endpoint of the announcement time interval
	$\Delta \overline{T}_A$	Decides lower endpoint of the announcement time interval
Delay	f_d	Fraction of the ships in the problem instance that are delayed
	β	Mean delay
Slowness	f_s	Fraction of the ships in the problem instance that are slow
	$\underline{SS}$	Minimum number of slow segments
	$\overline{SS}$	Maximum number of slow segments
Termination	T_{end}	Simulation horizon

Table 2: Simulation parameters summarized

A specific combination of the simulation parameters associated with delay and slowness constitute different cases which are studied in the computational study. *Base case* is the case with ships continually announcing their arrivals and arriving at the canal; *delay case* is the base case including delay as disruptive event; *slowness case* is the base case including slowness as disruptive event and *combined case* is the base case with both delay and slowness as disruptive events. All cases except base case are further divided into cases with low and high occurrence of events. BC, DL, DH, SL, SH, CL and CH denote base case, delay case low, delay case high, slowness case low, slowness case high, combined case low and combined case high, respectively. One of these cases, together with a RT and an announcement time interval ($T_{A,i}$-interval), constitute a scenario. Each scenario is given a name. E.g, a scenario named "RT1_AT24_BC" has a RT equal to 1, $T_{A,i}$-interval of 2-4 hours, and is a base case.

All scenarios are generated in advance and used as input to the simulator. This means that no events are randomly generated during the simulation run. This allows for a comparison of different scenarios. Due to the lack of historic data regarding delay and slowness, simulation parameters that seems plausible in a real-world situation are chosen in order to generate scenarios. Table 3 shows the combination of simulation parameters for the different cases, and for an $T_{A,i}$-interval of 2-4 hours.

Cases	f_d	β	f_s	$[SS_l, SS_u]$
Base case	0	0	0	[0, 0]
Delay case low	10%	20 min	0	[0, 0]
Delay case high	20%	40 min	0	[0, 0]
Slowness case low	0	0	20%	[1, 2]
Slowness case high	0	0	50%	[3, 5]
Combined case low	10%	20 min	20%	[1, 2]
Combined case high	20%	40 min	50%	[3, 5]

Table 3: Simulation parameters for different cases with an $T_{A,i}$-interval of 2-4 hours

In order to pre-generate the announcement time, a $T_{A,i}$-interval is required. As far as we are concerned, today's practice is to require ship captains to announce their arrival within 4-6 hours beforehand. The operators are considering to extend this interval to 6-8 hours in order to accumulate more information regarding arriving ships. However, it is also of interest to analyze the effect of reducing the interval. Therefore, three $T_{A,i}$-interval are used to create scenarios: 2-4 hours, 4-6 hours and 6-8 hours prior ETA. In Section 6.3.2 the effect of either increasing or reducing the interval is analyzed.

It is assumed that the simulator parameters related to delay are linearly correlated with the $T_{A,i}$-interval. The rationale is that ships with early announcements are more likely to be delayed, and for this delay to be long. For instance, if a ship calls only two hour prior arrival, the delay cannot exceed this time span. Furthermore, the captain will at that point have a fairly well understanding of weather conditions and other circumstances that can affect its travel up until arrival, and hence be able to predict the arrival accordingly. If a ship calls in many hours prior expected arrival, many incidents can happen that induce delays, and these delays can potentially become lengthy. Thus, f_d and β are scaled with the $T_{A,i}$-interval according to Table 4.

	Announcement time interval					
	[2,4]		[4,6]		[6,8]	
Instance	f_d	β	f_d	β	f_d	β
Delay case low	10%	20 min	15%	30 min	20%	40 min
Delay case high	20%	40 min	30%	60 min	40%	80 min
Combined case low	10%	20 min	15%	30 min	20%	40 min
Combined case high	20%	40 min	30%	60 min	40%	80 min

Table 4: Delay simulation parameters scaled for different $T_{A,i}$-intervals

Replanning on every ship announcement or every fifth announcement are considered as the two extremities for the RT. The latter implies that five ships must call in between two consecutive arriving ships if the RT should trigger replanning. Due to the frequency of ship arrivals, the RT will hardly trigger replanning if it is set higher than 5. Therefore, the three RTs used in the computational study are 1, 3 and 5. Unlike the choice of $T_{A,i}$-interval which changes the delay simulation parameters, the choice of RT does not change any of the simulation parameters.

6.1.3 Performance Indicators

Several performance indicators must be considered when evaluating solutions to the DKCSTOP. The indicators only account for ships that have fully traversed the canal. In other words, only ships that are contained in $\mathcal{V}^T$ at time T_{end} are accounted for.

Avg. time denotes the average travelling time per fully traversed ship, and is the total travelling time divided by the amount of fully traversed ships:

$$\text{Avg. time} = \frac{\sum_{i \in \mathcal{V}^T} \hat{t}_{i\overline{P}_i} + \hat{s}_{i\overline{P}_i} - T_{D,i}}{|\mathcal{V}^T|} \tag{41}$$

Note that $T_{D,i}$ is used in order to not incorporate delay in the average traversing time.

Avg. waiting denotes the average waiting time per fully traversed ship, and is the total waiting time divided by the amount of fully traversed ships:

$$\text{Avg. waiting} = \frac{\sum_{i \in \mathcal{V}^T} \sum_{p \in \mathcal{S}_i} \hat{w}_{ip}}{|\mathcal{V}^T|}, \tag{42}$$

Max. waiting denotes the maximum single waiting time observed among fully traversed ships.

$$\text{Max. waiting} = \max_{\substack{i \in \mathcal{V}^T \\ p \in \mathcal{P}_i}} \{\hat{w}_{ip}\} \tag{43}$$

Max. tot. waiting denotes the maximum total waiting time observed among fully traversed ships.

$$\text{Max. tot. waiting} = \max_{i \in \mathcal{V}^T} \left\{ \sum_{p \in \mathcal{P}_i} \hat{w}_{ip} \right\} \tag{44}$$

For scenarios where slowness occurs, the average traversing time per ship is necessarily higher than for the base case scenarios. This is due to several ships being fixed to a lower velocity on one or more segments. Thus, average traversing time is not a fair performance indicator for slowness and combined cases. An adjusted average traversing time is therefore used, denoted *Adj. avg. time*:

$$\text{Adj. avg. time}_{\text{x}} = \text{Avg. time}_{\text{BC}} + (\text{Avg. waiting}_x - \text{Avg. waiting}_{\text{BC}}) \tag{45}$$

Where x is one of the aforementioned cases.
The adjusted average traversing time corrects for prolonged traversing times due to slowness events. However, it also corrects for ships that happen to have a prolonged traversing time due to conflict resolution. Although this implies that the adjusted average traversing time has a downward bias, it is assumed fair as the majority of conflicts are resolved by waiting.

The simulator is initialized with an empty canal, meaning that the first few ships arriving at the canal experience an unusual low amount of conflicts. As a consequence their traversing times are lower than if the same ships were faced with a more busy canal. Thus, the ten first ships are excluded when calculating the performance indicators listed above.

Number of rolls is a performance indicator that reports the number of replanning procedures conducted in one simulation run. *Avg. comp. time* denotes the average computational time for each of these rolls. When solving the DKCSTOP over a scenario, the optimality gap in each roll is saved. The *Average optimality gap* is the average of all the the gaps from a simulation run. Note that the first ten rolls are considered as a start-up phase and excluded from the two aforementioned averages.

The purpose of the performance indicators is to compare and analyze different scenarios for which the DKCSTOP is solved over. The relative difference to the base case is often reported to ease the comparison of solutions. For a given performance indicator the relative difference is denoted ΔPerformance indicator, and calculated as:

$$\Delta\text{Performance indicator} = \frac{\text{Performance indicator}_x}{\text{Performance indicator}_{\text{BC}}} - 1 \qquad (46)$$

Where x is a case different from base case.

Often when a ΔPerformance indicator is used, a confidence level is also given. The confidence level is calculated from a paired t-test with the null hypothesis being that the ΔPerformance indicator is equal to zero. The confidence level is stated as a percentage, and a higher percentage means more confidence in the alternative hypothesis that the ΔPerformance indicator is different from zero. The terms *significant* and *insignificant* is used in order to state whether a given confidence level is above or below 95%, respectively. A thorough review of how the confidence level is calculated can be found in Appendix D.

6.1.4 Optimization Termination Time

As replanning happens quite frequently it is advantageous for both canal operators and ship captains that the process is fast. Thus, a maximum running time for the optimizer, which is called every time replanning is triggered, must be chosen. If it is set too low, it may compromise solution quality, but if it set too high, the replanning procedure becomes tedious. Some initial simulations are conducted with an optimization termination time of both 30 and 5 minutes. The DKCSTOP is simulated over the base case in combination with the different RTs and $T_{A,i}$-intervals, totalling nine scenarios.

Table 5 shows average amount of rolls solved to optimality in percentage within 30 and 5 minutes, and the reduction when decreasing the optimization termination time from 30 to 5 minutes. The ten first rolls are excluded due to the start-up phase. The overall average reduction is 15%, which is low considering that the execution time is reduced by 83%. Recall from Section 5.5 that solving more rolls to optimality does not necessarily result in better solutions. Therefore, having on average 15% more rolls being solved to optimality may not be beneficial in terms of solution quality. Thus, the optimization termination time is set to 5 minutes for the remainder of the computational study.

RT	$A_{T,i}$ -interval	Rolls solved to optimality within 30 minutes	Rolls solved to optimality within 5 minutes	Reduction in rolls solved to optimality
1	$[2, 4]$	97%	90%	7%
	$[4, 6]$	87%	67%	20%
	$[6, 8]$	63%	41%	21%
3	$[2, 4]$	97%	91%	6%
	$[4, 6]$	83%	67%	16%
	$[6, 8]$	58%	40%	19%
5	$[2, 4]$	98%	92%	7%
	$[4, 6]$	88%	67%	21%
	$[6, 8]$	56%	39%	18%
Case average		**81%**	**66%**	**15%**

Table 5: Percentage of rolls solved to optimality within 30 and 5 minutes

6.2 Effect of Disruptive Events

In this section, various performance indicators are compared in order to measure how disruptions affect the solution to the DKCSTOP. The effect of delay and slowness is examined in Subsection 6.2.1 and 6.2.2, respectively. Note that in all tables there is a slight discrepancy between percentages and actual differences due to rounding of the performance indicators.

6.2.1 Delay

Table 6 shows average traversing time for base case, delay case low and delay case high, for all combinations of replanning thresholds (RTs) and announcement times intervals ($A_{T,i}$-intervals).

		Base case	Delay case low			Delay case high		
RT	$A_{T,i}$ -interval	Avg. time (min)	Avg. time (min)	ΔAvg. time	Confidence level	Avg. time (min)	ΔAvg. time	Confidence level
1	$[2, 4]$	460	461	0.2%	51%	461	0.1%	20%
	$[4, 6]$	461	461	0.2%	37%	463	0.5%	76%
	$[6, 8]$	461	461	-0.1%	17%	461	0.0%	14%
3	$[2, 4]$	461	460	-0.2%	31%	460	-0.2%	34%
	$[4, 6]$	461	461	-0.1%	10%	460	-0.2%	17%
	$[6, 8]$	462	463	0.3%	55%	464	0.5%	66%
5	$[2, 4]$	462	461	-0.2%	33%	460	-0.4%	54%
	$[4, 6]$	458	461	0.6%	78%	462	0.7%	88%
	$[6, 8]$	462	465	0.6%	78%	463	0.3%	64%
Case average		**461**	**461**	**0.1%**	-	**462**	**0.2%**	-

Table 6: Average traversing time for base case, delay case low and delay case high

Observe that there is no significant difference between the base case and the delay cases when measured by average traversing time. In other words, delays do not affect the average traversing time, indicating that the disruption management strategy (replanning without recycling) is in fact successful. However, this nullifying of disruptions caused by delay comes at a cost. Recall from Section 5.4 that there exists a trade-off between computational time and solution quality. Further recall that replanning takes place both at the estimated and delayed arrival time of the ship. It is therefore of interest to examine just how many more schedules are generated in order to achieve this negation of delay disruptions.

A bar chart is depicted in Figure 22, and shows the number of rolls, i.e. number of schedules generated, in the delay cases compared to the base case. The number of rolls are averaged over both RT and $A_{T,i}$-interval.

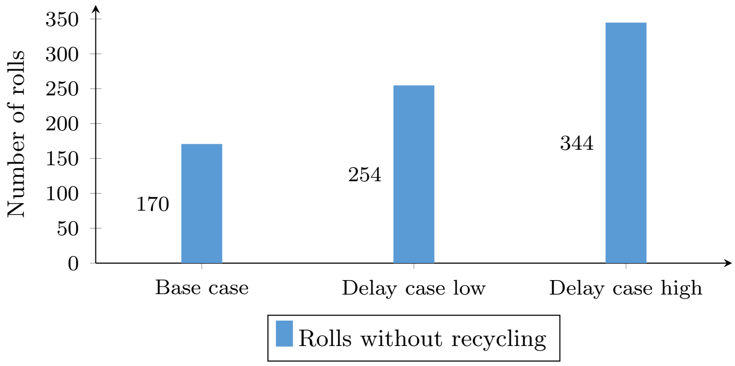


Figure 22: Number of rolls generated for base case, delay case low and delay case high, averaged over both RT and $A_{T,i}$-interval

It is apparent that the number of rolls increases with added delay events. Replanning takes place 170 times during one week of operation in the base case. With the current optimization termination time of five minutes, this is equivalent to spending maximum two hours a day[1] on computations. In comparison, the delay case high reschedules once every half hour and the delay case low every 40 minutes, resulting in maximum four and two hour a day being spent on computations, respectively. To double the computation time in order to only mitigate delays is quite severe. However, it is not known whether the disruptions caused could have been mitigated by spending less computational time. For instance, a replanning with recycling might have performed just as well. This is discussed further in Section 6.5.

