Scheduling two-way Ship Traffic for the Kiel Canal: Model, Extensions and a Matheuristic

Frank Meisel[a]*, Kjetil Fagerholt[b]

[a] *Faculty of Business, Economics, and Social Science, Christian-Albrechts-University Kiel, Germany*

[b] *Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway*

Abstract

The Kiel Canal is an artificial waterway of about 100 kilometers that connects the North Sea and the Baltic Sea. It allows ships to save several hundred kilometers of travel distance compared with going around the Jutland Peninsula (Denmark). Since the canal contains several narrow segments where large ships cannot pass each other, it needs to be decided on which ships have to wait in the wider siding segments to ensure a fast and safe passage of all ships. With this paper, several new optimization models are proposed for this traffic managing problem, which include variable ship speeds, capacities of siding segments, and limits for waiting times of ships. All model variants capture the relevant traffic rules and safety requirements with the goal to minimize the total transit time of ships. A matheuristic is proposed for solving the problem quickly. Experiments on real world data confirm the excellent performance of the heuristic and identify the potentials for providing high quality service to ships.

Keywords: Two-way ship traffic, Kiel Canal, mixed integer programs, matheuristic, scheduling

---

1. Introduction

The Kiel Canal is an artificial waterway that connects the North Sea and the Baltic Sea, see Figure 1. It received its name from the city of Kiel, which is located where the canal enters the Baltic Sea. The canal is one of the busiest waterways in the world with 30.000 to 40.000 ships traversing it every year. Depending on their ports of origin and destination, ships can save on average 463 kilometers compared with going around the Jutland Peninsula (Denmark), cf. WSV (2018). Exemplary routes between Rotterdam and Stockholm are shown in Figure 1. The shortcut through the Kiel Canal can effect a substantial reduction of travel time. It might also come at lower total travel cost if - depending on fuel price levels - savings in bunker cost outweigh the transit charge that has to be paid for using the canal, see Heitmann et al. (2013).

The canal is a two-way waterway meaning that ships going east-west and ships going west-east

*corresponding author, tel.: +49 431 880 1530, address: Olshausenstr. 40, 24098 Kiel, Germany

Email addresses: meisel@bwl.uni-kiel.de (Frank Meisel[a]), kjetil.fagerholt@ntnu.no (Kjetil Fagerholt[b])

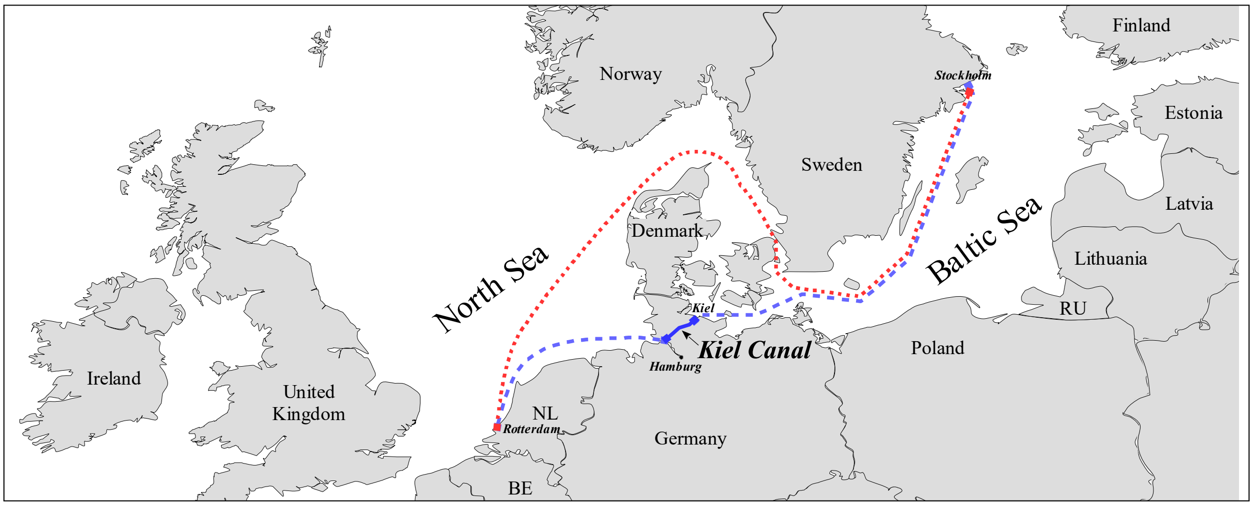

Figure 1: Northern Europe and the Kiel Canal

can use it at the same time. The layout of the canal is an alternating sequence of narrow *transit segments* and wider *siding segments*. In transit segments, only sufficiently small ships traveling in opposite directions can pass each other, whereas in sidings ships of any size can pass each other. Sidings also provide limited waiting areas where ships can wait and be overtaken by other ships traveling in the same direction. This leads to a traffic management problem, which is to schedule travel times and waiting times of ships so as to minimize the total transit time of all ships. The achieved transit times have a crucial impact on the attractiveness for ship operators to send their ships through the canal instead of going around Jutland. To offer an attractive service to ships of all kinds, the traffic managers strive for a fair traffic management that treats all ships equally.

Optimizing traffic decisions for the Kiel Canal has been investigated so far by Lübbecke (2015) and Lübbecke et al. (2018). They provided a mixed integer programming (MIP) formulation and heuristics for a basic version of the problem. The contributions of this paper are threefold:

- The base problem and MIP model of Lübbecke (2015) and Lübbecke et al. (2018) are extended by 1) deciding on speeds of ships rather than assuming constant given speeds, 2) restricting waiting times to guarantee maximum transit times for ships, and 3) capturing limited capacities of siding segments. All model variants meet relevant traffic rules and safety regulations.
- A new matheuristic is proposed for quickly solving the problem in all its variations.
- A comprehensive computational evaluation is conducted using real traffic data, which demonstrates the effectiveness of the proposed matheuristic. A number of experiments with varying waiting time limits for the ships are performed, providing important managerial insights for the traffic management for the Kiel Canal.

Although this study is dedicated to the traffic management for the Kiel Canal, several ideas for the modeling and the matheuristic presented here can be applied to the management of other inland waterways or to related problems like single track rail scheduling.

The outline of the paper is as follows. A literature review is provided in Section 2. Section 3 presents the basic optimization model and the three extensions. The matheuristic is described in Section 4. Section 5 presents the computational experiments. Section 6 concludes the paper.

2. Related Literature

Research on managing inland waterways often addresses the operations of locks. There, decisions are made on when to use a lock chamber for ships traveling up- or downstream and on which ships to handle in each lock-operation-cycle such that the maximum or total ship waiting time is minimized, see e.g. Campbell et al. (2007), Smith et al. (2009), Verstichel et al. (2015) and Passchyn et al. (2016). Although the Kiel Canal has locks at both ends too, the locks are operated independently of the traffic management, see Luy (2011). For this reason, lock operations planning provides input data to the traffic management in terms of ship entering times to the canal, but it is not in the scope of this research.

There is also some growing interest in managing the traffic flow of ships in a canal. A central issue is the problem of two-way traffic where ships traveling in opposite direction cannot pass each other in some narrow parts of the canal or in the whole canal. This research often considers the specific characteristics of a particular waterway. The early paper of Griffiths (1995) provides queuing models for the Suez Canal taking into account the canal's layout and the used convoy system. Ulusçu et al. (2009) provide an optimization model for determining the order in which ships go through the Strait of Istanbul. The Strait of Istanbul contains one narrow segment. If this segment is entered by a too large ship or by a ship carrying dangerous cargo, traffic in the opposite direction or even in both directions is suspended as long as this ship occupies the segment. The authors describe the corresponding traffic rules in detail and present a priority rule for ordering ships. Sluiman (2017) generalizes the research of Ulusçu et al. (2009) by providing an optimization model, a greedy-rule method, and improvement heuristics for scheduling ships for a waterway with a single narrow segment. They apply this approach to problem instances from the Strait of Istanbul and the Sunda Strait. They also consider the problem for waterways with a junction. Ship scheduling for a single bi-directional channel segment is also investigated in Zhang et al. (2017). A heuristic is proposed for building batches of ships for the two travel directions in order to minimize mean and maximum waiting times of ships with respect to safety restrictions. Lalla-Ruiz et al. (2018) present a MIP model and a Simulated Annealing heuristic for scheduling ships that have to pass the Yangtze river delta for accessing the ports of Shanghai. Contrasting the previous papers, the river delta offers several parallel waterways such that each ship must be assigned to exactly one waterway. Within each waterway, opposing ships can only meet if their total width does not exceed the width of the waterway. Since the waterways are operated in parallel, ships assigned to one waterway can be scheduled independently of the other waterways. The Yangtze river is furthermore investigated by Tan et al. (2018) who design a schedule for ships that visit different ports along that river. The ship traffic is subject to a dam with uncertain transit times due to lockage operations. The authors propose a mathematical model for minimizing bunker consumption and ship round-trip time under chance constraints for a punctual arrival. Righini (2016) investigate a network of bi-directional and one-way inland waterways in northern Italy. The focus in not on determining precise schedules for ships but on estimating the capacity of this waterway system. For this purpose, they present a network flow model that maximizes the flow of barges that can pass this system subject to capacities of waterway segments, traffic rules, and harbor capacities.