6.2.2 Slowness

Table 7 shows adjusted average traversing time for slowness case low and slowness case high compared to the base case. The relative differences reveal that the solution quality is worsened in slowness case high, while for slowness case low most changes are insignificant. For both slowness cases, the relative difference in adjusted average traversing time is worst for scenarios RT5_AT46_SL and RT5_AT46_SH. Why these two in particular are worst case scenarios is most likely owing to the particular combination of pre-generated events, engendering more conflicts to resolve. It should be noted that RT5_AT46_SL is the only scenario with a significant relative change among the slowness low scenarios. The average worsening in adjusted average traversing time for slowness case high is 3.0%, while for the corresponding delay case no significant change was found. Thus, it can be stated that slowness scenarios have a lower solution quality than delay scenarios relative to base case. It should however be pointed out that slowness can occur several times for each ship contrary to delay, and hence this result is somewhat expected.

[1]One day is understood as 24 hours

RT	$A_{T,i}$ -interval	Base case Avg. time (min)	Slowness case low Adj. avg. time (min)	ΔAdj. avg. time	Confidence level	Slowness case high Adj. avg time (min)	ΔAdj. avg. time	Confidence level
1	[2, 4]	460	461	0.1%	51%	471	2.3%	100%
	[4, 6]	461	460	-0.1%	30%	476	3.3%	100%
	[6, 8]	461	463	0.3%	51%	474	2.9%	100%
3	[2, 4]	461	463	0.5%	80%	473	2.6%	100%
	[4, 6]	461	463	0.4%	64%	474	2.8%	100%
	[6, 8]	462	463	0.2%	46%	477	3.3%	100%
5	[2, 4]	462	463	0.4%	66%	473	2.5%	100%
	[4, 6]	458	464	1.3%	100%	475	3.8%	100%
	[6, 8]	462	465	0.7%	94%	478	3.4%	100%
Case average		**461**	**463**	**0.4%**	-	**475**	**3.0%**	-

Table 7: Adjusted average traversing time for base case, slowness case low and slowness case high

The disruption management strategy implemented in order to mitigate slowness is to replan with recycling every time slowness occurs. Fixing the ship sequence variables reduces the complexity of the constrained version of the KCSTOP drastically. However, this strategy may compromise solution quality, which the adjusted average traversing times suggest. Another observation is that slowness case high has a profoundly larger worsening than slowness case low. This is due to the two simulation parameters associated with slowness (the fraction of slow ships f_s and the amount of slow segments ranging between $[SS_l, SS_u]$) are both more than doubled in slowness case high. Thus, disproportionately more slowness events occur in the high case, resulting in several more replanning triggers. This is illustrated in Figure 23 where the number of rolls in slowness case high are a lot higher than slowness case low. Although the total number of rolls increase, the added rolls in both slowness case low and high are solved with recycling. This means that approximately all binary variables are fixed, and the problem in each roll is solved nearly instantaneously. With the current optimization termination time of five minutes, maximum two hours is on average spent on computations daily, and this does not increase with the number of slowness events.

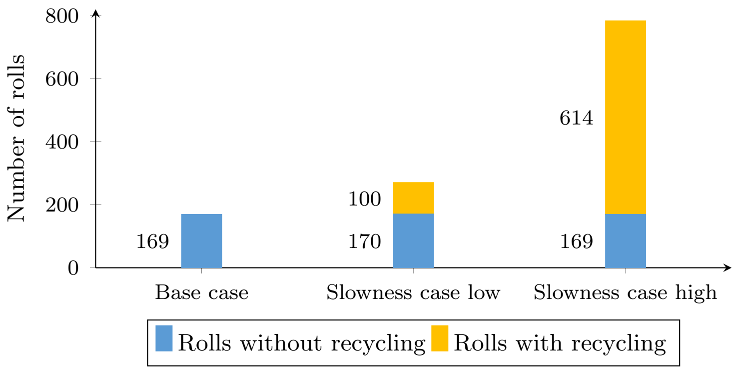


Figure 23: Number of rolls generated for base case, slowness case low and slowness case high, averaged over both RT and $A_{T,i}$-interval

The schedule is updated roughly every 15 minute on average in slowness case high. The corresponding replanning frequency for delay case high is every 30 minute. Furthermore, the average computational time used daily is two and four hours, respectively. Thus, in delay case high twice as much computational time is spent on generating half as many schedules compared to slowness case high. This illustrates how the framework prioritizes time when managing disruptions caused by slowness, and solution quality when managing disruption caused by delay.

6.3 Real-life Planning Situation

In this section, the ramifications of both disruptive events are examined. Firstly, the combined cases are studied in context of the base case in Subsection 6.3.1. The managerial decisions regarding RT and $A_{T,i}$-interval are scrutinized in Subsection 6.3.2. The section is concluded in Subsection 6.3.3 with a study of desirable features concerning waiting times for ships.

6.3.1 Combined Effect of Disruptive Events

Table 8 shows adjusted average traversing time for combined case low and and combined case high compared to the base case. As expected, average traversing time is estimated to be larger for the combined cases than for the base case. However, only the relative changes for the combined case high are significantly higher. Most scenarios in the combined case low have insignificant changes, but confidence levels are in general quite high. Recall that in the combined cases, both delays and slowness events occur. When delay happens, replanning without recycling is conducted and high solution quality is presumably obtained, but in return more computational effort is required. When slowness happens the opposite is likely to be true. At first it is conceivable that replanning without recycling could offset the less good solutions obtained from replanning with recycling. However, observe that the numbers are more similar in magnitude to the adjusted average traversing times for the slowness cases than the average traversing times for the delay cases. Thus, the replanning procedures carried out without recycling are not frequent enough to outbalance the compromised solution quality from replanning with recycling.

		Base case	Combined case low			Combined case high		
RT	$A_{T,i}$ -interval	Avg. time (min)	Adj. avg. time (min)	ΔAdj. avg. time	Confidence level	Adj. avg time (min)	ΔAdj. avg. time	Confidence level
1	$[2,4]$	460	463	0.6%	81%	470	2.1%	100%
	$[4,6]$	461	464	0.6%	85%	473	2.7%	100%
	$[6,8]$	461	465	0.9%	91%	474	2.8%	100%
3	$[2,4]$	461	464	0.7%	83%	473	2.5%	100%
	$[4,6]$	461	463	0.5%	71%	474	2.7%	100%
	$[6,8]$	462	464	0.5%	69%	477	3.3%	100%
5	$[2,4]$	462	463	0.2%	44%	471	2.0%	100%
	$[4,6]$	458	465	1.4%	100%	473	3.1%	100%
	$[6,8]$	462	466	0.9%	96%	475	2.8%	100%
Case average		**461**	**464**	**0.7%**	-	**473**	**2.7%**	-

Table 8: Adjusted average traversing time for base case, combined case low and combined case high

A stacked bar chart is depicted in Figure 24, and shows that the number of rolls is made up by summing the number of rolls with recycling in the slowness cases and the number of rolls in the delay cases. Thus, both the number of rolls with and without recycling are increased in the combined cases. This is expected since both slowness events and delays are added in the simulator. The consequence is an increase in computational effort in addition to a compromised solution quality. "The worst of both worlds" is obtained when combining the two disruptive events.

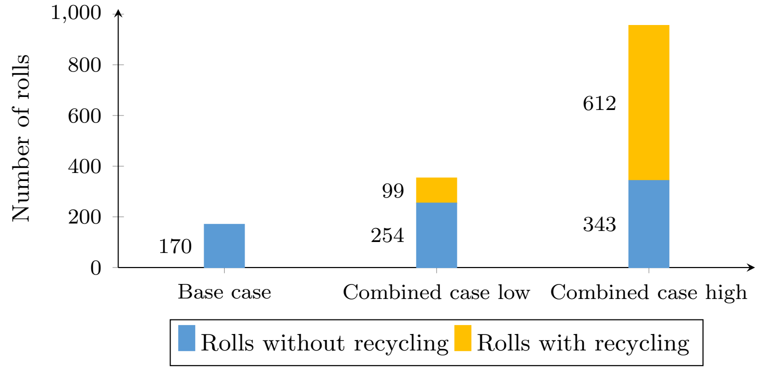


Figure 24: Number of rolls generated base case, combined case low and combined case high, averaged over both RT and $A_{T,i}$-interval

6.3.2 Managerial Decisions

In this section, the managerial decisions regarding RT and $A_{T,i}$-interval are discussed. Recollect that the canal operators are in control of these parameters. An $A_{T,i}$-interval of 4-6 hours is the current practice, but the operators are considering to extend the interval to 6-8 hours. It is unknown which re-planning threshold that is used per today.

Figure 25 shows average traversing time and average computational time for combined case low and high with all configurations of RT and $A_{T,i}$-interval. The average computational time is the average of all rolls in each scenario where a full regeneration of the schedule takes place. In other words, rolls where replanning happens with recycling are not considered. It is evident from the figure that computational time is highly sensitive to RTs and $A_{T,i}$-intervals, while the average traversing time is rather insensitive.

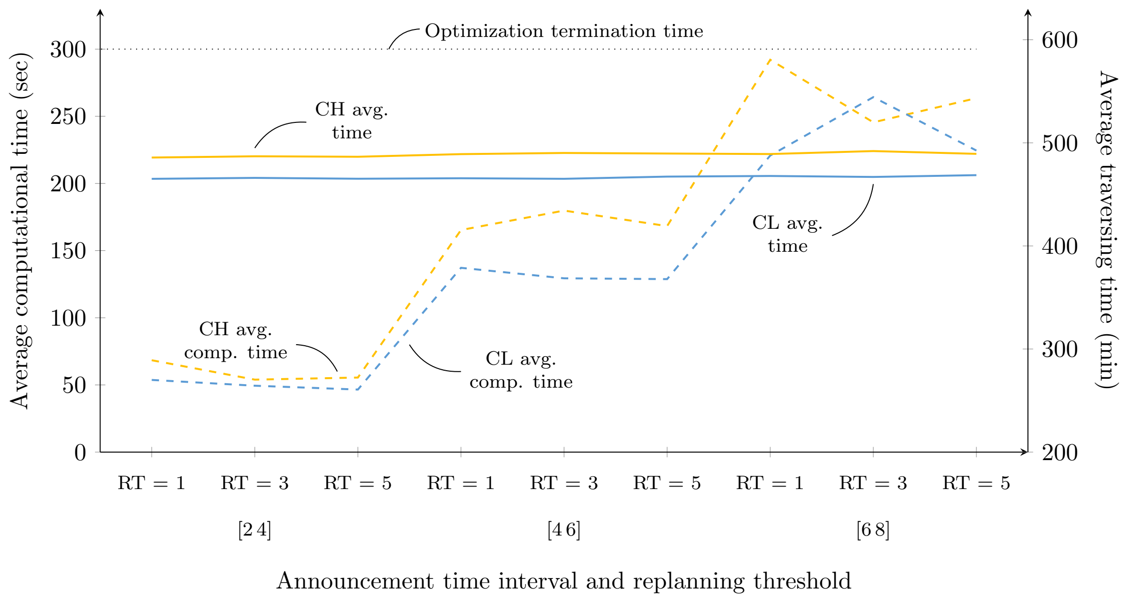


Figure 25: Average computational time and traversing time for combined case low and combined case high

For RT1_AT24_CH the average time spent on creating a schedule is 68 seconds. Compare this to the the scenario RT1_AT68_CH, where average computational time is 292 seconds, approaching the optimization termination time of 300 seconds. The reason for this increase is likely due to the larger look-ahead period implied by the increased $A_{T,i}$-interval. A larger look-ahead period means that more ships are included in each schedule, and thus, the amount of ship sequence variables increases.

In theory, extending the look-ahead period could be beneficial as more information regarding arrivals is available, and the optimizer may account for more ships. However, this is not the case for the scenarios studied here. Instead, average traversing time is marginally increased as the look-ahead period grows. There are two plausible explanation for this phenomena: Firstly, the severity of delay increases with increasing $A_{T,i}$-interval. Thus, the worsening in average traversing time might be due to the increased extent and frequency of delays. Secondly, a larger average computational time implies that more rolls reaches the optimization termination time before optimality is achieved. It has throughout this thesis been stressed that an optimal solution to the KCSTOP not necessarily performs the best in a composite solution. However, it is likely that the solution to the DKCSTOP is enhanced when it is made up of from near-optimal or optimal schedules. Therefore, stopping the optimizer before it finds sufficiently good schedules might reduce the solution quality of the composite. Figure 26 shows the amount of rolls solved to optimality for the scenarios discussed. Note how the percentages decrease when the $A_{T,i}$-interval increases, meaning that scenarios with a smaller $A_{T,i}$-interval have more rolls solved to optimality. Bearing in mind that the adjusted average traversing time is decreased for smaller $A_{T,i}$-intervals, the hypothesis that composite solutions are enhanced when using optimal solutions to the KCSTOP is supported. Also observe how higher RTs increase the number of rolls solved to optimality when keeping the $A_{T,i}$-interval constant.