A further stream of research analyses ship traffic scheduling for navigation channels that connect ports to the open sea in combination with berth allocation decisions at the port facilities. Zhang et al. (2016) consider a single navigation channel where ships have to be scheduled such that a berth is available when they enter the port through the channel. A Simulated Annealing algorithm and a Genetic Algorithm are proposed for minimizing the total ship waiting time. Zhen et al. (2017) focus on berthing decisions where the navigation channel is subject to tide cycles. The tide cycles effect individual time windows for ships depending on their draft. A further constraint prescribes a maximum number of ships that can use the channel at a time. A column generation method is proposed for solving the problem. Jia et al. (2018) schedule in- and out-going ships for a terminal with respect to a limited anchorage area within the terminal. A Lagrangian relaxation heuristic minimizes penalties for tardy services and unsatisfied service requests. Corry and Bierwirth (2018) combine traffic management for a navigation channel with berth allocation, where berths are modeled as additional channel segments. The proposed model is based on a flow-shop problem with parallel machines. Tidal restrictions are considered as an extension to this problem. Constructive algorithms are proposed for solving the base problem and the extentions thereof. As the navigation channels considered in these papers do not have sidings there are no decisions involved on waiting times of ships within the channel but just on entering times of ships.

Contrasting the waterways studied in the above papers, the Kiel Canal consists of a *sequence* of narrow and wide segments where ships can wait in the wide segments. For this reason, interdependent ship scheduling decisions have to be made for the different segments of the canal. The corresponding planning problem has been investigated by Lübbecke (2015) and Lübbecke et al. (2018). These works provide a basic MIP model that describes the propagation of ship entering times and waiting times from one segment to the other and show that the problem is $\mathcal{NP}$-hard. They propose a labeling algorithm for scheduling ships under given and constant travel speed that is then embedded in a local search and a rolling horizon framework for solving large test instances with a 24 hours planning horizon. These heuristics can also handle capacities of sidings although this feature is not present in the MIP model. The paper at hand provides model formulations that also incorporate capacities of sidings, waiting time limits and speed decisions. Furthermore, a matheuristic is presented that can handle all these problem variants.

The problem considered here also has similarities to scheduling trains on a single rail track, where opposing trains can only meet at siding tracks or at stations, see Lusby et al. (2011) for a literature review. Works in this field like Szpigel (1973), Kraay et al. (1991), Kraay and Harker (1995), Higgins et al. (1996), Zhou and Zhong (2007), Castillo et al. (2011), Lamorgese and Mannino (2015) and Gafarov et al. (2015) propose exact and heuristic solution methods for various settings that differ in requirements on train arrival and departure times, dwell times at stations etc. Anyhow, ship traffic management for a two-way waterway differs from single track rail scheduling in that opposing ships are allowed to meet in transit segments under certain conditions. Furthermore, features and extensions considered in the paper at hand like ship-dependent safety distances, discretized speed decisions, capacity constraints for sidings, and waiting time limits are covered only partly in these works or do not apply there at all.

3. Modeling the Traffic Management Problem

3.1. Problem Setting

A formal description of the considered problem is provided below. The used notation is summarized in Table 1. The layout of the Kiel Canal is schematically sketched in Figure 2. The canal consists of 23 segments that are indexed here from 0 to 22. The segments alternate between wide sidings and narrow transit segments. The first and the last segment are sidings. Hence, all sidings have an even index number whereas transit segments have an odd index number. $\mathcal{E} = \{0, 1, 2, \ldots 22\}$ denotes the index set of all canal segments (edges), $\mathcal{S} = \{0, 2, 4, \ldots 22\}$ denotes the subset of sidings, and $\mathcal{T} = \mathcal{E} \backslash \mathcal{S} = \{1, 3, 5, \ldots 21\}$ denotes the subset of transit segments. The last siding of the canal is addressed by $\overline{s} = 22$. Each segment $s \in \mathcal{E}$ is characterized by its length l_s and by a so-called *passage number* p_s. The passage number reflects the width of a segment and is used for determining whether ships of particular sizes can meet in a segment. The sidings $s \in \mathcal{S}$ of the Kiel Canal have a passage number of $p_s = 12$ whereas the transit segments have

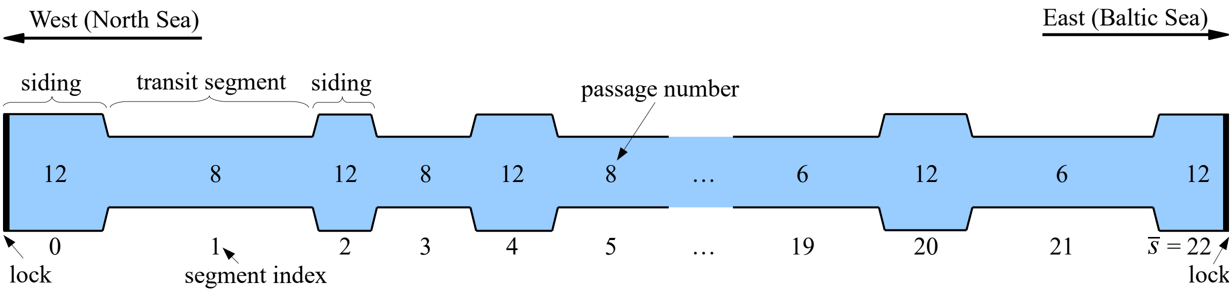

Figure 2: Schematic structure of the canal.

Table 1: Notation.

Canal data:	
$\mathcal{E}$	Set of all canal segments (edges), $\mathcal{E} = \{0, 1, 2, \ldots 22\}$
$\mathcal{S}$	Set of sidings, $\mathcal{S} = \{0, 2, 4, \ldots 22\}$ with $\overline{s} = 22$ as the last siding of the canal
$\mathcal{T}$	Set of transit segments, $\mathcal{T} = \mathcal{E} \backslash \mathcal{S} = \{1, 3, 5, \ldots 21\}$
l_s	Length of segment $s \in \mathcal{E}$
p_s	Passage number of segment s
Ship data:	
$\mathcal{V}$	Set of all ships
$\mathcal{V}^E$	Subset of ships traveling eastbound (from left (segment 0) to right (segment $\overline{s}$))
$\mathcal{V}^W$	Subset of ships traveling westbound (from right (segment $\overline{s}$) to left (segment 0))
h_i	Length of ship i
g_i	Traffic group number of ship i
$\overline{v}_i$	Default travel speed of ship $i \in \mathcal{V}$
$d_{i,s}$	Transit duration (travel time) of ship $i \in \mathcal{V}$ for traversing segment $s \in \mathcal{E}$, i.e. $d_{i,s} = \frac{l_s}{\overline{v}_i}$
ETA_i	Expected time of arrival of ship i
Safety requirements and conflict avoidance:	
$\mathcal{C}_s^O$	Set of pairs of opposing ships that might have a conflict in transit segment $s \in \mathcal{T}$
$\mathcal{C}_s^A$	Set of pairs of aligned ships that might have a conflict in transit segment $s \in \mathcal{T}$
$\gamma_{i,j}$	Safety distance that ship j has to keep if it runs behind ship i
$\Delta_{i,j,s}$	Least safety time required between ships i and j to enter segment s without a conflict
Decision variables:	
$t_{i,s}$	Time at which ship $i \in \mathcal{V}$ enters segment $s \in \mathcal{E}$
$w_{i,s}$	Waiting time of ship i in siding $s \in \mathcal{S}$
$z_{i,j,s}$	Binary, 1 if ship i gets priority over ship j in transit segment $s \in \mathcal{T}$, 0 if j gets priority over i
Notation for modelling speed decisions (Section 3.4):	
$\underline{v}_i$	Parameter, minimum speed level of ship i
$d_{i,s,v}$	Parameter, transit duration (travel time) of ship $i \in \mathcal{V}$ for traversing segment $s \in \mathcal{E}$ at speed level $v \in [\underline{v}_i, \overline{v}_i]$, i.e. $d_{i,s,v} = \frac{l_s}{v}$
$u_{i,s,v}$	Binary decision variable, 1 if ship i travels in segment $s \in \mathcal{E}$ at speed level $v \in [\underline{v}_i, \overline{v}_i]$
$d_{i,s}$	Continuous decision variable, transit duration (travel time) of ship $i \in \mathcal{V}$ for traversing segment $s \in \mathcal{E}$, $d_{i,s} \in \left[d_{i,s,\overline{v}_i}, d_{i,s,\underline{v}_i}\right]$. (Replaces former parameter $d_{i,s}$)
Notation for modelling waiting time limits (Section 3.5):	
$w_g^{\max}$	Maximum waiting time per siding for a ship with traffic group number g
$W_g^{\max}$	Maximum total waiting time for a ship with traffic group number g
$EET_{i,s}$	Earliest entering time of ship i at segment s
$LET_{i,s}$	Latest entering time of ship i at segment s
$EXT_{i,s}$	Earliest exiting time of ship i from segment s
$LXT_{i,s}$	Latest exiting time of ship i from segment s
Notation for modelling capacity of sidings (Section 3.6):	
γ	Default safety distance to be kept between two waiting ships
MIS_s	Set of all minimum infeasible subsets of ships for siding $s \in \mathcal{S}$
$y_{i,j,s}$	Binary decision variable, 1 if ship i ends waiting in siding s before ship j starts waiting in s

a passage number between 6 and 8. The canal has locks on both ends but no locks in-between. The total length of the canal is about 100 kilometers.