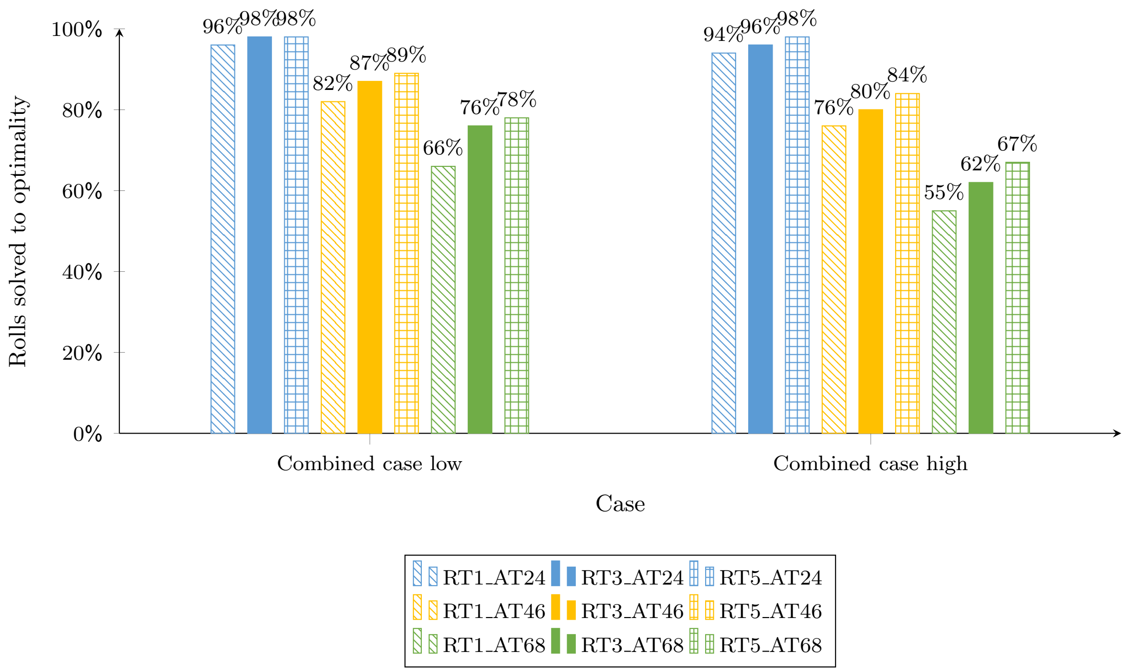


Figure 26: Rolls solved to optimality for combined case low and combined case high

In summary, the average traversing time is rather insensitive to changes in the RT and $A_{T,i}$-interval. Conversely, computational time is very sensitive to changes in these parameters. This indicates that the canal operators can to reduce the $A_{T,i}$-interval in order to generate schedules quicker, without compromising solution quality. However, as the operators intend to increase the $A_{T,i}$-interval, they should consider optimization termination times larger than five minutes in order to possibly obtain better solutions. The data shows that RT has marginal impact on average traversing time.

6.3.3 Desirable Features

According to Lübbecke et al. (2014), the operators desire an upper limit on the amount of time a ship must wait during its travel. This limit is set on both the total waiting time a ship experiences throughout the canal, and on the time it waits on a single siding. The total waiting time during a ship's travel throughout the canal should not exceed three hours (180 minutes) for ships of $TGN_i \leq 5$, and two hours (120 minutes) for ships of $TGN_i = 6$. Furthermore, ships of $TGN_i \leq 5$ should not wait more than one and a half hour (90 minutes) in each individual siding, while this is reduced to one hour (60 minutes) per siding for ships of $TGN_i = 6$.

Table 9 shows the maximum total waiting observed, average total waiting, and the fraction of ships that satisfy the maximum total waiting target for combined case high and low. Since the traversed ships assessed are not partitioned by TGN, the most restrictive target of 120 minutes is used. Thus, the fractions reported are somewhat pessimistic. It is evident that in all combined case high scenarios, and in all except one combined case low scenarios, there is at least one ship that has a maximum total waiting time far above the target. In fact, for RT3_AT68_CH, the maximum total waiting is 324, 170% above the target. However, for the same case, 90% of the ships have waiting times below the target of 120 minutes. On average, a satisfying 95% and 92% of the ships have a total waiting time less than the target in combined case low and high, respectively. Furthermore, the average waiting for each ship is 39 and 48 minutes for combined case low and high, respectively. Thus, the majority of ships traversing the canal experience a total waiting less than half of the most restrictive target.

		Combined case low			Combined case high		
RT	$A_{T,i}$ -interval	Max. tot. waiting (min)	Avg. waiting (min)	Ships meeting target	Max. tot. waiting (min)	Avg. waiting (min)	Ships meeting target
1	$[2,4]$	223	38	95%	217	45	93%
	$[4,6]$	218	38	95%	285	48	92%
	$[6,8]$	205	40	94%	256	49	90%
3	$[2,4]$	214	39	94%	287	47	92%
	$[4,6]$	182	38	94%	311	48	92%
	$[6,8]$	174	39	94%	324	52	90%
5	$[2,4]$	203	38	94%	229	46	93%
	$[4,6]$	203	40	96%	231	48	92%
	$[6,8]$	174	41	96%	244	50	92%
Case average		**200**	**39**	**95%**	**265**	**48**	**92%**

Table 9: Maximum total waiting, average waiting, and ships meeting the waiting target for combined case low and combined case high

Table 10 shows the maximum single waiting observed, and the fraction of ships that satisfy the maximum single waiting target for all scenarios in combined case high and low. Once again the strictest target is used, and hence the fractions represent the amount of ships that experience single waiting times of less than 60 minutes. Observe that in all scenarios there is at least one ship with a longer waiting time than the softest target. Nevertheless, the amount of ships that do meet the target is once again satisfying. 93% and 90% of all ships waited less than 60 minutes on each siding on average in the combined case low and high, respectively.

RT	$A_{T,i}$ -interval	Combined case low: Max. waiting (min)	Combined case low: Ships meeting target	Combined case high: Max. waiting (min)	Combined case high: Ships meeting target
1	[2, 4]	97	93%	125	92%
	[4, 6]	135	94%	135	89%
	[6, 8]	133	95%	133	89%
3	[2, 4]	109	94%	109	89%
	[4, 6]	112	91%	112	92%
	[6, 8]	133	94%	133	88%
5	[2, 4]	112	93%	112	91%
	[4, 6]	95	94%	95	90%
	[6, 8]	112	91%	112	89%
Case average		**116**	**93%**	**119**	**90%**

Table 10: Maximum single waiting and ships meeting the single waiting target for combined case low and combined case high

Although average waiting times are satisfying, and that a profound fraction of all ships do meet the targets, both the maximum total and single waiting times observed in all scenarios are severely high. Hilstad and Skjaeveland (2017) develop a Waiting Constrained Model as an alternative formulation of the KCSTOP (this can be found in Appendix E). The model contains constraints that account for the desirable features. An extension of this model can be implemented in the simulation-optimization framework to account for the waiting limits, and is discussed further in Section 8.1.3.

6.4 Effect of the Weighted Approach

The goal of this section is to measure and analyze the effect of the weighted approach introduced in Section 4.4. In Subsection 6.4.1, the performance of the weighted approach is studied, followed by Subsection 6.4.2, which contains a rough estimate on the improvements made with the weighted approach in monetary terms.

6.4.1 Performance of Weighted Approach

Figure 27 compares the average traversing time, averaged over both RT and $A_{T,i}$-interval, for both solution approaches. From the figure it can be observed that the weighted approach is consistently improving the average traversing times by some minutes for all cases.

In order to quantify how large the improvement really is, it is necessary to compare the reduction to the maximum reduction achievable. Remember that all ships must obey an upper speed limit depending on their TGN. Ships of $TGN_i \geq 5$ have a speed limit of 12 km/h and hence use minimum 6.5 hours to traverse the full length of the canal, while ships of $TGN_i < 5$ have a speed limit of 15 km/h and use at least 8 hours. In Table 11 this *conflict-free traversing time* is used as a lower

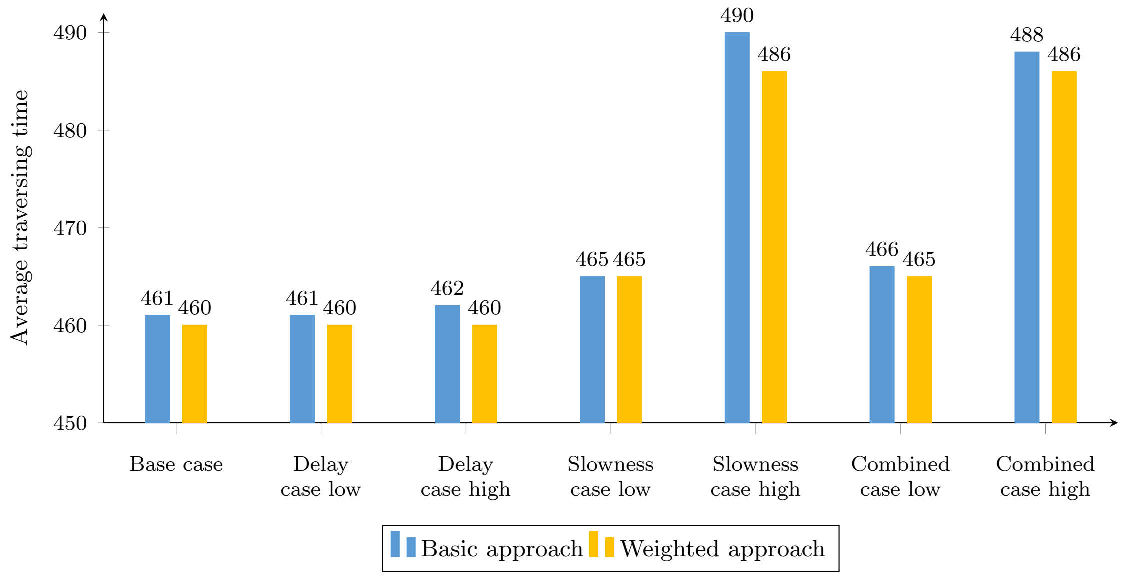


Figure 27: Average traversing time for all cases using basic approach and weighted approach

bound in order to measure the quality of the solutions. The table shows the improvements, or achieved reductions, in average traversing time when using the weighted approach as opposed to the basic approach for all cases. It also shows the maximum possible reduction calculated by taking the difference between the average traversing time using the basic approach and the average conflict-free traversing time. The achieved improvements are further shown as a percentage of the maximum possible reduction. All numbers are averaged over both RT and $A_{T,i}$-interval. Table 11 reveals that the improvements make up several percent of the maximum possible reduction. Averaging across all 63 scenarios, 3.2% of the maximum possible reduction is achieved when using the weighted approach.

Case	Achieved reduction (min)	Maximum possible reduction (min)	Achieved reduction as percentage of maximum possible reduction
BC	0.6	35.6	1.6%
DL	1.0	36.3	2.8%
DH	1.1	36.3	2.9%
SL	0.1	38.8	0.3%
SH	3.7	64.7	5.7%
CL	1.9	41.1	4.5%
CH	2.9	63.4	4.6%
Case average	**1.6**	**45.3**	**3.2%**

Table 11: Improvement in average traversing time for the weighted approach compared to the maximum possible reduction

As stated in Section 4.4, the purpose of introducing the weighted approach is two-fold. Firstly, it is meant to reduce the problem of degeneracy and hence decrease computational time. Secondly, the approach should benefit the solution of the DKCSTOP by lessening the impact of disruptive events.

To analyze whether the weighted approach is able to solve the degeneracy problem, it is helpful to look at the average optimality gap for the schedules created in each roll. Figure 28 shows the average optimality gap in every combined scenario. Observe that the combined case low has a higher average gap than the combined case high, despite having less disruptive events. This observation is explained by the fact that combined case high has disproportionately more slowness events than the combined case low. All rolls triggered by a slowness event are solved to optimality, and thus these rolls drastically lower the average gap for combined case high. It is evident from Figure 28 that the average gap gradually increases as the $A_{T,i}$-interval changes from 2-4 hours to 6-8 hours. For scenarios with the smallest $A_{T,i}$-intervals, both solution approaches are able to reach an average optimality gap of approximately 0. However, the weighted approach is able to reduce the gap by some additional percentages compared to the basic approach for the scenarios of larger intervals. This difference in optimality gap reveals that the weighted approach is able to search the branch-and-bound tree more efficiently than the basic approach, resulting in either better solutions being found or stronger LP bounds. As pointed out several times throughout this thesis, better solutions to the KCSTOP does not necessarily mean that the composite solution is improved, but it is assumed that good composites are in general made up of near-optimal solutions. Therefore, in the scenarios where the weighted approach is able to find better solutions in one or more rolls, it is plausible that the composite solution improves. Thus, it is likely that the improvements shown in Table 11 are not only due to the postponement of waiting, but also due to the reduction of degeneracy.