Let $\mathcal{V}$ denote the set of ships for which the passage through the canal needs to be planned. Subset $\mathcal{V}^E \subseteq \mathcal{V}$ refers to the set of ships traveling eastbound (from segment 0 to segment $\overline{s}$) and $\mathcal{V}^W = \mathcal{V} \backslash \mathcal{V}^E$ refers to the set of ships traveling westbound (from segment $\overline{s}$ to segment 0). The length of ship $i \in \mathcal{V}$ is denoted by h_i. According to its length, a ship also is assigned a so-called *traffic group number* $g_i \in [1, 6]$. The smallest ships, which have a length of around 50 meters, are assigned traffic group number 1. The largest ships that can use the canal have a length of 235 meters. They are assigned traffic group number 6. Ships $i \in \mathcal{V}$ with a traffic group number $g_i \leq 5$ have to pass the canal at a default travel speed of $\overline{v}_i = 15$ km/h whereas large ships of $g_i = 6$ have to travel at a lower speed of $\overline{v}_i = 12$ km/h for safety reasons and to protect the shoreline from heavy bow waves. The transit duration (travel time) of ship $i \in \mathcal{V}$ for traversing segment $s \in \mathcal{E}$ is then given by $d_{i,s} = \frac{l_s}{\overline{v}_i}$. For reasons of simplicity, accelerations and decelerations of the ships are ignored here, meaning that ships can instantaneously reach any prescribed speed, and that they can come to a full stop immediately, if they have to wait in a siding. Eventually, each ship i has an expected time of arrival ETA_i, which refers to the point in time when the ship exits the lock and enters its first segment within the canal. Since the lock operations are out of scope of this study and because lock operations are planned independently and ahead of the traffic management, it is assumed that ETA_i is given as an input to the problem considered here.

3.2. Safety Rules and Conflict Avoidance

For safety reasons and to avoid conflicts among ships, various rules have to be respected when scheduling ship traffic. These are rules for *opposed ships*, which are ships traveling in opposite directions, and rules for *aligned ships*, which are ships traveling into the same direction.

One rule is that two opposed ships i and j are not allowed to meet in transit segment s if the sum of their traffic group numbers exceeds the segment's passage number, i.e. if $g_i + g_j > p_s$. In such a situation either i or j has to wait in a siding segment while the other ship occupies transit segment s. Figure 3a illustrates such a situation. Here, the eastbound ship 2 with $g_2 = 3$ and the westbound ships 3 and 4 with $g_3 = 1$ and $g_4 = 3$ can simultaneously use the transit segment with passage number $p_s = 6$. The two large ships 1 and 5 with $g_1 = 5$ and $g_5 = 6$ have to wait in the sidings. Figure 3b presents the considered situation in a time-space diagram. Slanting lines represent movements of ships whereas vertical lines represent waiting times of ships in the sidings. Since ship 5 with traffic group number $g_5 = 6$ has a lower travel speed than the other ships, its slanting lines are steeper. The horizontal dotted line indicates the situation that is illustrated in Figure 3a. The time-space diagram reveals further that ship 1 waits until ship 4 leaves the transit segment and that ship 5 waits until ship 1 leaves this segment.

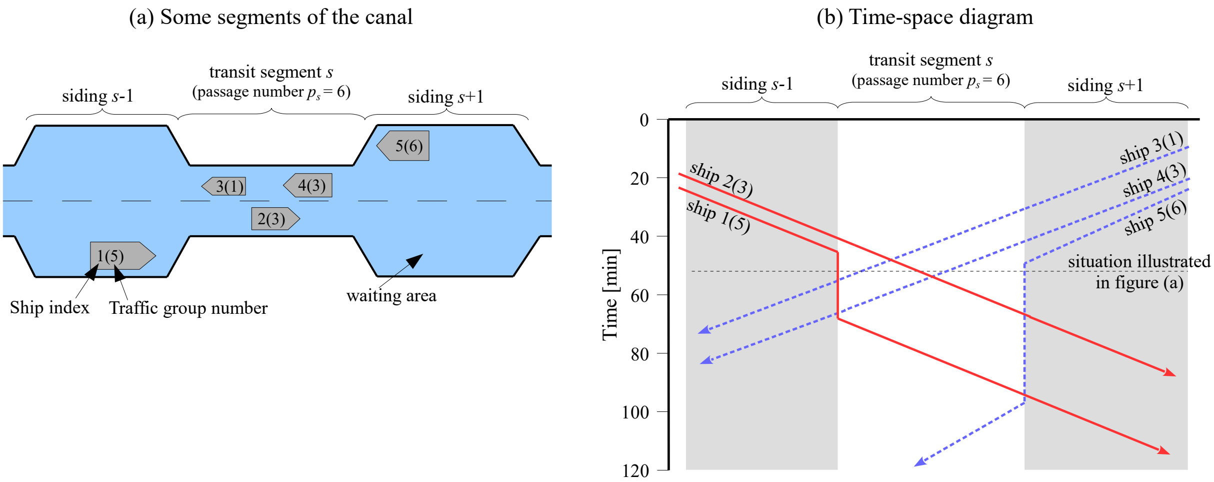

Figure 3: Illustration of traffic rules and time-space diagram representation.

For aligned ships the following rules apply. First, ships are allowed to overtake each other only in sidings. Second, ships have to keep a safety distance. The safety distance depends on the size of the ship that runs behind. More precisely, if a ship j travels behind a ship i, j's bow and i's stern have to be separated by 600 m if $g_j \leq 3$ and by 1000 m if $g_j > 3$. By adding half the ship lengths of i and j to the respective distance, a safety distance $\gamma_{i,j}$ is obtained that has to be kept between the mid-points of the two ships.

In order to meet these rules in the decision making, equation (1) defines the set of pairs of opposed ships $\mathcal{C}_s^O$ that might have a safety conflict in transit segment $s \in \mathcal{T}$ and equation (2) defines the set of pairs of aligned ships $\mathcal{C}_s^A$ that might have a safety conflict in transit segment $s \in \mathcal{T}$.

$$\mathcal{C}_s^O = \{(i,j) | i \in \mathcal{V}^E, j \in \mathcal{V}^W, g_i + g_j > p_s\} \tag{1}$$

$$\mathcal{C}_s^A = \{(i,j) | i,j \in \mathcal{V}^E, i < j\} \cup \{(i,j) | i,j \in \mathcal{V}^W, i < j\} \tag{2}$$

Conflicts among ships $(i,j) \in \mathcal{C}_s^O \cup \mathcal{C}_s^A$ can be avoided by appropriately separating the times at which these ships enter segment s. For this purpose, equation (3) computes the minimum *safety time* $\Delta_{i,j,s}$ that has to elapse between the entering of ship i and the subsequent entering of ship j at segment s. For situations where j enters the segment before i, a corresponding value $\Delta_{j,i,s}$ is computed by simply swapping the roles of i and j in the formula.

$$\Delta_{i,j,s} = \begin{cases} d_{i,s} + \frac{1}{2} \cdot h_i / \overline{v}_i + \frac{1}{2} \cdot h_j / \overline{v}_j & \text{if } (i,j) \in \mathcal{C}_s^O \\ \gamma_{i,j} / \overline{v}_i & \text{if } (i,j) \in \mathcal{C}_s^A \wedge \overline{v}_i \geq \overline{v}_j \\ \gamma_{i,j} / \overline{v}_i + d_{i,s} - d_{j,s} & \text{if } (i,j) \in \mathcal{C}_s^A \wedge \overline{v}_i < \overline{v}_j \end{cases} \tag{3}$$

The first case in equation (3) handles situations of opposing ships $(i,j) \in \mathcal{C}_s^O$ where i enters segment s before ship j, see corresponding illustration in Figure 4a. In such a situation, ship

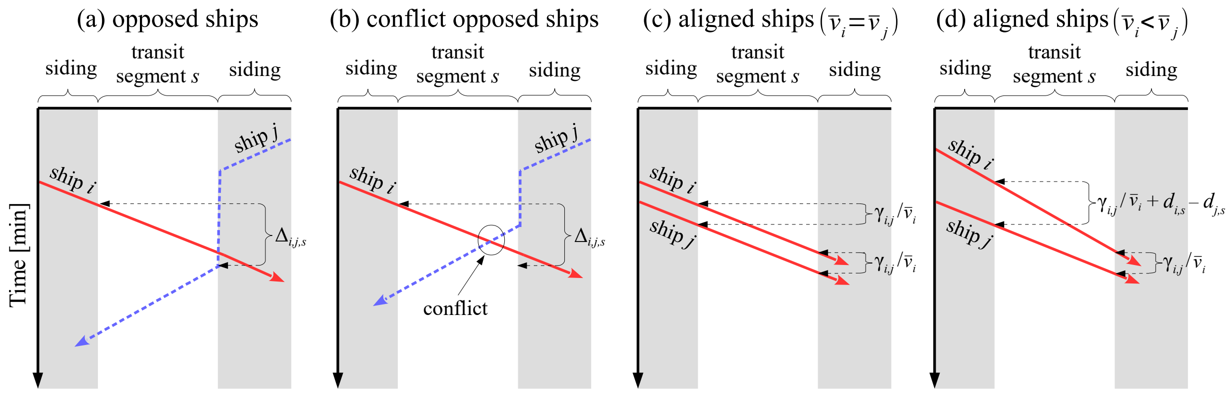

Figure 4: Temporal separation of ships.

j cannot enter the segment as long as ship i is within that segment, meaning that the entering times have to be separated by at least the time $d_{i,s}$ that ship i requires for traversing the segment. Since ship positions refer to the mid-points of ships, the temporal distances $\frac{1}{2} \cdot h_i / \overline{v}_i$ and $\frac{1}{2} \cdot h_j / \overline{v}_j$ are added to ensure that there is sufficient time to have the stern section of ship i leave the segment and to have the bow section of ship j enter the segment. If the entering times are not separated as described, both ships would meet in the segment which is infeasible, see Figure 4b. The second case in equation (3) handles aligned ships where the preceding ship i is at least as fast as ship j. If both ships travel at the same speed, their entering times must be separated at least by the safety time $\gamma_{i,j} / \overline{v}_i$, see Figure 4c. This ensures that the ships are sufficiently separated also at the end of the segment, which avoids any conflicts within segment. Such a separation of entering times also ensures a safe passage if i is faster than j ($\overline{v}_i > \overline{v}_j$). Then, the separation of both ships even enlarges while travelling through the segment. Anyhow, if the preceding ship i is slower than ship j, their entering must be separated by at least $\gamma_{i,j} / \overline{v}_i + d_{i,s} - d_{j,s}$ time units according to the third case in equation (3), see Figure 4d. This ensures that both ships, although getting closer to each other within the segment due to their different speeds, are still separated by at least $\gamma_{i,j} / \overline{v}_i$ time units when reaching the end of the segment.

3.3. Base Model Formulation

This section describes the mathematical model for the basic version of the planning problem that reflects the current decision making policy of the canal authority, see also Lübbecke et al. (2018). The task is to schedule the given ships in such a way that they transit the canal free of conflicts within minimum total transit time. For this purpose, it needs to be decided for each ship $i \in \mathcal{V}$ and each segment $s \in \mathcal{E}$ at what time $t_{i,s}$ the ship enters the segment. This decision determines the waiting time $w_{i,s}$ of ship i in siding $s \in \mathcal{S}$. Eventually, for all pairs of ships $(i,j) \in \mathcal{C}_s^A$ and $(i,j) \in \mathcal{C}_s^O$ that might have a conflict in transit segment $s \in \mathcal{T}$, it needs to be decided, which ship gets priority over the other. This is modelled using binary decision variable $z_{i,j,s}$, where $z_{i,j,s} = 1$ if ship i gets priority over ship j in segment s and $z_{i,j,s} = 0$ if ship j gets priority over ship i in this segment. The optimization model is formulated as follows:

$$\min \rightarrow TTT = \sum_{i\in\mathcal{V}^E} (t_{i,\overline{s}} + d_{i,\overline{s}} - ETA_i) + \sum_{i\in\mathcal{V}^W} (t_{i,0} + d_{i,0} - ETA_i) \tag{4}$$

$$t_{i,0} = ETA_i \qquad i \in \mathcal{V}^E \tag{5}$$

$$t_{i,\overline{s}} = ETA_i \qquad i \in \mathcal{V}^W \tag{6}$$

$$t_{i,s} + d_{i,s} = t_{i,s+1} \qquad i \in \mathcal{V}^E, s \in \mathcal{T} \tag{7}$$

$$t_{i,s} + d_{i,s} = t_{i,s-1} \qquad i \in \mathcal{V}^W, s \in \mathcal{T} \tag{8}$$

$$t_{i,s} + d_{i,s} + w_{i,s} = t_{i,s+1} \qquad i \in \mathcal{V}^E, s \in \mathcal{S}\backslash\{\overline{s}\} \tag{9}$$

$$t_{i,s} + d_{i,s} + w_{i,s} = t_{i,s-1} \qquad i \in \mathcal{V}^W, s \in \mathcal{S}\backslash\{0\} \tag{10}$$

$$t_{i,s} + \Delta_{i,j,s} \leq t_{j,s} + \mathrm{M} \cdot (1 - z_{i,j,s}) \qquad s \in \mathcal{T}, (i,j) \in \mathcal{C}_s^A \cup \mathcal{C}_s^O \tag{11}$$

$$t_{j,s} + \Delta_{j,i,s} \leq t_{i,s} + \mathrm{M} \cdot z_{i,j,s} \qquad s \in \mathcal{T}, (i,j) \in \mathcal{C}_s^A \cup \mathcal{C}_s^O \tag{12}$$

$$w_{i,s} \geq 0 \qquad i \in \mathcal{V}, s \in \mathcal{S} \tag{13}$$

$$z_{i,j,s} \in \{0,1\} \qquad s \in \mathcal{T}, (i,j) \in \mathcal{C}_s^A \cup \mathcal{C}_s^O \tag{14}$$

The objective (4) is to minimize the *total transit time* (TTT) of all ships, where eastbound ships leave the canal via segment $\overline{s}$ at time $t_{i,\overline{s}} + d_{i,\overline{s}}$ whereas westbound ships leave the canal via segment 0 at time $t_{i,0} + d_{i,0}$. Since the canal authority strives for an equal treatment of all ships, the transit times are not weighted by ship size or the like. Constraints (5) and (6) set the entering time of the first canal segment visited by a ship to the expected time of arrival. Constraints (7) derive for each eastbound ship $i \in \mathcal{V}^E$ and each transit segment $s \in \mathcal{T}$ the entering time at the subsequent segment $s+1$. Constraints (8) do the same for westbound ships, where segment s is followed by segment $s-1$. Accordingly, Constraints (9) and (10) propagate the entering time of siding $s \in \mathcal{S}$ to the subsequent segment $s+1$ (respectively, $s-1$), where not just the travel time $d_{i,s}$ but also the waiting time $w_{i,s}$ delays the entering of the next segment. The temporal distances between potentially conflicting ships are ensured by Constraints (11) and (12). If ship i gets priority over ship j ($z_{i,j,s} = 1$), Constraints (11) ensure that i enters segment s before j. If j gets priority over i ($z_{i,j,s} = 0$), Constraints (12) ensure that j enters first. In these constraints, M denotes a sufficiently large positive scalar. Constraints (13) and (14) define the domains of the decision variables.

3.4. Model Extension 1: Variable Ship Speed

In the base model, all ships $i \in \mathcal{V}$ travel at their default speed $\overline{v}_i$ in each segment of the canal, which reflects the current practice at the canal. In order to make the traffic management more flexible, the travel speeds of ships might be taken into account for the decision making. While the current default speed $\overline{v}_i$ can serve as a maximum speed limit for ship i, a lower bound $\underline{v}_i$ needs to be defined for the speed to avoid too slow movement where ships lose their maneuverability. It is assumed here that $\underline{v}_i$ and $\overline{v}_i$ are both integer values and that the speed chosen for a ship i and a segment s is an integer value in the range $[\underline{v}_i, \overline{v}_i]$. The reason for only considering integer speeds is that the chosen speeds have to be communicated to the captains for steering their ships. Hence continuous values are impractical. A corresponding binary decision variable $u_{i,s,v}$ takes value 1 if ship i travels in segment s at speed $v \in [\underline{v}_i, \overline{v}_i]$. Furthermore, the parameter $d_{i,s,v} = \frac{l_s}{v}$ denotes the travel time of ship i when traversing segment s at speed $v \in [\underline{v}_i, \overline{v}_i]$. Eventually, $d_{i,s}$ is now a continuous variable that reflects the actual travel time of ship i in segment s under the chosen speed.