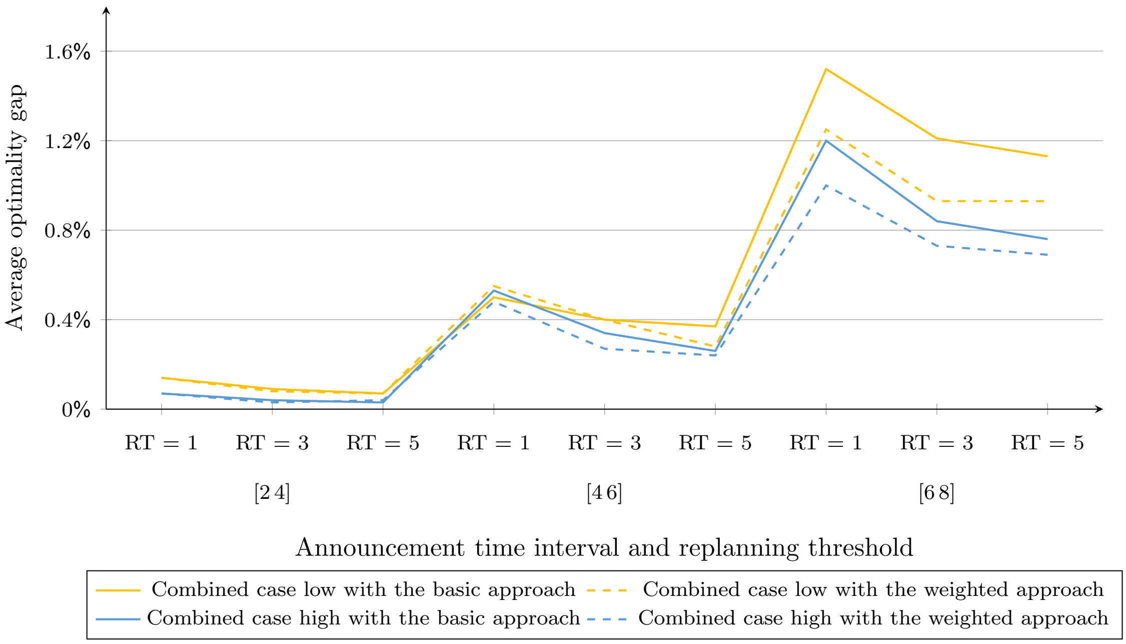


Figure 28: Average optimality gap for combined case high and combined case low using the basic approach and the weighted approach

6.4.2 Improvements in Monetary Terms

In order to put the improvements made by the weighted solution approach into perspective, the improvements are estimated in monetary terms. Note that all numbers presented in this paragraph are rough estimates. The combined case high is assumed the most realistic case, with both delays and slowness occurring frequently, and is therefore scrutinized. As of 1st of June 2018, the daily Container Shipping Rate for vessels with a capacity of 1 700 TEUs (Twenty-Foot-Equivalent Units) is 10 700 USD (Petersen (2018)). Ships with a capacity of 1 700 TEU are assigned a traffic group number of 5 in the Kiel Canal. Based on the one month historical data provided, every fourth ship has this TGN. Therefore, it is assumed that the average daily charter rate is substantially lower than 10 700 USD. For the sake of simplicity, a daily rate of 5 000 USD, or equivalently 3.5 USD per minute, is used. Thus, when considering that about 30 000 ships sail through the canal every year, and that each ship saves 2.9 minutes each (Table 11), possible yearly savings for the shipping companies using the Kiel Canal are around 300 000 dollars. Although this number is not particularly large in a worldwide shipping context, it does signify the value of decreasing each ship's traversing time with just a couple of minutes.

The improvement can also be measured in terms of additional income generated by the Kiel Canal operators. Again, using the numbers for the combined case high, the basic approach schedules on average 300 ships through the canal during one week of operations. The weighted approach manages to schedule 301 ships on average in the same time span, which is 0.5% more ships relative to the basic approach. Thus, assuming that the basic approach schedules 30 000 ships yearly, the weighted approach manages to schedule 30 150 ship yearly. For the ships in the data set, the weighted average transfer fee in the canal is 3 980 euros[2]. This corresponds to an additional, yearly income of around 600 000 euros if the weighted approach is used (assuming excess demand).

6.5 Changing Disruption Management Strategies

In Subsection 6.2.1, it is shown that replanning without recycling is able to more or less completely mitigate disruptions caused by delay. In Subsection 6.2.2, it is shown that replanning with recycling is not completely sufficient in handling disruptions caused by slowness as the solution quality is somewhat compromised, but in return the required computational time is decreased. However, it might be possible that replanning with recycling mitigates the disruptions caused by delay to a satisfying extent, and thereby saving computational time. Conversely, when slowness is the triggering event, it might be beneficial to spend more computational time and do a complete regeneration of the schedule in order to increase the solution quality. Finding the optimal way of handling disruptions is worth a study in its own right and is out of scope of this thesis. Nevertheless, this section aims at giving insight into the consequences of changing the disruption management decisions implemented in the simulation-optimization framework.

For the sake of simplicity, the different configurations of disruption management strategies used in this section are given names:

- *Original setting* is the original configuration in which a full regeneration of the schedule is done when delay is the replanning trigger, while replanning with all binary variables being re-used is done when slowness is the trigger
- *Altered setting* is the configuration where the framework calls the opposite procedure than in the original setting. I.e., replanning with recycling is done when delay is the trigger, while a full regeneration with no recycling is done when slowness is the trigger

[2] See Appendix F for the calculation of weighted transfer fee

Table 12 shows the average traversing time and average computational time when solving the DKCSTOP over four scenarios with both the original and the altered setting. The scenarios studied are delay case high and low and slowness case high and low with RT of 3 and $A_{T,i}$-interval of 4-6 hours. Note that the reported numbers for the slowness cases are not adjusted since the relative change is measured across the same case.

Case	Original setting		Altered setting		Relative	
	Avg. time (min)	Avg. comp time (min)	Avg. time (min)	Avg. comp time (min)	ΔAvg. time	ΔAvg. comp time
Delay case low	461	2.1	462	1.1	0.3%	-47.9%
Delay case high	460	2.4	465	0.7	1.0%	-72.3%
Slowness case low	466	1.3	459	1.8	-1.3%	45.4%
Slowness case high	489	0.4	459	1.8	-6.2%	325.5%

Table 12: Average traversing time for delay and slowness cases with original and altered configuration

The results from the delay cases are first examined. The increase in average traversing time is small, but noticeable. For delay case low, the solution is one minute worse when using the altered setting, while for delay case high the worsening is five minutes. The slight reduction in solution quality has to be measured against the reduction in computational time. Figure 29 shows the number of schedules generated for the two configurations.

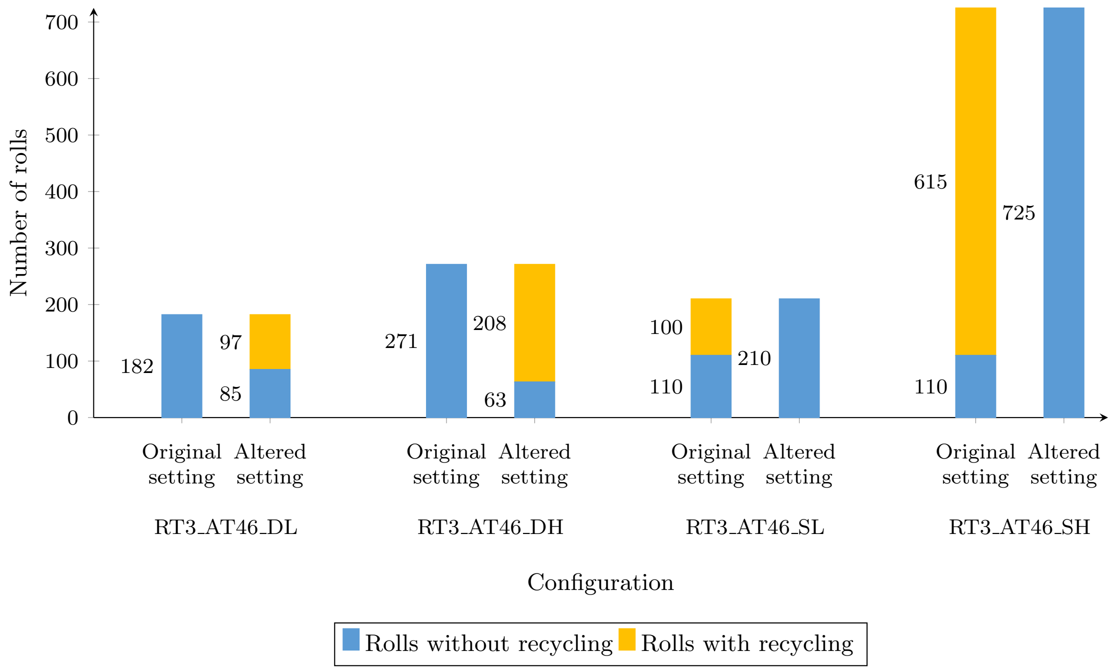


Figure 29: Number of schedules generated when solving the DKCSTOP with original and altered setting

In RT3_AT46_DL more than 50% of the rolls are with recycling, while in RT3_AT46_DH this increases to about 75%. Thus, a major decrease in computational time is achieved by changing the strategy that handles delay. An average of 7.8 hours is saved weekly, or 1.1 hours daily. This reduction in computational time indicates the benefit of recycling binary variables when rescheduling is due to delay. It is argued in Section 5.4 that replanning without recycling is reasonable when handling delay just because time is assumed to not be a scarce resource. This assumption could be revisited. Recall that when delay is the triggering event, a new schedule is required both at the estimated time of arrival and when the ship eventually arrives. Therefore, an idea could to use the original setting for one of the two replanning situations, and the altered setting for the other. Computational time is then saved, and solution quality is probably preserved.

For slowness case high, the reduction in average traversing time is 6.2%, which is equivalent to 30 minutes saved per ship on average. This is a major increase in solution quality. The price that must be payed is having 85% more rolls where a full regeneration takes place. This is equivalent to an increase of 16.8 hours in computational time weekly, or, 2.4 hours daily. For slowness case low, the trade-off for decreasing the average traversing time by 1.3% is an increase in computational time of mere 0.3 hours daily. Thus, the altered setting is beneficial if slowness events occur at a low frequency.

As seen from these tests, the changes when switching from one setting to another is quite extreme. It is probable that the best trade-off between solution quality and computational time lies somewhere in between the values observed here. In order to achieve this middle-ground, it might be worth testing *replanning with partial recycling*, i.e, replanning where only a specific number of ship sequence variables are re-used. This is discussed further in Section 8.3. Furthermore, the results are highly dependent on the frequency of events. The distributions of events used in the high and low frequency cases are not anchored in historic data. This is discussed further in Subsection 8.2.2.

7 Concluding Remarks

In this thesis a mathematical model of the Kiel Canal Ship Traffic Optimization Problem (KCSTOP) is presented. The problem is concerned with the scheduling of maritime vessels through the Kiel Canal. A solution to the problem is a conflict-free schedule that minimizes the total travelling time for a set of ships. The KCSTOP does not account for dynamism and stochasticity which the daily scheduling task in reality is exposed to. Thus, the Dynamic Kiel Canal Ship Traffic Optimization Problem under Uncertainty (DKCSTOP) is established. The problem is complex to model mathematically, and therefore a simulation-optimization framework is developed. The framework consists of a simulator and an optimizer, where the former simulates the traffic flow and disruptive events in the canal, and the latter replans whenever needed. The DKCSTOP is implemented in the framework and a solution is found using a rolling horizon approach. The solution is a composite schedule made of consecutive solutions to constrained versions of the KCSTOP.

Ships arriving later than planned (delay), and ships traversing segments slower than intended (slowness) are the two disruptive events accounted for. In order to mitigate the impact of disruptions, appropriate disruptions management strategies are implemented. Replanning either with or without recycling of binary ship sequence variables and the proposed weighted approach are the three strategies developed.

In order to handle late ship arrivals, replanning without recycling is the chosen strategy. Results show that no significant change in average traversing time is observed when comparing scenarios with delay to scenarios without. Thus, the strategy manages to nullify the disruptions caused by delays, but this comes at a cost in terms of computational time. In the event that a ship fails to maintain its scheduled velocity, replanning with recycling is conducted. This procedure is costless in terms of computational effort. For the scenarios where the slowness frequency is low, the relative changes compared to base case are insignificant. Thus, disruptions are seemingly well managed. On the other hand, for the scenarios where the slowness frequency is high, solution quality is compromised. Nevertheless, the solution is on average only 3% worse than disruption-free scenarios.

The weighted approach manages to improve solution quality for all seven cases, averaged over all their belonging scenarios, compared to the basic approach. The reason why is two-fold: The enhancement strategy reduces the number of degenerate solutions that the optimizer searches among. This enables a quicker search through the branch-and-bound tree, and thus, with the same amount of computational time a better solution or stronger LP bound can be found. The other reason stems from the fact that postponing waiting makes it possible to avoid unnecessary waiting. For scenarios where both disruptive events occur frequently, the approach reduces average traversing time with nearly three minutes on average. In monetary terms, a three minute improvement roughly corresponds to 300 000 dollars saved for shipping companies. Moreover, an estimated 600 000 euros in additional revenue is generated by the canal operators.

Besides minimizing traversing time, no ship should have to remain at rest for more than a certain limit, and no ship should experience an unfair amount of total waiting. The mathematical formulation of the KCSTOP does not account for these features. Nonetheless, it is observed that more than 90% of the ships indeed satisfy the strictest of these limits.

It is assessed how the trade-off between solution quality and time is affected when changing the disruption management strategies implemented in the framework. When using the altered setting in the delay scenarios, the average traversing time is not profoundly decreased. However, nearly eight hours of computational time is on average saved weekly when delays occur frequently. Similarly, when using the altered setting in the slowness scenarios, results reveal that these scenarios are sensitive to the strategy used. On average, with high slowness frequency, the average traversing time is decreased by 6.2% compared to the original setting. This major decrease comes at the cost of nearly 17 hours more in computational time per week.

All scenarios constitute of a re-planning threshold (RT) and an announcement time interval ($A_{T,i}$-interval), two decisions the canal operators are in charge of. Experiments reveal that the average traversing time is rather insensitive to changes in both the RT and the $A_{T,i}$-interval. Computational time is on the other hand very sensitive to change in the $A_{T,i}$-interval, and is reduced when the $A_{T,i}$-interval is reduced. The operators are currently considering to increase the $A_{T,i}$-interval. However, our study shows that this increases computational time needed without improving solution quality. The operators can instead choose to reduce the $A_{T,i}$-interval in order for faster computations, without compromising solution quality. Having that said, if time is not scarce and the operators prefer to accumulate more information regarding arriving ships, these preferences should be prioritized as solution quality is insensitive.

In conclusion, the simulation-optimization framework developed is a flexible decision support tool for the scheduling of ships through the Kiel Canal. The framework is able to simulate the real-life planning situation in the Kiel Canal; measure the performance of different disruption management strategies; test different solution approaches and aid in the scheduling of ships. The strategies tested in this thesis manages to mitigate disruptions to a great extent, and at the same time keep computational requirements reasonably low. However, the human aspect of decision making is not entirely replaceable by an optimization tool. The best solutions are created when the experience and knowledge of the canal operators are exploited and used together with the valuable insight the simulation-optimization framework provides.

8 Further Research

The simulation-optimization framework developed in this thesis can be improved in multiple ways. This chapter outlines some directions for further work in order to develop a better decision support tool for the canal operators. Section 8.1 contains a discussion on methods that might improve the mathematical formulation and hence the optimization routine, while a discussion on advancing the simulator is found in Section 8.2. Finally, Section 8.3 presents how the framework as a whole can be improved.

8.1 Improving the Mathematical Formulation

This section outlines possible improvements to the mathematical formulation of the KCSTOP presented in Chapter 4. Robustness is reflected on in Subsection 8.1.1, ideas for solving the degeneracy issue in the KCSTOP is presented in Subsection 8.1.2, an alternative solution approach is discussed in Subsection 8.1.3, and lastly, Subsection 8.1.4 suggests how the limiting siding capacity can be handled.

8.1.1 Adding Robustness

In the developed framework, replanning is conducted every time a disruptive event occurs. However, this might not be the best solution method for the DKCSTOP. Robustness can be added to the KCSTOP, making each schedule more robust with respect to disruptive events. The purpose is to keep the temporary schedules feasible under as many circumstances as possible. This implies that when a disruptive event occurs, replanning is not necessarily required. By reducing the amount of replanning procedures needed, the operators can carry on with the current schedule. E.g., this can reduce the computational time of approximately four hours a day in the scenarios of high delay frequency.

There exists several studies on robust planning within ship routing and scheduling. The research usually regards liner shipping, where ships are assigned to fixed routes in an optimal manner, and industrial and tramp shipping, where optimal routes for the fleet are found. In these problems the Kiel Canal would be a sailing stretch in a bigger scheduling scheme, and the modelling and solving of the KCSTOP is hence of a different nature than the aforementioned problems. Another major difference is that the main purpose of the KCSTOP is to provide the canal operators with a decision support tool, and the problem is considered from their perspective. In contrary, the planning problem in liner and tramp shipping is considered from the shipping companies perspective. However, robustness strategies found in these studies may be of inspiration if robustness is added to the KCSTOP. Halvorsen-Weare and Fagerholt (2011) study a supply vessel planning problem. Robustness is incorporated in the initial schedule by adding slack to the voyages and schedules, and by combining optimization and simulation to provide robust and resilient schedules. Halvorsen-Weare et al. (2013) study vessel routing and scheduling in the LNG business. One of several robustness strategies implemented is to add slack to the schedule in terms of extra sailing time. Fischer et al. (2016) incorporate robustness to the fleet deployment in roll-on-roll-of liner shipping by using four strategies. One of them is adding extra sailing time on each sailing leg. It is apparent that adding slack in sailing times could be a strategy to start out with when adding robustness to the KCSTOP.

Visentini et al. (2014) state that building robustness into an initial schedule, i.e. in the in-advance planning stage, can handle disruptions. However, this is not always enough when disruptions are severe. Besides, the schedule can become very costly if robustness should account for such severe disruptions. The authors mention that robust scheduling and real-time replanning, which earlier have been treated isolated and as independent study fields, should be combined in decision support tools. Thus, a combination of adding robustness to the KCSTOP and resolving a constrained version of it when robustness does not suffice, appears as an excellent method for managing disruptions in the DKCSTOP. However, the benefit of added robustness must be closely evaluated when bearing in mind that the replanning procedure in the developed framework produces satisfying schedules within a short time span. I.e., it might be more beneficial to reschedule when a disruption occurs instead of having costly robustness dealing with it.

8.1.2 Solve the Problem of Degeneracy

A substantial issue with the formulation of the KCSTOP is the problem of degeneracy. Hence, extensions of the mathematical model should include elements reducing the degeneracy problem. The weighted approach developed in this thesis reduce the amount of degenerate solutions to some extent. Another possible strategy to the degeneracy issue is to have several objective functions, similar to what is done by Castillo et al. (2009). The procedure could be to first solve a mathematical formulation of the KCSTOP with a first objective, for instance the one proposed in Equation (9). Then, a modified formulation with a secondary objective is solved, only searching among the equally good solutions with respect to the previous objective. A tertiary objective and so one are considered in an iteratively manner. Each iteration reduces the number of optimal solutions, until the best solution is attained.

In the mathematical formulation of the KCSTOP presented in this thesis, the time variables are continuous. By making all time variables time-discrete, the problem of degeneracy can be mitigated to a great extent.

8.1.3 Other Solution Approaches

In this thesis, two solution approaches are tested in the simulation-optimization framework when solving the DKCSTOP, namely the basic and the weighted approach where the latter includes an enhancement strategy of postponing waiting. Several other solution approaches can be developed, and by implementing them in the framework it can be investigated if these yield solutions quicker or solutions of higher quality. For instance, the Waiting Constrained Model (WCM) developed by Hilstad and Skjaeveland (2017) can be considered. The model targets waiting to be below some maximum limits. The rationale behind the waiting targets is to ensure a fair distribution of waiting time among ships. For instance, if one ship must wait for a very long time while others pass through with much less waiting, this is considered unfair. A fair distribution of waiting time might stimulate to make the Kiel Canal the preferred choice for shipping companies rather than the detour around Jutland, and thus, help the canal operators in sustaining their market shares.

Furthermore, an advantage of adding waiting time constraints is the possibility of reducing the amount of binary variables and the size of big M. This enables computational requirements to be lessened. The WCM can therefore be tested for faster problem solving of the DKCSTOP in addition to reducing maximum waiting times. However, as explained in Appendix E, the use of the WCM may yield an infeasible problem. A way to overcome this hinder is to allow for the waiting time constraints to be defied, but penalize such a failure to comply in the objective function. See Appendix E for the mathematical model of the WCM.

8.1.4 Siding Capacity

The mathematical formulation of the problem can be extended in order to model the traffic management in the Kiel Canal more realistically. The sidings of the canal are of finite size, meaning that each siding can only hold a certain number of ships. The number depends on the length of the siding and the length of the ships located on it, in addition to some safety parking distance. The formulation of the KCSTOP in this thesis have relaxed these constraints. However, such siding capacities may have a significant impact on the solution and should be considered. One way to account for siding capacity could be to develop an iterative procedure that checks whether the siding capacities are satisfied after solving the KCSTOP. If not, capacity constraints can be added for the sidings where the capacity is exceeded. By doing so, a solution where all siding capacities are satisfied is eventually found, given that a feasible solution exist.

8.2 Improving the Simulator

One way of improving the developed simulator is to make it imitate the traffic flow in the Kiel Canal more realistically. Several disruptive events discussed in Subsection 8.2.1 can be included. Furthermore, the distributions used when generating events should be assessed, and is discussed in 8.2.2.

8.2.1 Adding Several Disruptive Events

According to Prof. Dr. Frank Meisel there are several more disruptive events that are known to happen besides delay and slowness. Some examples of such disruptive events are described in the following.

Extended Waiting
When ships are waiting for prolonged periods, it might take some time for them to get going again. This can happen if the ship crew is inattentive to the traffic lights, or if the ship does not manage to accelerate as planned. Whatever reason, the result is that the ship rests on a siding longer than scheduled, which may lead to infeasibilities. It is difficult to generate this event prior to simulation as it is impossible to predict on which sidings a certain ship will wait while traversing the canal. The existing framework requires all events to be (partially) pre-generated, and thus, it might require another simulation framework in order to implement this event properly.

Internal Entering
As mentioned in Chapter 2, there are several harbours along the canal where ships may want

to dock (Figure 1 shows the most important ones). Some ships may opt to stay in one of these harbours for several days before continuing their voyage. According to Prof. Dr. Frank Meisel, it is unusual that these ships announce their arrival. Therefore, the event of a ship entering internally could be regarded as a disruption.

ETA swap
When several ships arrive at the canal nearly simultaneously, it is common to let them enter through the locks in batches. Due to this, the order in which ships arrive at the canal, and the order they start to traverse in, may differ. In an event like this, the $T_{ETA,i}$ are switched around among the ships entering in the batch, making the traversing order of the first segment different.

TGN swap
When a ship announces its arrival at the canal, the dimensions and the draft of the ship are reported to the canal operators. This information is used to assign a traffic group number (TGN) to the ship. However, the draft can be difficult to estimate accurately, as both water density and ship velocity affects it. Thus, it is sometimes necessary for the canal operators to change the TGN of a ship at its arrival.

8.2.2 Distribution of Events

In this thesis delays are assumed to be exponentially distributed, a choice based on several papers regarding train delays. The generation of slowness is based on ships' TGNs with the assumption that larger ships sail slow more frequently than smaller ships. Thus, neither delay nor slowness distributions are anchored in historic data. However, it could be possible to gather historic data regarding events and their occurrences and thereby make distributions more realistic. During our thesis period we did not have direct contact with the canal operators. Thus, we are unaware of what historic data that exist. If such data exists this could be acquired and utilized with the purpose of making empirical event distributions. If not, it could be an idea to initiate data collection. By using more realistic distributions of events, the simulator can become even more realistic.

8.3 Improving the Simulation-Optimization Framework

In order to implement the DKCSTOP in the simulation-optimization framework, several important decisions related to disruption management must be made. If and when to create a new schedule, and when and what to re-use from the current one are examples of such decisions. All of these decisions are known to have an impact on computational time, but will likely affect solution quality as well. Hence, it is of interest to analyze how the performance indicators change with these replanning decisions. As shown in Section 6.5, it can be beneficial to replan with fixed binary variables when delay is the trigger, or to do a full regeneration when slowness is the trigger. However, there is a major trade-off between computational time and solution quality when restricting the optimizer to either use all or none of the ship sequence variables. In this thesis, replanning with recycling re-uses *all* of the ship sequence variables from the current schedule. This choice is not necessarily optimal. Thus, in further development of the framework it should be investigated whether partial recycling could be effective.

In the developed simulator it is assumed that each replanning procedure is done instantaneously, while in fact, the optimizer has a termination time of five minutes. This entails a logical shortcoming as the replanning frequency in simulation minutes is sometimes less than the time it takes to replan. E.g., a replanning procedure may take place at time $\tau = 320$, and the procedure uses a real time of five minutes. The developed simulator allows for a next replanning to take place before $\tau = 325$. This hitch should be removed in an improved framework.

References

Acuna-Agost, R., Michelon, P., Feillet, D., and Gueye, S. (2011). A mip-based local search method for the railway rescheduling problem. *Networks*, 57(1):69–86.

Almaz, A., Or, I., and Ozbas, B. (2006). Investigation of the transit maritime traffic in the strait of istanbul through simulation modeling and scenario analysis. *International Journal of Simulation*, 7(7):1–9.

Almaz, O. A. and Altiok, T. (2012). Simulation modeling of the vessel traffic in delaware river: Impact of deepening on port performance. *Simulation Modelling Practice and Theory*, 22:146–165.

Amaran, S., Sahinidis, N. V., Sharda, B., and Bury, S. J. (2014). Simulation optimization: a review of algorithms and applications. *4OR*, 12(4):301–333.

Castillo, E., Gallego, I., Ureña, J. M., and Coronado, J. M. (2009). Timetabling optimization of a single railway track line with sensitivity analysis. *Top*, 17(2):256–287.

Cordeau, J.-F., Toth, P., and Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation science*, 32(4):380–404.

Cortés, P., Muñuzuri, J., Ibáñez, J. N., and Guadix, J. (2007). Simulation of freight traffic in the seville inland port. *Simulation Modelling Practice and Theory*, 15(3):256–271.

Fagerholt, K., Christiansen, M., Hvattum, L. M., Johnsen, T. A., and Vabø, T. J. (2010). A decision support methodology for strategic planning in maritime transportation. *Omega*, 38(6):465–474.

Fischer, A., Nokhart, H., Olsen, H., Fagerholt, K., Rakke, J. G., and Stålhane, M. (2016). Robust planning and disruption management in roll-on roll-off liner shipping. *Transportation Research Part E: Logistics and Transportation Review*, 91:51–67.

Fu, M. C. (2002). Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215.

Golkar, J., Shekhar, A., and Buddhavarapu, S. (1998). Panama canal simulation model. In *Simulation Conference Proceedings, 1998. Winter*, volume 2, pages 1229–1237. IEEE.

Halvorsen-Weare, E. E. and Fagerholt, K. (2011). Robust supply vessel planning. In *Network optimization*, pages 559–573. Springer.

Halvorsen-Weare, E. E., Fagerholt, K., and Rönnqvist, M. (2013). Vessel routing and scheduling under uncertainty in the liquefied natural gas business. *Computers & Industrial Engineering*, 64(1):290–301.

Harris, M. (2006). *Analysis and modelling of train delay data*. PhD thesis, MSc Thesis, University of York, UK.

Higgins, A. and Kozan, E. (1998). Modeling train delays in urban networks. *Transportation Science*, 32(4):346–357.

Higgins, A., Kozan, E., and Ferreira, L. (1996). Optimal scheduling of trains on a single line track. *Transportation Research Part B: Methodological*, 30(2):147–161.