The speed decisions are included into the base model by adding Constraints (15)-(17). Constraints (15) ensure that exactly one speed is chosen for each ship and each segment. Constraints (16) compute the resulting transit times $d_{i,s}$, which will take a value in the range $\left[d_{i,s,\overline{v}_i}, d_{i,s,\underline{v}_i}\right]$. Constraints (17) define the domains of the new decision variables.

$$\sum_{v=\underline{v}_i}^{\overline{v}_i} u_{i,s,v} = 1 \qquad i \in \mathcal{V}, s \in \mathcal{E} \tag{15}$$

$$d_{i,s} = \sum_{v=\underline{v}_i}^{\overline{v}_i} u_{i,s,v} \cdot d_{i,s,v} \qquad i \in \mathcal{V}, s \in \mathcal{E} \tag{16}$$

$$u_{i,s,v} \in \{0,1\} \qquad i \in \mathcal{V}, s \in \mathcal{E}, v \in [\underline{v}_i, \overline{v}_i] \tag{17}$$

A similar approach for speed optimization has been proposed by Andersson et al. (2015). They use continuous variables $0 \leq u_{i,s,v} \leq 1$ instead of binary ones as there is no need to end up with integer speed values in their problem. Anyhow, no matter whether speed variables are continuous or binary, Constraints (15) and (16) are required in both cases to obtain a linear formulation for computing ship travel times as a function of the chosen speed. For this reason, the modeling approach proposed here serves two purposes: (1.) for practical reasons to obtain integer speeds that ease communication of traffic managers and captains and (2.) for technical reasons to obtain a linear formulation of the speed modeling.

Furthermore, since safety times are depending on the speeds of ships, they are no longer preprocessed as a parameter matrix $\Delta_{i,j,s}$ by equation (3) but are now computed within the corresponding constraints. For this purpose, former constraint set (11) is now expressed by Constraints (18), (19), and (20), which individually addresses the three cases of the $\Delta_{i,j,s}$-definition in equation (3).

$$t_{i,s} + d_{i,s} + \sum_{v=\underline{v}_i}^{\overline{v}_i} u_{i,s,v} \cdot \tfrac{1}{2} \cdot h_i/v + \sum_{v=\underline{v}_j}^{\overline{v}_j} u_{j,s,v} \cdot \tfrac{1}{2} \cdot h_j/v \leq t_{j,s} + \mathrm{M} \cdot (1 - z_{i,j,s}) \quad s \in \mathcal{T}, (i,j) \in \mathcal{C}_s^O \tag{18}$$

$$t_{i,s} + \sum_{v=\underline{v}_i}^{\overline{v}_i} u_{i,s,v} \cdot \gamma_{i,j}/v \leq t_{j,s} + \mathrm{M}\cdot(1 - z_{i,j,s}) \quad s \in \mathcal{T}, (i,j) \in \mathcal{C}_s^A \quad (19)$$

$$t_{i,s} + \sum_{v=\underline{v}_i}^{\overline{v}_i} u_{i,s,v} \cdot \gamma_{i,j}/v + d_{i,s} - d_{j,s} \leq t_{j,s} + \mathrm{M}\cdot(1 - z_{i,j,s}) \quad s \in \mathcal{T}, (i,j) \in \mathcal{C}_s^A \quad (20)$$

Constraints (18) ensure that if ship i gets priority over the *opposed* ship j in transit segment s (i.e. $z_{i,j,s} = 1$), then j cannot enter s before i has left, where $d_{i,s}$ is now a dependent decision variable. Constraints (19) and (20) capture the cases of aligned ships. Constraints (19) separate the entering times of aligned ships according to the safety distance $\gamma_{i,j}$ and the speed chosen for the prioritized ship i. This requirement is valid, no matter whether ship i is faster than ship j or not, which is why this constraint is stated for all $(i,j) \in \mathcal{C}_s^A$. If ship i is slower than ship j, the entering times at the segment have to be further separated by the time span $d_{i,s} - d_{j,s}$, see Figure 4d, which is ensured through Constraints (20). Note that in case that ship i is actually faster than ship j, $d_{i,s} - d_{j,s}$ is negative and Constraints (20) are not binding whereas Constraints (19) do. Therefore, also Constraints (20) are stated for all $(i,j) \in \mathcal{C}_s^A$ and there is no need to test for the actual relationship of the speeds of ships i and j.

Eventually, the optimization model with variable speeds is obtained by adding to the base formulation (4)-(14) the new Constraints (15)-(17), replacing Constraints (11) with Constraints (18)-(20), and replacing Constraints (12) with three similar constraints that handle the case $z_{i,j,s} = 0$.

3.5. Model Extension 2: Waiting Time Limits

The traffic planners at the Kiel Canal also attempt to meet some soft service requirements in order to achieve a fair and acceptable service quality for all ships. According to these soft rules, the total waiting time of a ship in the canal should not exceed three hours for ships with traffic group numbers 1 to 5 and two hours for ships of traffic group 6. Furthermore, the waiting time per siding should not exceed one and a half hour for ships of traffic group 1 to 5 and one hour for ships of traffic group 6. These requirements are mentioned in Lübbecke et al. (2018) but not considered any further. In the following, they are added as hard constraints to the model. Bounding the maximum waiting time of a ship has the attractive side effect that a *maximum transit time* is then guaranteed for the ship. This can be used as a service promise to further promote the attractiveness of the canal. Additionally, these requirements give a tighter model. Clearly, too tight requirements might lead to infeasibility of the planning problem. Therefore, an experiment later analyzes how 'costly' it is to strictly follow these rules.

The following notation is used here. For a ship with traffic group number g, $w_g^{\max}$ is the maximum waiting time per siding and $W_g^{\max}$ is the maximum total waiting time (both measured in minutes). The default settings are $w_g^{\max} = 90$ and $W_g^{\max} = 180$ for $g \in [1, .., 5]$ as well as $w_6^{\max} =$

60 and $W_6^{\max} = 120$ for $g = 6$. The service requirements are enforced by Constraints (21) and (22).

$$w_{i,s} \leq w_{g_i}^{\max} \qquad i \in \mathcal{V}, s \in \mathcal{S} \tag{21}$$

$$\sum_{s \in \mathcal{S}} w_{i,s} \leq W_{g_i}^{\max} \qquad i \in \mathcal{V} \tag{22}$$

Furthermore, with waiting times being bounded by $w_{g_i}^{\max}$ and $W_{g_i}^{\max}$ and travel speeds being bounded to $[\underline{v}_i, \overline{v}_i]$, an earliest entering time $EET_{i,s}$, a latest entering time $LET_{i,s}$, an earliest exiting time $EXT_{i,s}$ and a latest exiting time $LXT_{i,s}$ can be computed for each ship i and each segment $s \in \mathcal{E}$. For an eastbound ship these values are computed as follows:

$$EET_{i,s} = ETA_i + \sum_{s'=0}^{s-1} d_{i,s',\overline{v}_i} \tag{23}$$

$$LET_{i,s} = ETA_i + \sum_{s'=0}^{s-1} d_{i,s',\underline{v}_i} + \min\{w_{g_i}^{\max} \cdot |\mathcal{S}_{<s}|, W_{g_i}^{\max}\} \tag{24}$$

$$EXT_{i,s} = ETA_i + \sum_{s'=0}^{s} d_{i,s',\overline{v}_i} \tag{25}$$

$$LXT_{i,s} = ETA_i + \sum_{s'=0}^{s} d_{i,s',\underline{v}_i} + \min\{w_{g_i}^{\max} \cdot |\mathcal{S}_{\leq s}|, W_{g_i}^{\max}\} \tag{26}$$

Here, $\mathcal{S}_{<s}$ is the set of siding segments with an index strictly lower than s and $\mathcal{S}_{\leq s}$ is the set of siding segments with an index lower or equal to s. The min-functions in equations (24) and (26) compute the maximum waiting time that can occur for ship i before entering segment s and before exiting segment s, respectively. EET, LET, EXT, and LXT for westbound ships are computed in a similar fashion.