Hilstad, K. and Skjaeveland, G. (2017). Scheduling of bidirectional ship traffic in the kiel canal. Project thesis at NTNU.

Kiel Canal Official Website (2017). Kiel canal official website. [https://www.kiel-canal.de/index.htm. Last accessed June 06, 2018].

Köse, E., Başar, E., Demirci, E., Güneroğlu, A., and Erkebay, Ş. (2003). Simulation of marine traffic in istanbul strait. *Simulation Modelling Practice and Theory*, 11(7-8):597–608.

Krüger, N. A., Vierth, I., and Fakhraei Roudsari, F. (2013). Spatial, temporal and size distribution of freight train delays: evidence from sweden.

Lalla-Ruiz, E., Shi, X., and Voß, S. (2016). The waterway ship scheduling problem. *Transportation Research Part D: Transport and Environment*.

Li, Q. and Lam, J. S. L. (2017). Conflict resolution for enhancing shipping safety and improving navigational traffic within a seaport: vessel arrival scheduling. *Transportmetrica A: Transport Science*, pages 1–15.

Lübbecke, E., Lübbecke, M. E., and Möhring, R. H. (2014). Ship traffic optimization for the kiel canal. In *Technical Report 4681*. (Optimization Online).

Özbaş, B. and Or, I. (2007). Analysis and control of maritime transit traffic through the istanbul channel: a simulation approach. *Central European Journal of Operations Research*, 15(3):235–252.

Petersen, H. (2018). https://ycharts.com/indicators/container_shipping_rate_for_1700_teu_vessels. Online; Last accessed June 09, 2018.

Smith, L. D., Sweeney, D., and Campbell, J. F. (2009). Simulation of alternative approaches to relieving congestion at locks in a river transportion system. *Journal of the Operational Research Society*, 60(4):519–533.

Thiers, G. F. and Janssens, G. K. (1998). A port simulation model as a permanent decision instrument. *Simulation*, 71(2):117–125.

Törnquist, J. and Persson, J. A. (2007). N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B: Methodological*, 41(3):342–362.

Ulusçu, Ö. S., Özbaş, B., Altıok, T., Or, İ., and Yılmaz, T. (2009). Transit vessel scheduling in the strait of istanbul. *The Journal of Navigation*, 62(1):59–77.

UNCTAD (2017). Review of maritime transport 2016. http://unctad.org/en/PublicationsLibrary/rmt2017_en.pdf. Online; Last accessed June 06, 2018.

und Schifffahrtsamt Kiel-Holtenau, W. (2018). Die verkehrsgruppen auf dem nord-ostsee-kanal. http://www.wsa-kiel.wsv.de/Service/Publikationen/Anlagen/NOK-Verkehrsgruppen.pdf. Online; Last accessed June 10, 2018.

Visentini, M. S., Borenstein, D., Li, J.-Q., and Mirchandani, P. B. (2014). Review of real-time vehicle schedule recovery methods in transportation services. *Journal of Scheduling*, 17(6):541–567.

Wikipedia (2017). Kiel canal map. [https://upload.wikimedia.org/wikipedia/commons/7/77/Map_of_the_Kiel_Canal.png. Last accessed October 10, 2017].

Yu, G. and Qi, X. (2004). *Disruption management: framework, models and applications*. World Scientific.

Yuan, J., Goverde, R., and Hansen, I. (2002). Propagation of train delays in stations. *WIT Transactions on The Built Environment*, 61.

Zhou, X. and Zhong, M. (2007). Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B: Methodological*, 41(3):320–341.

Appendix A Mathematical Formulation of the KCSTOP

A.1 Notation for the KCSTOP

All notation used in the mathematical model of the KCSTOP is given beneath:

<u>Sets:</u>

$\mathcal{V}$	Set of all ships.
$\mathcal{V}^E$	Set of eastbound ships, i.e. sailing from West to East
$\mathcal{V}^W$	Set of westbound ships, i.e. sailing from East to West
$\mathcal{T}_i$	Set of transit segments ship i will traverse through its journey
$\mathcal{S}_i$	Set of siding segments ship i will traverse through its journey
$\mathcal{P}_i$	Set of all segments ship i will traverse through its journey, $\mathcal{P}_i = \mathcal{T}_i \cup \mathcal{S}_i$
$\mathcal{C}_p^A$	Set of all conflicting pairs of aligned ships on segment p
$\mathcal{C}_p^O$	Set of all conflicting pairs of opposed ships on segment p, i.e. opposed ships were $TGN_i + TGN_j > P_p$
$\mathcal{C}_p$	Set of all conflicting pairs of ships on segment p, $\mathcal{C}_p = \mathcal{C}_p^A \cup \mathcal{C}_p^O$

<u>Indices:</u>

i, j	Indexing the set of all ships, $\mathcal{V}$
p	Indexing the set of segments, $\mathcal{P}$

<u>Parameters :</u>

$\underline{S}_{ip}$	Shortest time ship i uses to traverse segment p, based on the ship's speed limit
$\underline{T}_{ip}$	Earliest arrival time of ship i at segment p, based on the cumulative length of the segments up until p and the ship's speed
H_{ij}	Smallest safety time between ship i and j
M	$= \lvert\mathcal{V}\rvert \cdot \max\limits_{i \in \mathcal{V}} \left\{ \sum_{p \in \mathcal{P}} \underline{S}_{ip} \right\}$
$T_{ETA,i}$	Estimated time of arrival of ship i to the canal
TGN_i	Traffic group number of ship i, defining the ship's width and speed
P_p	Passage number of segment p, defining the segment's width

<u>Variables :</u>

w_{ip}	Waiting time for ship i in segment p
t_{ip}	Time at which ship i enters segment p
s_{ip}	Time ship i uses to traverse a segment p

z_{ijp}	set to 1 if ship j is scheduled after ship i in segment p, and 0 if ship i is scheduled after ship j

A.2 Mathematical Model of the KCSTOP

The compact mathematical model of the KCSTOP is given beneath:

$$\min \quad \sum_{i\in\mathcal{V}}\sum_{p\in\mathcal{S}_i} w_{ip} + \sum_{i\in\mathcal{V}}\sum_{p\in\mathcal{P}_i} s_{ip} + \sum_{i\in\mathcal{V}} \left(t_{i\underline{P}_i} - T_{ETA,i}\right) \tag{47a}$$

$$\text{s.t.} \quad t_{ip} + s_{ip} - t_{ip+1} = 0 \qquad i \in \mathcal{V}^E, \quad p \in \mathcal{T}_i \tag{47b}$$

$$t_{ip} + s_{ip} + w_{ip} - t_{ip+1} = 0 \qquad i \in \mathcal{V}^E, \quad p \in \mathcal{S}_i \backslash \{|S|\} \tag{47c}$$

$$t_{ip} + s_{ip} - t_{ip-1} = 0 \qquad i \in \mathcal{V}^W, \quad p \in \mathcal{T}_i \tag{47d}$$

$$t_{ip} + s_{ip} + w_{ip} - t_{ip-1} = 0 \qquad i \in \mathcal{V}^W, \quad p \in \mathcal{S}_i \backslash \{1\} \tag{47e}$$

$$t_{ip} + H_{ij} - t_{jp} \leq M\left(1 - z_{ijp}\right) \qquad \{i,j\} \in \mathcal{C}_p^A, \quad p \in \mathcal{P}_i \tag{47f}$$

$$t_{jp} + H_{ji} - t_{ip} \leq M z_{ijp} \qquad \{i,j\} \in \mathcal{C}_p^A, \quad p \in \mathcal{P}_i \tag{47g}$$

$$t_{ip} + s_{ip} - t_{jp} \leq M\left(1 - z_{ijp}\right) \qquad \{i,j\} \in \mathcal{C}_p^O, \quad p \in \mathcal{P}_i \tag{47h}$$

$$t_{jp} + s_{jp} - t_{ip} \leq M z_{ijp} \qquad \{i,j\} \in \mathcal{C}_p^O, \quad p \in \mathcal{P}_i \tag{47i}$$

$$z_{ijp+1} - z_{ijp} \leq 0 \qquad \{i,j\} \in \mathcal{C}_p^A \cap \mathcal{V}^E, \quad p \in \mathcal{T}_i \tag{47j}$$

$$z_{ijp-1} - z_{ijp} \leq 0 \qquad \{i,j\} \in \mathcal{C}_p^A \cap \mathcal{V}^W, \quad p \in \mathcal{T}_i \tag{47k}$$

$$s_{ip} \geq \underline{S}_{ip} \qquad i \in \mathcal{V}, \quad p \in \mathcal{P}_i \tag{47l}$$

$$t_{ip} \geq \underline{T}_{ip} \qquad i \in \mathcal{V}, \quad p \in \mathcal{P}_i \tag{47m}$$

$$w_{ip} \geq 0 \qquad i \in \mathcal{V}, \quad p \in \mathcal{S}_i \tag{47n}$$

$$z_{ijp} \in \{0,1\} \qquad \{i,j\} \in \mathcal{C}_p, \quad p \in \mathcal{P}_i \tag{47o}$$

Appendix B Notation for the DKCSTOP

All new notation used in the Simulation-optimization framework in addition to the notation from the mathematical formulation found in A.1 given beneath:

Sets:

$\mathcal{V}^C$	Set of all ships in the current schedule
$\mathcal{V}^N$	Set of all ships to be planned for in the new schedule
$\mathcal{V}^A$	Set of all announced ships, but not yet included in a plan
$\mathcal{V}^T$	Set of all fully traversed ships

Superscripts:

C	Indicating the current schedule
N	Indicating the new schedule

Simulation parameters:

$\Delta \underline{T}_A$	Decides upper endpoint of the announcement time interval
$\Delta \overline{T}_A$	Decides lower endpoint of the announcement time interval
f_d	Fraction of the ships in the problem instance that are delayed
β	Mean delay
f_s	Fraction of the ships in the problem instance that are slow
$\underline{SS}$	Minimum number of slow segments
$\underline{SS}$	Maximum number of slow segments
T_{end}	Simulation horizon

Parameters:

$\hat{w}_{ip}^C$	Value assigned to the waiting time variable in the current schedule
$\hat{t}_{ip}^C$	Value assigned to the entering segment time variable in the current schedule
$\hat{s}_{ip}^C$	Value assigned to the traversing time variable in the current schedule
$\hat{z}_{ijp}^C$	Value assigned to the ship sequence variable in the current schedule
τ	Simulation time
τ_0	Simulation start time
CS_i	Current segment of ship i, the segment p it is position on at time τ
$T_{A,i}$	Announcement time of ship i
$\underline{T}_{A,i}$	Lower bound of announcement time interval
$\overline{T}_{A,i}$	Upper bound of announcement time interval

$T_{d,i}$	Delay of a delayed ship i
$T_{L,i}$	The time at which a delayed ship i arrives at the canal
$T_{S,ip}$	The time at which a slow ship i on segment p should have arrived at the border of the following segment
T_S	$\min_{\substack{i \in \mathcal{V}^C \\ p \in \mathcal{P}_i}} \{T_{S,ip}\}$
$\overline{V}_{\max}$	Maximum velocity reduction of slow ships
R^N_{ip}	Reduced velocity of a slow ship i on segment p

Variables :

w^N_{ip}	Waiting time for ship i in segment p in the new schedule
t^N_{ip}	Time at which ship i enters segment p in the new schedule
s^N_{ip}	Time ship i uses to traverse a segment p in the new schedule
z^N_{ijp}	set to 1 if ship j is scheduled after ship i in segment p, and 0 if ship i is scheduled after ship j in the new schedule