Using this information, the model can be tightened by restricting the time when ship $i \in \mathcal{V}$ enters segment $s \in \mathcal{E}$ to $EET_{i,s} \leq t_{i,s} \leq LET_{i,s}$. Furthermore, the ship will exit the segment within time window $[EXT_{i,s}, LXT_{i,s}]$. Together this defines a temporal corridor, see Figure 5a. Such corridors indicate whether or not ships can actually have a conflict in a transit segment. In Figures 5a and 5b, the opposing ships $i \in \mathcal{V}^E$ and $j \in \mathcal{V}^W$ cannot have a conflict in the considered segment as their corridors do not overlap, i.e. $LXT_i < EET_j$ (Figure 5a) and $EET_i > LXT_j$

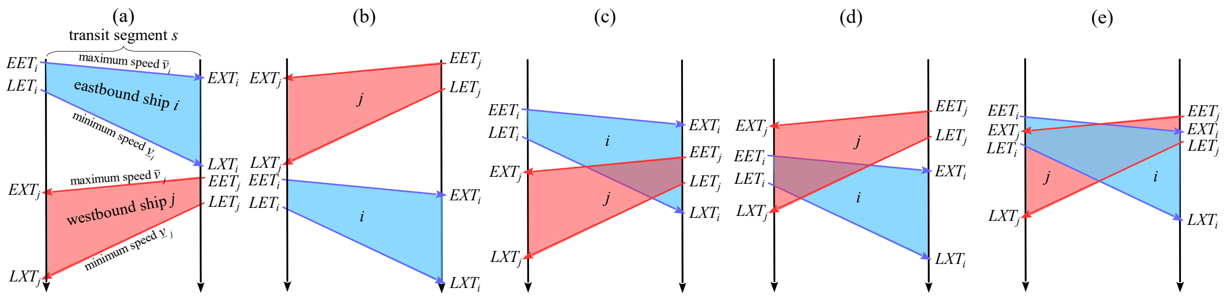

Figure 5: Five cases of entering and exiting time windows for two opposed ships i and j.

(Figure 5b). Such pairs (i, j) can be excluded from set $\mathcal{C}_s^O$, which eliminates the corresponding $z_{i,j,s}$-variable and Constraints (11) and (12) in the base model, respectively Constraints (18)-(20) in the speed model. If corridors overlap at one side (Figures 5c and d), the ships can have a conflict, but it is known a priori which ship has to get priority and the corresponding z-variable can be fixed. In Figure 5c, ship i definitely enters the segment before ship j leaves it ($LET_i < EXT_j$) and, thus, $z_{i,j,s} = 1$ is fixed. If $LET_j < EXT_i$, $z_{i,j,s} = 0$ is fixed, see Figure 5d. Only in case that corridors overlap at both sides, it is up to the optimization model to decide which ship to prioritize in this segment, see Figure 5e. Corresponding situations for aligned ships are sketched in Figure 6. In Figure 6a, ships i and j cannot have a conflict because of $LET_i < EET_j$ and $LXT_i < EXT_j$. Such pairs (i, j) are excluded from set $\mathcal{C}_s^A$, which avoids variables $z_{i,j,s}$ and corresponding constraints. If corridors overlap at one side as in Figure 6b, $z_{i,j,s} = 1$ is fixed as ship i has to enter the segment before ship j. For Figure 6c, $z_{i,j,s} = 0$ is fixed. If corridors overlap at both sides (Figure 6d) the optimization model decides on the ordering of ships.

Furthermore, tight big-M values can be derived for Constraints (11) and (12). Concerning Constraints (11), since $LET_{i,s}$ is an upper bound for $t_{i,s}$ and $EET_{j,s}$ is a lower bound for $t_{j,s}$, M can be specified by $M_{i,j,s} = LET_{i,s} + \Delta_{i,j,s} - EET_{j,s}$. In Constraints (12), M is replaced by $M_{j,i,s} = LET_{j,s} + \Delta_{j,i,s} - EET_{i,s}$. If the speed optimization from Section 3.4 is extended by the waiting time restrictions described here, M in Constraints (18), (19) and (20) can be set respectively to $M_{i,j,s} = LET_{i,s} + d_{i,s,\underline{v}_i} + \frac{1}{2} \cdot h_i/\underline{v}_i + \frac{1}{2} \cdot h_j/\underline{v}_j - EET_{j,s}$, $M_{i,j,s} = LET_{i,s} + \gamma_{i,j}/\underline{v}_i - EET_{j,s}$, and $M_{i,j,s} = LET_{i,s} + \gamma_{i,j}/\underline{v}_i + d_{i,s,\underline{v}_i} - d_{j,s,\overline{v}_j} - EET_{j,s}$.

3.6. Model Extension 3: Capacity of Sidings

A final extension of the model is to respect the limited capacity of sidings for hosting waiting ships. Since each siding has separate side tracks for the waiting of eastbound and westbound ships, see Figure 3a, the capacity of sidings needs to be respected for sets of aligned ships but not for sets of opposed ships. To formalize this, let K be some set of aligned ships. If the total length of ships in K plus some safety distance between the ships exceeds the length l_s of siding s, it is infeasible to have all ships from K wait in this segment at a same time. In order to

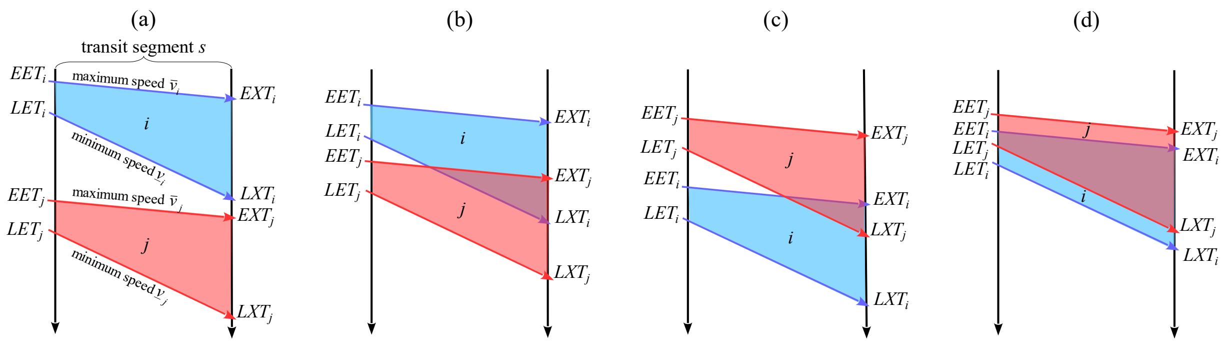

Figure 6: Four cases of entering and exiting time windows for two aligned ships i and j.

avoid such infeasibility, so-called *minimum infeasible subsets* (MIS) of ships are identified. An MIS is a set of aligned ships whose total length including safety distances exceeds the length of a given siding while there exists no strict subset that also has this property. Minimum infeasible subsets appear in many MIP models and solution methods, for example in Combinatorial Benders' Decomposition (see e.g. Codato and Fischetti (2006) for a general analysis and Verstichel et al. (2015) for an application to lock scheduling). They are also known as ‚minimal covers' in the context of knapsack cover inequalities for MIP models, see e.g. Gu et al. (2000). In the context of the traffic scheduling problem that is investigated here, an MIS is formally defined as follows: a set of aligned ships $K \subseteq \mathcal{V}^E$ (or $K \subseteq \mathcal{V}^W$) is an MIS for siding segment s if and only if $\sum_{i \in K} h_i + (|K| - 1) \cdot \gamma > l_s$ and $\sum_{i \in L} h_i + (|L| - 1) \cdot \gamma \leq l_s$ for all strict subsets $L \subset K$. In this formula, γ denotes the default safety distance that has to be kept between two waiting ships, which is here set to $\gamma = 50$ m. Then, by definition, if at least one ship from MIS K does not wait simultaneously with the other ships, there is no capacity violation from ships K at the considered segment. In order to completely avoid capacity violations for siding s, MIS_s denotes the set of *all* minimum infeasible subsets that exist for this segment.

To avoid capacity violations, binary decision variables $y_{i,j,s}$ are introduced for all $s \in \mathcal{S}, K \in MIS_s$ and $i, j \in K, i \neq j$. Variable $y_{i,j,s}$ takes value 1 if ship i's waiting time in siding s ends before ship j's waiting time. The following constraints are added to the model.

$$t_{i,s} + d_{i,s} + w_{i,s} \leq t_{j,s} + \mathrm{M} \cdot (1 - y_{i,j,s}) \qquad s \in \mathcal{S}, K \in MIS_s, i, j \in K, i \neq j \tag{27}$$

$$\sum_{i,j \in K | i \neq j} y_{i,j,s} \geq 1 \qquad s \in \mathcal{S}, K \in MIS_s \tag{28}$$

$$y_{i,j,s} \in \{0, 1\} \qquad s \in \mathcal{S}, K \in MIS_s, i, j \in K, i \neq j \tag{29}$$

Constraints (27) set the y-variables. The left hand side represents the point in time when ship i leaves siding s whereas the right hand side expresses the point in time when ship j enters the siding. Clearly, if i leaves the siding before j enters, the two ships do not wait simultaneously and $y_{i,j,s}$ can take value 1. The constraints do not consider the actual waiting periods of ships, because the model does not decide on whether a ship waits right after entering a siding, just before leaving it, or at any time in between. In other words, the waiting position of a ship is not determined in the model, which is different to Lübbecke et al. (2018) who also define a position for a waiting ship. This is done here because the waiting position of a ship may alter as it moves up when some other ship waiting in front leaves the siding. This helps to effectively utilize the siding capacity in practice. However, since Constraints (27) use an upper bound for the end of ship i's waiting and a lower bound for the begin of ship j's waiting in segment s, it is guaranteed that $y_{i,j,s} = 1$ if and only if these ships do not wait simultaneously. Constraints (28) ensure for each siding s and each MIS K that at least two ships are not waiting simultaneously, which avoids the capacity conflict.