Appendix C Performance Indicators

C.1 Performance Indicators from the Basic Approach

Scenario	Performance Indicators											
	Tot. time	$\lvert\mathcal{V}^T\rvert$	Avg. time	Max. time	Avg. waiting	Max. waiting	Max. tot. waiting	Number of rolls	Rolls with reused schedule (%)	Rolls with new schedule (%)	Rolls due to RT (%)	Rolls to optimality (%)
RT5_AT24_BC	139416	302	462	707	37	96	210	69	0	100	84	100
RT5_AT46_BC	138825	303	458	730	33	212	236	66	0	100	100	100
RT5_AT68_BC	140878	305	462	725	37	193	231	66	0	100	100	97
RT5_AT24_DL	140099	304	461	708	36	117	215	123	0	100	37	100
RT5_AT46_DL	137808	299	461	668	36	103	174	140	0	100	30	99
RT5_AT68_DL	140750	303	465	702	40	140	209	173	0	100	17	92
RT5_AT24_DH	137932	300	460	633	35	104	139	174	0	100	18	100
RT5_AT46_DH	139392	302	462	668	36	93	174	233	0	100	11	96
RT5_AT68_DH	138960	300	463	685	39	122	191	281	0	100	7	91
RT5_AT24_SL	141563	304	466	708	38	110	215	169	59	41	34	100
RT5_AT46_SL	141373	303	467	768	39	210	271	166	60	40	40	98
RT5_AT68_SL	141589	303	467	710	40	121	213	166	60	40	40	94
RT5_AT24_SH	147504	302	488	944	48	160	414	684	90	10	8	100
RT5_AT46_SH	148216	302	491	840	50	124	339	681	90	10	10	100
RT5_AT68_SH	150028	304	494	837	52	191	344	681	90	10	10	98
RT5_AT24_CL	140484	302	465	696	38	112	203	223	45	55	21	100
RT5_AT46_CL	141109	302	467	696	40	95	203	239	41	59	18	99
RT5_AT68_CL	141066	301	469	668	41	112	174	272	36	64	14	94
RT5_AT24_CH	146437	301	487	787	46	123	229	785	78	22	4	100
RT5_AT46_CH	146891	300	490	757	48	180	231	846	72	28	3	99
RT5_AT68_CH	147777	302	489	839	50	133	244	892	68	32	2	96
RT3_AT24_BC	138752	301	461	668	36	102	174	111	0	100	97	100
RT3_AT46_BC	139211	302	461	699	36	97	206	110	0	100	100	96
RT3_AT68_BC	139931	303	462	691	37	121	197	110	0	100	100	89
RT3_AT24_DL	138972	302	460	680	35	109	186	164	0	100	57	99
RT3_AT46_DL	138625	301	461	668	35	109	217	182	0	100	47	99
RT3_AT68_DL	140845	304	463	727	39	99	234	214	0	100	37	94
RT3_AT24_DH	139003	302	460	668	35	98	174	213	0	100	36	100
RT3_AT46_DH	138542	301	460	713	39	117	220	271	0	100	23	97
RT3_AT68_DH	141163	304	464	693	39	120	200	323	0	100	19	90
RT3_AT24_SL	140139	301	466	690	38	113	196	211	47	53	51	100
RT3_AT46_SL	140592	302	466	746	38	89	252	210	48	52	52	99
RT3_AT68_SL	140854	303	465	763	38	210	270	210	48	52	52	94
RT3_AT24_SH	147101	302	487	835	48	124	309	725	85	15	15	100
RT3_AT46_SH	147309	301	489	855	49	184	329	725	85	15	15	100
RT3_AT68_SH	150110	305	492	867	52	151	354	726	85	15	15	98
RT3_AT24_CL	140735	302	466	707	39	109	214	264	38	62	36	99
RT3_AT46_CL	139069	299	465	676	38	112	182	281	35	65	30	99
RT3_AT68_CL	141476	303	467	668	39	133	174	313	32	68	26	95
RT3_AT24_CH	145111	298	487	827	47	133	287	825	74	26	9	100
RT3_AT46_CH	147544	301	490	837	48	182	311	884	69	31	7	98
RT3_AT68_CH	147604	300	492	829	52	122	324	931	60	35	6	95
RT1_AT24_BC	138557	301	460	675	35	99	181	331	0	100	100	100
RT1_AT46_BC	139108	302	461	657	36	131	164	331	0	100	100	99
RT1_AT68_BC	139318	302	461	667	36	101	174	332	0	100	100	92
RT1_AT24_DL	139232	302	461	708	36	107	215	399	0	100	83	99
RT1_AT46_DL	138863	301	461	759	36	116	216	428	0	100	77	99
RT1_AT68_DL	139622	303	461	709	36	109	215	466	100	0	71	93
RT1_AT24_DH	136786	297	461	675	36	117	182	465	0	100	71	100
RT1_AT46_DH	139382	301	463	679	38	138	185	539	0	100	61	97
RT1_AT68_DH	139285	302	461	656	36	122	162	594	0	100	56	91
RT1_AT24_SL	136008	294	463	707	36	120	210	431	23	77	77	99
RT1_AT46_SL	139204	301	462	690	35	118	196	431	23	77	77	98
RT1_AT68_SL	139368	300	465	740	37	88	247	432	23	77	77	92
RT1_AT24_SH	146426	301	486	795	46	118	243	946	65	35	35	100
RT1_AT46_SH	148955	303	492	788	51	178	232	944	65	35	35	98
RT1_AT68_SH	148593	303	490	792	49	225	267	946	65	35	35	96
RT1_AT24_CL	139063	299	465	716	38	97	223	499	20	80	66	99
RT1_AT46_CL	139228	299	466	719	38	135	218	527	19	81	63	98
RT1_AT68_CL	140828	301	468	698	40	133	205	565	18	82	59	92
RT1_AT24_CH	144751	298	486	781	45	125	217	1076	57	43	31	100
RT1_AT46_CH	146716	300	489	796	48	158	285	1152	53	47	29	10
RT1_AT68_CH	146254	299	489	773	49	188	256	1204	51	49	27	930

C.2 Performance Indicators from the Weighted Approach

Scenario	Performance Indicators											
	Tot. time	$\lvert\mathcal{V}^T\rvert$	Avg. time	Max. time	Avg. waiting	Max. waiting	Max. tot. waiting	Number of rolls	Rolls with reused schedule (%)	Rolls with new schedule (%)	Rolls due to RT (%)	Rolls to optimality (%)
RT5_AT24_BC	138478	302	459	670	33	108	175	69	0	100	84	100
RT5_AT46_BC	139053	303	459	667	34	87	174	66	0	100	100	100
RT5_AT68_BC	142004	306	464	671	39	109	177	66	0	100	100	89
RT5_AT24_DL	139315	303	460	677	35	107	174	123	0	100	35	99
RT5_AT46_DL	138646	302	459	668	34	91	174	140	0	100	30	100
RT5_AT68_DL	140447	304	462	710	37	111	216	173	0	100	23	97
RT5_AT24_DH	138969	302	460	668	35	108	174	174	0	100	18	100
RT5_AT46_DH	139342	302	461	725	36	113	231	233	0	100	11	97
RT5_AT68_DH	138727	302	459	661	35	121	167	281	0	100	7	95
RT5_AT24_SL	140962	304	464	708	36	101	215	169	59	41	34	100
RT5_AT46_SL	141315	304	465	715	37	108	222	166	60	40	40	99
RT5_AT68_SL	141957	304	467	744	39	130	250	166	60	40	40	96
RT5_AT24_SH	147358	303	486	845	46	134	305	683	90	10	8	100
RT5_AT46_SH	147344	303	486	739	46	124	188	682	90	10	10	100
RT5_AT68_SH	146630	301	487	787	46	135	281	681	90	10	10	99
RT5_AT24_CL	140998	303	465	716	38	116	223	223	45	55	21	100
RT5_AT46_CL	140266	302	464	732	37	114	239	239	41	59	18	100
RT5_AT68_CL	141406	304	465	715	38	110	222	272	36	64	14	97
RT5_AT24_CH	146907	302	486	816	46	297	323	787	78	22	4	100
RT5_AT46_CH	146696	301	487	796	47	182	224	848	72	28	3	99
RT5_AT68_CH	148727	304	489	800	49	146	277	893	69	31	2	97
RT3_AT24_BC	139122	302	461	690	36	107	196	111	0	100	97	99
RT3_AT46_BC	139442	303	460	740	35	127	243	110	0	100	100	98
RT3_AT68_BC	140716	305	461	691	36	120	197	110	0	100	100	94
RT3_AT24_DL	139173	302	461	708	36	92	215	164	0	100	57	99
RT3_AT46_DL	138489	302	459	698	34	110	194	182	0	100	47	99
RT3_AT68_DL	140222	304	461	668	36	115	174	214	0	100	37	94
RT3_AT24_DH	139178	302	461	699	36	97	206	213	0	100	36	100
RT3_AT46_DH	138998	302	460	668	35	108	174	271	0	100	23	96
RT3_AT68_DH	139590	303	461	683	36	99	190	323	0	100	19	93
RT3_AT24_SL	140857	303	465	695	37	129	191	211	47	53	51	100
RT3_AT46_SL	140131	302	464	715	36	101	222	210	48	52	52	100
RT3_AT68_SL	141692	303	468	714	40	93	221	210	48	52	52	95
RT3_AT24_SH	146808	303	485	798	44	200	298	726	85	15	15	100
RT3_AT46_SH	147463	302	488	847	47	142	317	725	85	15	15	99
RT3_AT68_SH	148946	304	490	785	49	155	274	726	85	15	15	98
RT3_AT24_CL	140284	302	465	716	37	98	223	264	38	62	36	88
RT3_AT46_CL	140392	302	465	725	37	211	231	281	35	65	30	99
RT3_AT68_CL	141292	304	465	707	37	85	210	313	32	68	26	97
RT3_AT24_CH	146200	302	484	764	44	123	225	825	74	26	9	100
RT3_AT46_CH	146664	301	487	750	46	165	257	885	58	42	7	99
RT3_AT68_CH	146747	302	486	771	46	146	265	933	65	35	6	97
RT1_AT24_BC	139048	303	459	690	34	108	196	331	0	100	100	99
RT1_AT46_BC	139177	303	459	657	34	114	164	331	0	100	100	98
RT1_AT68_BC	140041	304	461	799	35	130	306	332	0	100	100	93
RT1_AT24_DL	139744	303	461	708	36	110	215	399	0	100	83	99
RT1_AT46_DL	138854	302	460	655	35	95	161	428	0	100	77	100
RT1_AT68_DL	140423	304	462	668	37	115	174	466	0	100	71	94
RT1_AT24_DH	139116	302	461	676	36	107	182	465	0	100	71	100
RT1_AT46_DH	139871	303	462	668	36	113	174	539	0	100	61	96
RT1_AT68_DH	139786	304	460	725	35	166	226	594	0	100	56	92
RT1_AT24_SL	140854	304	463	675	36	106	181	431	23	77	77	100
RT1_AT46_SL	140491	303	464	710	36	126	213	431	23	77	77	98
RT1_AT68_SL	141328	304	465	715	37	125	222	432	23	77	77	94
RT1_AT24_SH	145685	302	482	744	42	120	230	946	65	35	35	100
RT1_AT46_SH	146785	302	486	773	45	142	242	946	65	35	35	98
RT1_AT68_SH	147151	303	486	794	45	119	223	947	65	35	35	95
RT1_AT24_CL	140342	303	463	716	36	94	223	499	20	80	66	100
RT1_AT46_CL	140182	302	464	675	37	103	182	527	19	81	63	98
RT1_AT68_CL	141240	304	465	725	37	112	231	565	18	82	59	96
RT1_AT24_CH	145333	301	483	734	43	125	209	1077	57	43	31	100
RT1_AT46_CH	146806	301	488	768	46	215	275	1154	53	47	29	98
RT1_AT68_CH	144005	299	482	769	41	125	205	1204	51	49	27	96

Appendix D Hypothesis Testing

This appendix outlines how the statistical tests regarding significance of relative measures have been conducted. Section D.1 explains the testing procedure, while Section D.2 presents the test results.

D.1 Testing Procedure

All test results are generated using a *paired difference t-test*, where the null hypothesis is that there is no difference in means from a case x to the base case. Since the approach is similar for all performance indicators tested, the approach is explained in light of adjusted average traversing time for combined case high in one specific scenario.

Let μ_{CH} denote the mean of adjusted average time for the combined case high, and μ_{BC} denote the mean of the average time for the base case. Let μ_d denote the difference of these means. The hypotheses can then be stated as:

$$H_0: \quad \mu_d = \mu_{BC} - \mu_x = 0$$
$$H_1: \quad \mu_d = \mu_{BC} - \mu_x \neq 0$$

The test was conducted only on the ships that were fully traversed in both cases. Let d_i denote the difference between adjusted traversing time and traversing time in the base case for ship i. For all ships that are fully traversed in both cases d_i is calculated as shown in Equation (48).

$$d_i = \text{Adj. trav. time} - \text{Trav. time base case} = (\text{Waiting time} - \text{Waiting time base case}) \tag{48}$$

It is assumed that d_i is normally distributed. Thus the test statistic defined in Equation (49) is t-distributed with $\left|\mathcal{V}^T\right| - 1$ degrees of freedom (Here $\left|\mathcal{V}^T\right|$ is the set of ships that are fully traversed in both cases).

$$t = \frac{\frac{1}{|\mathcal{V}^T|}\sum_{i \in |\mathcal{V}^T|} d_i - \mu_d}{\frac{S}{\sqrt{|\mathcal{V}^T|}}} \tag{49}$$

All test were carried in the programming software MATLAB. Note that d_i is defined equally when testing waiting and adjusted traversing due to Equation (48), thus corresponding test will have the same result.

D.2 Test Results

The test results are shown below.