Although only the MIS are considered here to keep the number of variables and constraints low, it is ensured that no conflict can occur for larger sets of ships. This is because *all* MIS of a segment are considered and each larger set of ships contains by definition at least one of these MIS. Figure 7 illustrates this for a siding and four ships. The example ignores detailed ship lengths and assumes that at most two ships fit into this siding. Therefore, $MIS_s = \{\{1,2,3\},\{1,2,4\},\{1,3,4\},\{2,3,4\}\}$. Avoiding simultaneous waiting of ships for all these MIS also avoids simultaneous waiting for the larger ship set $\{1,2,3,4\}$. Although the constraints are only defined for MIS, their number still grows exponentially in the number of ships, which limits the tractability of the model. For example, the test instances that are used for the experiments in Section 5 have about $4.9 \cdot 10^3$ MIS per instance with 20 ships, about $8.0 \cdot 10^4$ MIS per instance with 30 ships, and about $1.2 \cdot 10^6$ MIS per instance with 40 ships.

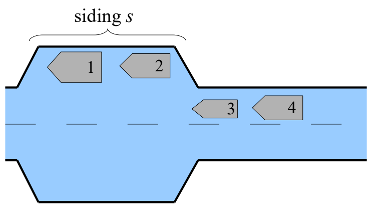

Figure 7: Capacity example.

4. Heuristic Solution Method

For solving the base problem without extensions, Lübbecke et al. (2018) propose a labeling algorithm that, for a given fixing of all z-variables, determines segment entering times and waiting times for all ships. The space of z-variables is explored by a local search method. Although the labeling algorithm is extended to cope with capacities of sidings, it seems difficult to generalize this method to handle further extensions like variable ship speeds that are investigated here. Therefore, ideas from single track train scheduling are adapted here to come up with a simple yet effective matheuristic that can easily cope with the various extensions presented in the previous sections. In single track train scheduling, a common solution approach is to first solve a relaxed problem that typically ignores all potential conflicts and then gradually resolve the occurring conflicts to obtain a feasible solution to the overall problem, see e.g. Szpigel (1973), Higgins et al. (1996), Kraay et al. (1991), Kraay and Harker (1995), Zhou and Zhong (2007), Castillo et al. (2011). For example, Higgins et al. (1996) and Zhou and Zhong (2007) solve a relaxed problem and then iteratively resolve the earliest conflict in a Branch-and-Bound scheme. A similar concept is adapted here within a matheuristic solution framework. The method can be controlled in terms of the number of conflicts resolved per iteration. Additional mechanisms are applied for speeding up the solution process.

What makes the presented model difficult to solve is the large number of binary z-variables and corresponding Constraints (11) and (12) that follow from sets $\mathcal{C}_s^A$ and $\mathcal{C}_s^O$. Therefore, the matheuristic fills the sets $\mathcal{C}_s^A$ and $\mathcal{C}_s^O$ on demand with just those pairs of ships that actually caused a conflict (infeasibility) in the current solution. This adds only those z-variables and constraints to the model that are indeed needed to obtain a feasible solution. More precisely, first, a relaxed model with empty sets $\mathcal{C}_s^A$ and $\mathcal{C}_s^O$ is solved using a commercial solver. Since there are no binary variables in this relaxed model, the resulting LP can be solved quickly but its solution typically contains many conflicts among ships. Afterwards, subsets of these conflicts are iteratively eliminated by adding the corresponding ship pairs to sets $\mathcal{C}_s^A$ and $\mathcal{C}_s^O$ and by resolving the problem until a feasible solution is reached. If siding capacities are to be considered, the algorithm also starts with empty sets MIS_s and fills them with those sets of ships that actually violated the capacity of a siding in a solution. This way, only those y-variables and Constraints (27) and (28) are added to the model that are relevant for obtaining a feasible solution. The heuristic is sketched in Algorithm 1.

Algorithm 1: Matheuristic

Input : Problem instance
Output: A feasible solution
1 Initialization: $\mathcal{C}_s^O \leftarrow \emptyset$, $\mathcal{C}_s^A \leftarrow \emptyset$, $MIS_s \leftarrow \emptyset$ for all s
2 Solve the optimization model using a standard MIP solver.
3 Identify the set of $Conflicts$ that take place in the current solution.
4 **if** $Conflicts \neq \emptyset$ **then**
5     $EarliestConflicts \leftarrow$ the Λ earliest conflicts from set $Conflicts$
6   **foreach** $Conflict \in EarliestConflicts$ **do**
7   **if** $Conflict$ is a conflict of aligned ships i and j in segment s **then**
            $\mathcal{C}_s^A \leftarrow \mathcal{C}_s^A \cup \{(i,j)\}$
8   **if** $Conflict$ is a conflict of opposed ships i and j in segment s **then**
            $\mathcal{C}_s^O \leftarrow \mathcal{C}_s^O \cup \{(i,j)\}$
9   **if** $Conflict$ is a capacity conflict of a set K of ships in siding s **then**
            $MIS_s \leftarrow MIS_s \cup \{K\}$
10   **end**
11   Fix feasible part of current solution up to the earliest conflict.
12   Repeat from Step 2.
13 **end**
14 **return** obtained solution;

The procedure includes three mechanisms for speeding up the solution process:

1. The allowed runtime for solving the intermediate models in Step 2 is restricted, as it is sufficient to obtain a good heuristic solution quickly in this step. In the experiments, a runtime limit of five seconds is set for this step of the algorithm.
2. Instead of adding all observed conflicts to the corresponding sets, merely the Λ earliest conflicts are added, see Step 5. Parameter Λ therefore controls the lookahead of conflicts

that are to be resolved in the next iteration. It trades off the growth of the size of the model against the number of iterations the heuristic needs for obtaining a final feasible solution. The parameter has a significant impact on the runtime of the heuristic and, therefore, its tuning is subject of the experimental study.

3. A feasible part of the current solution is fixed in Step 11. This is done by first identifying the earliest conflict in the solution. Then, the time variables $t_{i,s}$ are fixed for those ships and segments that are traversed before this conflict time. This preserves the partial schedule when solving the optimization model again in Step 2. For ships with a capacity violation in siding s, entering times are not fixed for the two preceding segments as these ships might have to wait in the previous siding to resolve the capacity conflict.

Note that both the first and third speedup mechanism turn the procedure into a heuristic. Without these two mechanisms, the procedure would be an exact algorithm that delivers optimal solutions. For example, when setting $\Lambda = 1$ (i. e. only one infeasibility is resolved per iteration) and ignoring the two speedup mechanisms, the procedure would resemble the Branch-and-Bound schemes of Higgins et al. (1996) and Zhou and Zhong (2007). However, preliminary testing showed that especially the fixation of the partial solution is mandatory for achieving short runtimes and that values $\Lambda > 1$ yield a further acceleration of the solution process. Therefore, all three mechanisms are applied here, which makes the procedure a fast matheuristic. Eventually, the matheuristic can solve the base problem of Section 3.3, the three extensions for speed decisions, waiting time limits, and capacities of sidings (Sections 3.4-3.6) as well as combinations of these extensions by calling the respective optimization model in Step 2 of the algorithm.

Furthermore, with the ability to preserve partial solutions, the procedure can be applied in a rolling horizon fashion for the online version of the problem where ship arrivals become known to the traffic managers in the course of time. In such a setting, newly arriving ships would be simply added to set $\mathcal{V}$ and the matheuristic resolves the resulting conflicts in the subsequent iterations. Ships that have passed the canal would be removed from sets $\mathcal{V}$, $\mathcal{C}_s^A$, $\mathcal{C}_s^O$, and MIS_s. The objective function should then be to minimize the average transit time of ships rather than the total transit time as the number of considered ships changes in the course of time. For such an approach, one could experiment with various ship arrival rates, with different replanning intervals for the heuristic, or how far in advance ships should announce their arrival to the Kiel Canal. Anyhow, such analysis is out of scope of this paper and left for future research.

5. Computational Study

Experiments are conducted to investigate the solvability of the presented models, to analyze the capabilities of the heuristic, and to derive managerial recommendations regarding the traffic management at the Kiel Canal. The MIP solver ILOG Cplex 12.8 is used for solving the optimization models directly and for solving the subproblems in Step 2 of the matheuristic. The heuristic has been implemented in Java. All experiments are performed on a PC with 3.60 GHz and 32 GB RAM. The next subsection describes the test instances used for the experiments, followed by the presentation of the computational results and findings.