Scenario	P-value	Confidence lower bound	Confidence upper bound	t-statistic	Degrees of freedom	Standard error	Confidence level
RT1_AT24_DH	0.80	-3.58	4.66	0.26	295	35.99	20%
RT1_AT24_DL	0.49	-2.62	5.43	0.69	299	35.42	51%
RT1_AT46_DH	0.24	-1.72	6.89	1.18	299	37.92	76%
RT1_AT46_DL	0.63	-2.80	4.60	0.48	299	32.59	37%
RT1_AT68_DH	0.86	-4.48	3.75	-0.17	299	36.20	14%
RT1_AT68_DL	0.83	-4.87	3.90	-0.22	300	38.67	17%
RT3_AT24_DH	0.66	-5.47	3.46	-0.44	299	39.27	34%
RT3_AT24_DL	0.69	-4.93	3.28	-0.40	300	36.19	31%
RT3_AT46_DH	0.83	-4.98	3.98	-0.22	300	39.47	17%
RT3_AT46_DL	0.90	-4.14	3.64	-0.13	300	34.29	10%
RT3_AT68_DH	0.34	-2.37	6.87	0.96	302	40.86	66%
RT3_AT68_DL	0.45	-2.67	5.99	0.75	301	38.25	55%
RT5_AT24_DH	0.46	-5.96	2.68	-0.75	299	38.03	54%
RT5_AT24_DL	0.67	-4.19	2.71	-0.42	301	30.47	33%
RT5_AT46_DH	0.12	-0.92	7.76	1.55	301	38.33	88%
RT5_AT46_DL	0.22	-1.54	6.67	1.23	298	36.07	78%
RT5_AT68_DH	0.36	-2.25	6.22	0.92	299	37.25	64%
RT5_AT68_DL	0.22	-1.62	6.93	1.22	301	37.75	78%

Table 15: Test results from paired t-tests, testing if the change in average traversing time compared to base case is significant for delay cases

Scenario	P-value	Confidence lower bound	Confidence upper bound	t-statistic	Degrees of freedom	Standard error	Confidence level
RT1_AT24_DH	0.79	-3.54	4.65	0.27	295	35.82	21%
RT1_AT24_DL	0.47	-2.55	5.51	0.72	299	35.47	53%
RT1_AT46_DH	0.22	-1.60	7.00	1.24	299	37.82	78%
RT1_AT46_DL	0.71	-2.97	4.33	0.37	299	32.11	29%
RT1_AT68_DH	0.97	-4.19	4.00	-0.04	299	36.04	3%
RT1_AT68_DL	0.92	-4.59	4.14	-0.10	300	38.50	8%
RT3_AT24_DH	0.69	-5.37	3.56	-0.40	299	39.29	31%
RT3_AT24_DL	0.72	-4.85	3.36	-0.36	300	36.21	28%
RT3_AT46_DH	0.12	-0.88	7.93	1.57	300	38.86	88%
RT3_AT46_DL	0.91	-4.11	3.66	-0.11	300	34.22	9%
RT3_AT68_DH	0.30	-2.17	7.07	1.04	302	40.88	70%
RT3_AT68_DL	0.40	-2.48	6.19	0.84	301	38.29	60%
RT5_AT24_DH	0.46	-5.91	2.70	-0.73	299	37.90	54%
RT5_AT24_DL	0.68	-4.17	2.73	-0.41	301	30.49	32%
RT5_AT46_DH	0.13	-0.95	7.69	1.53	301	38.17	87%
RT5_AT46_DL	0.22	-1.55	6.61	1.22	298	35.84	78%
RT5_AT68_DH	0.32	-2.07	6.37	1.00	299	37.12	68%
RT5_AT68_DL	0.19	-1.40	7.12	1.32	301	37.63	81%

Table 16: Test results from paired t-tests, testing if the change in average waiting time compared to base case is significant

Scenario	P-value	Confidence lower bound	Confidence upper bound	t-statistic	Degrees of freedom	Standard error	Confidence level
RT1_AT24_SH	0.00	6.25	15.54	4.62	299	40.87	100%
RT1_AT24_SL	0.49	-2.20	4.54	0.68	292	29.31	51%
RT1_AT46_SH	0.00	10.44	20.52	6.04	301	44.51	100%
RT1_AT46_SL	0.70	-3.58	2.42	-0.38	299	26.39	30%
RT1_AT68_SH	0.00	7.96	17.84	5.14	300	43.58	100%
RT1_AT68_SL	0.49	-2.30	4.78	0.69	297	31.03	51%
RT3_AT24_SH	0.00	7.47	16.47	5.23	299	39.61	100%
RT3_AT24_SL	0.20	-1.26	6.10	1.29	299	32.39	80%
RT3_AT46_SH	0.00	8.54	17.75	5.62	300	40.60	100%
RT3_AT46_SL	0.36	-2.18	6.00	0.92	301	36.12	64%
RT3_AT68_SH	0.00	9.87	20.46	5.64	302	46.82	100%
RT3_AT68_SL	0.54	-2.66	5.09	0.62	301	34.25	46%
RT5_AT24_SH	0.00	6.58	16.19	4.66	301	42.41	100%
RT5_AT24_SL	0.34	-1.93	5.61	0.96	301	33.27	66%
RT5_AT46_SH	0.00	12.95	21.54	7.90	301	37.93	100%
RT5_AT46_SL	0.00	2.15	9.68	3.09	302	33.30	100%
RT5_AT68_SH	0.00	10.78	20.69	6.24	303	43.94	100%
RT5_AT68_SL	0.06	-0.16	7.20	1.88	302	32.56	94%

Table 17: Test results from paired t-tests, testing if the change in Adjusted average traversing time and average waiting time compared to base case is significant for slowness cases

Scenario	P-value	Confidence lower bound	Confidence upper bound	t-statistic	Degrees of freedom	Standard error	Confidence level
RT1_AT24_CH	0.00	4.52	14.26	3.80	296	42.65	100%
RT1_AT24_CL	0.19	-1.16	5.94	1.33	297	31.12	81%
RT1_AT46_CH	0.00	7.51	17.57	4.91	298	44.20	100%
RT1_AT46_CL	0.15	-1.09	7.15	1.45	297	36.15	85%
RT1_AT68_CH	0.00	7.94	17.87	5.11	296	43.49	100%
RT1_AT68_CL	0.09	-0.56	7.72	1.70	298	36.35	91%
RT3_AT24_CH	0.00	6.74	17.36	4.47	295	46.42	100%
RT3_AT24_CL	0.17	-1.20	6.94	1.39	299	35.79	83%
RT3_AT46_CH	0.00	8.01	17.85	5.18	299	43.28	100%
RT3_AT46_CL	0.29	-1.84	6.19	1.07	298	35.24	71%
RT3_AT68_CH	0.00	9.06	20.66	5.04	299	51.05	100%
RT3_AT68_CL	0.31	-1.90	5.94	1.01	300	34.56	69%
RT5_AT24_CH	0.00	4.58	14.05	3.87	300	41.75	100%
RT5_AT24_CL	0.56	-2.76	5.07	0.58	300	34.54	44%
RT5_AT46_CH	0.00	9.40	19.38	5.67	299	43.94	100%
RT5_AT46_CL	0.00	2.50	10.28	3.23	301	34.36	100%
RT5_AT68_CH	0.00	8.43	17.97	5.45	301	42.12	100%
RT5_AT68_CL	0.04	0.25	8.38	2.09	300	35.84	96%

Table 18: Test results from paired t-tests, testing if the change in Adjusted average traversing time and average waiting time compared to base case is significant for combined cases

Appendix E Waiting Constrained Model

This appendix gives a brief overview of the Waiting Constrained Model (WCM) developed in Hilstad and Skjaeveland (2017). First the benefits and challenges of using the WCM is briefly discussed in Section E.1 Then some new notation is introduced in Section E.2, before the mathematical model is given in Section E.3.

E.1 Benefits and challenges of the WCM

A direct mathematical consequence of introducing limits on waiting time is that the set of conflicting ships can be drastically reduced. The reason is that ships now have an upper limit on traversing time, and thus it can be predefined which ships that may have a potential conflict. This decreases both the number binary variables and the number of constraints. Thus, the WCM allows for solutions to be found significantly faster than the model formulated in A.2. The exact procedure for removing these variables and constraints are explained in Hilstad and Skjaeveland (2017). Furthermore, the upper limit on traversing time makes it possible to reduce the size of big M using the following formula:

$$M_{ijp} = \begin{cases} \max\{\overline{T}_{jp-1} - \underline{T}_{ip}, \quad \overline{T}_{ip+1} - \underline{T}_{jp}\}, & \{i,j\} \in \mathcal{C}_p^O, \quad i \in \mathcal{V}^E, \quad j \in \mathcal{V}^W \\ \max\{\overline{T}_{jp+1} - \underline{T}_{ip}, \quad \overline{T}_{ip-1} - \underline{T}_{jp}\}, & \{i,j\} \in \mathcal{C}_p^O, \quad j \in \mathcal{V}^E, \quad i \in \mathcal{V}^W \\ \max\{\overline{T}_{jp} + H_{ji} - \underline{T}_{ip}, \quad \overline{T}_{ip} + H_{ij} - \underline{T}_{jp}\}, & \{i,j\} \in \mathcal{C}_p^A \end{cases} \tag{50}$$

Even though the WCM introduces a lot of benefits, there are some limitations. Unlike the formulation used throughout this thesis, the WCM is not able to guarantee a feasible solution to the problem. Thus, in order to incorporate the WCM in a simulation-optimization framework where feasibility is required, it is necessary to soften the waiting time constraints in some way. One way of doing this is introduce a penalty in the objective function that punishes high waiting times. Although this seems to be an easy way to avoid the feasibility issue, it does also remove the added benefit of upper limit on traversing time. How the WCM performs in a simulation-optimization framework is left as an area of further research.

E.2 Notation

All new notation used in the mathematical formulation of WCM is given beneath. The reader is referred to Appendix A.1 for other notation.

Parameters :

$\overline{W}_i^{TS}$	Waiting time capacity in total for ship $i \in \{\mathcal{V} \mid TGN_i \leq 5\}$
$\overline{W}_i^{TB}$	Waiting time capacity in total for ship $i \in \{\mathcal{V} \mid TGN_i = 6\}$
$\overline{W}_i^{IS}$	Waiting time capacity in each siding for ship $i \in \{\mathcal{V} \mid TGN_i \leq 5\}$
$\overline{W}_i^{IB}$	Waiting time capacity in each siding for ship $i \in \{\mathcal{V} \mid TGN_i = 6\}$

E.3 Mathematical formulation of the WCM

The compact mathematical model of the WCM is given beneath:

$$\min \quad \sum_{i\in\mathcal{V}}\sum_{p\in\mathcal{S}} w_{ip} + \sum_{i\in\mathcal{V}}\sum_{p\in\mathcal{P}} s_{ip} + \sum_{i\in\mathcal{V}^E} (t_{i1}-E_i) + \sum_{i\in\mathcal{V}^W} \left(t_{i|P|}-E_i\right) \tag{51a}$$

$$\text{s.t.} \quad t_{ip} + s_{ip} - t_{ip+1} = 0 \qquad i\in\mathcal{V}^E, \quad p\in\mathcal{T} \tag{51b}$$

$$t_{ip} + s_{ip} + w_{ip} - t_{ip+1} = 0 \qquad i\in\mathcal{V}^E, \quad p\in\mathcal{S}\backslash\{|S|\} \tag{51c}$$

$$t_{ip} + s_{ip} - t_{ip-1} = 0 \qquad i\in\mathcal{V}^W, \quad p\in\mathcal{T} \tag{51d}$$

$$t_{ip} + s_{ip} + w_{ip} - t_{ip-1} = 0 \qquad i\in\mathcal{V}^W, \quad p\in\mathcal{S}\backslash\{1\} \tag{51e}$$

$$t_{ip} + H_{ij} - t_{jp} \leq (1-z_{ijp})\, M_{ijp} \qquad \{i,j\}\in\mathcal{C}_p^A, \quad p\in\mathcal{P} \tag{51f}$$

$$t_{jp} + H_{ji} - t_{ip} \leq z_{ijp} M_{ijp} \qquad \{i,j\}\in\mathcal{C}_p^A, \quad p\in\mathcal{P} \tag{51g}$$

$$t_{ip} + s_{ip} - t_{jp} \leq (1-z_{ijp})\, M_{ijp} \qquad \{i,j\}\in\mathcal{C}_p^O, \quad p\in\mathcal{P} \tag{51h}$$

$$t_{jp} + s_{jp} - t_{ip} \leq z_{ijp} M_{ijp} \qquad \{i,j\}\in\mathcal{C}_p^O, \quad p\in\mathcal{P} \tag{51i}$$

$$z_{ijp+1} - z_{ijp} \leq 0 \qquad \{i,j\}\in\mathcal{C}_p^A \cap \mathcal{V}^E, \quad p\in\mathcal{T} \tag{51j}$$

$$z_{ijp-1} - z_{ijp} \leq 0 \qquad \{i,j\}\in\mathcal{C}_p^A \cap \mathcal{V}^W, \quad p\in\mathcal{T} \tag{51k}$$

$$\sum_{p\in\mathcal{S}} w_{ip} + (t_{i1}-E_i) \leq \overline{W}_i^T \qquad i\in\mathcal{V}^E \tag{51l}$$

$$\sum_{p\in\mathcal{S}} w_{ip} + \left(t_{i|\mathcal{P}|}-E_i\right) \leq \overline{W}_i^T \qquad i\in\mathcal{V}^W \tag{51m}$$

$$t_{i1} - E_i \leq \overline{W}_i^I \qquad i\in\mathcal{V}^E \tag{51n}$$

$$t_{i|\mathcal{P}|} - E_i \leq \overline{W}_i^I \qquad i\in\mathcal{V}^W \tag{51o}$$

$$w_{ip} \leq \overline{W}_i^I \qquad i\in\mathcal{V}, \quad p\in\mathcal{P} \tag{51p}$$

$$s_{ip} \leq \overline{X}_{ip} \qquad i\in\mathcal{V}, \quad p\in\mathcal{P} \tag{51q}$$

$$s_{ip} \geq \underline{X}_{ip} \qquad i\in\mathcal{V}, \quad p\in\mathcal{P} \tag{51r}$$

$$t_{ip} \geq \underline{T}_{ip} \qquad i\in\mathcal{V}, \quad p\in\mathcal{P} \tag{51s}$$

$$w_{ip} \geq 0 \qquad i\in\mathcal{V}, \quad p\in\mathcal{S} \tag{51t}$$

$$s_{ip} \geq 0 \qquad i\in\mathcal{V}, \quad p\in\mathcal{P} \tag{51u}$$

$$z_{ijp} \in \{0,1\} \qquad \{i,j\}\in\mathcal{C}_p, \quad p\in\mathcal{P} \tag{51v}$$

Appendix F Weighted Average Transfer Fee

Table 20 shows the total transfer fee for ships of different TGN. It also shows the realtive frequency of the different ships.

TGN	Transfer fee (Euro)	Relative frequency
1	1727	1%
2	2042	4%
3	3154	43%
4	4949	24%
5	6328	24%
6	7902	2%

Table 20: Transfer fee and relative frequency for ships of different TGN in the Kiel Canal. All data from und Schifffahrtsamt Kiel-Holtenau (2018)

The calculation of weighted average transfer fee is carried out as shown in equation (52)

$$\text{Weighted average transfer fee} = \sum_{TGN} \text{Transfer fee}\,(TGN) \cdot \text{Relative frequency}\,(TGN) \tag{52}$$