5.1. Description of Test Instances

For the experiments, the canal authority provided real ship data, which states for each ship i in the data set: the travel direction (eastbound or westbound), the ETA_i at which the ship exited the lock and entered the first segment of the canal, and the traffic group number g_i. The data does not contain ship lengths h_i. These parameters are set to 50, 75, 130, 145, 205, and 235 meters for traffic groups 1 to 6, which are realistic values. Test instances of different size (in terms of number of ships) and time horizon length are extracted from the provided data. More precisely, ten small instances with $|\mathcal{V}| = 20$ ships each are derived, ten medium sized instances with 30 ships each, and ten large instances with 40 ships each. The ships in these instances arrive within a 'wide' time horizon of 12 hours and are refered to as sets '20w', '30w' and '40w'. Furthermore, two sets of instances with ten instances each are derived, where 20 ships and 30 ships arrived within a 'dense' time horizon of just six hours. These instance sets are named '20d' and '30d'. There are no instances '40d' as the provided data does not contain such a large number of ships arriving within so short time. Eventually, a set called '110' is generated with ten extremely large instances that contain on average 110 ships arriving within a 24 hour time horizon. This is far beyond what is relevant for a planner in practice, but Lübbecke et al. (2018) considered such instance size for their local-search approach and, thus, it is tested here if the matheuristic can handle this too. The instances are publicly available at `www.scm.bwl.uni-kiel.de/de/forschung/research-data`.

The canal authority also provided the relevant canal data like length l_s and passage number p_s for segments $s \in \mathcal{E}$. For the speed extension, the minimum speed level is set to $\underline{v}_i = 10$ km/h for all ships. All further parameters are set to the values mentioned in the earlier sections of this paper.

5.2. Computational Experiment 1: Exact Solution of Optimization Models

In a first experiment, the base model and its extended versions are solved for all test instances using Cplex with a runtime limit of one hour (3600 seconds) per instance. Table 2 shows the obtained results. Each row reports aggregated results for the ten instances of an instance set. The first column indicates the instance set. The next four columns report for the base model from Section 3.3 the following measures: the number of integer feasible solutions (f) found for the ten instances within the restricted runtime, the number of optimal solutions (o), the average total transit time TTT (i.e. the objective function value), and the average computation time (*cpu*) in seconds consumed by the solver. The next parts of the table report the same measures

Table 2: Results of Cplex runs for all instance sets.

	base				speed				service				capacity				speed+service			
set	f	o	TTT	cpu	f	o	δ	cpu	f	o	δ	cpu	f	o	δ	cpu	f	o	δ	cpu
20w	10	10	8231.7	73	10	9	0.0	372	10	10	0.0	4	10	9	0.0	383	10	10	0.0	32
30w	10	9	12278.3	678	10	8	0.0	1159	10	10	0.0	32	10	5	0.0	1866	10	9	0.0	593
40w	10	0	16732.1	3600	10	0	1.1	3600	10	1	0.0	3441	0	0	-	3600	10	0	0.5	3600
20d	10	9	8229.0	366	10	9	0.0	393	10	10	0.0	129	10	9	0.0	401	10	10	0.0	212
30d	10	7	12414.2	1615	10	6	0.1	1860	10	7	0.0	1118	8	1	-	3251	10	7	0.0	1405
110	10	0	51719.3	3600	10	0	56.7	3600	9	0	-	3600	0	0	-	3600	2	0	-	3600
avg.	10	6	18267.4	1655	10	5	9.7	1830	10	6	0.0	1387	6	4	0.0	2184	9	6	0.1	1574

for the extended models, where 'speed' refers to the model with speed decisions (Section 3.4), 'service' refers to the service-oriented model with waiting time limits (Section 3.5), and 'capacity' refers to the model with capacity constraints for sidings (Section 3.6). For all these models, not the total transit times TTT are reported but the relative deviation of transit times observed under the extended model compared with the transit times of the base model. This measure is computed as $\delta = (TTT_{extension} - TTT_{base})/TTT_{base}$ and reported in %. The table finally shows the results for a model with combined speed extension and service extension. Other combinations of model extensions are not considered due to the weak performance of the capacity extension that is described below.

The results for instance sets '20w', '30w', and '40w' are analyzed first. The base model finds feasible solutions consistently for all instances within the allowed runtime. Clearly the total transit times TTT increase with the size of instances, where average waiting times per ship (not shown in Table 2) range from 15 minutes for instances '20w' to 22 minutes for instances '40w'. Optimal solutions are frequently found for instances of small and medium size only but not for the large instances in set '40w'. This is also reflected by the average cpu time which is about one minute for the small instances whereas all large instances consume the allowed runtime of 3600 seconds. The optimality gaps of those instances that were not solved to optimality range from 0.1% to 2.7% (not shown in Table 2). The results for the speed-model show that this extension complicates the solution process with higher average cpu times and a δ-gap of 1.1% for the instances '40w'. A test with the linear relaxation of the $u_{i,s,v}$-speed variables as proposed by Andersson et al. (2015) and discussed in Section 3.4 was also conducted but did not reveal an advantage. For example, the average runtime for instances '30w' decreased from 1159 seconds to 1008 seconds when replacing the binary u-variables by continuous ones but it increased from 1860 seconds to 2219 seconds for instances '30d'. It seems that the complication does not come from the binary character of the u-variables but from the large number of these variables and those constraints (15) and (16) that are inevitable for a linear formulation of ship travel times. For the service extension, the following observations are made. Although one could expect that adding additional service constraints might deteriorate the average solution quality, perfect solution quality ($\delta = 0.0$) and proven optimality is achieved for all instances in '20w' and '30w'. For instances 'w40', even one optimal solution is found which was not the case for the base model. Furthermore, the service-model comes along with comparably short runtimes, mainly due to the model tightening discussed in Section 3.5. As an example, if these tightenings are not applied, solving the instances of set '30w' takes on average 249 seconds. When fixing the z-variables as described in Section 3.5, the runtimes reduce to 117 seconds per instance. If also the domains of variables and the big-M values are modified as discussed in Section 3.5, the runtimes are as low as 32 seconds, which is the value reported in Table 2.

Eventually, limited waiting times and, thus, maximum transit times can be guaranteed for all ships in these instances without difficulty. Therefore, the service extension actually provides two advantages: guaranteed maximum transit times for ships and faster solution of the planning problem. In contrast, adding the capacity extension deteriorates solvability of the problem drastically. It causes higher computation times and less instances are solved to optimality. Even worse, for the instances '40w' it not even obtains any feasible solutions as the number of minimum infeasible subsets gets too large.

Finally, the model that combines the speed extension and the service extension is considered. The idea here is that purposefully slowing down ships could help in meeting the waiting time limits. However, a lower solution quality is obtained compared to the sole service extension together with larger runtimes. Hence, in this experiment, using the speed extension provides no advantage. Interestingly, the total transit times do hardly change when adding any of the model extensions. The reason is the high substitutability of ships in this traffic management problem. For example, if a ship has a very long waiting time in a segment that is then forbidden in the service extension, an alternative solution is found by the solver where waiting times are distributed among different ships without affecting the total transit time. Similarly, if there are too many ships waiting in a siding, the capacity extension lets some of these ships already wait in an earlier siding, again not affecting the objective function.

Considering the further instance sets, it is observed that the 'dense' instances in '20d' and '30d' show results almost identical to '20w' and '30w' but at somewhat larger cpu times. It seems that this property has only little impact on solving the problem. For the extremely large instances in set '110', which are actually irrelevant from a practical perspective, Cplex runs into its boundaries. Although the base model can be solved to feasibility for all 10 instances, not a single optimal solution is obtained here. With the speed extension, feasible solutions are obtained for all instances but with an extremely large δ-gap. Cplex then fails in obtaining feasible solutions for some or even all instances under all remaining model extensions.

5.3. Computational Experiment 2: Performance of the Matheuristic

The next experiment addresses the performance of the matheuristic. The heuristic takes the parameter Λ that controls the number of conflicts that are resolved in each iteration of the procedure. A first test evaluates the proper setting of this parameter, as it impacts the growth in the model size and the number of iterations needed by the heuristic to obtain a feasible solution. For this purpose, Λ is varied in the range [1,50] and the instance set '40w' is solved once for each value under the base model (i.e. without any extensions). Figure 8 shows the resulting δ-gap with regards to the Cplex-solutions for model 'base' reported in Table 2 together with the runtime (*cpu*) of the heuristic. It can be seen that the δ-gap declines a little with increasing Λ. It is close to 1% for very low values of Λ and approaches 0% for larger values of Λ. The cpu times first decrease and later increase. This is because very low values of Λ give a large number of iterations whereas high values require only few iterations but with a complex model that includes many binary variables. There are also some values where δ is merely 0.3% (e.g., $\Lambda = 37$), but this comes at relatively high cpu time and, also, the $\delta-$plot shows that the noise in the heuristic solution quality is stronger than the trend in improving solution quality. For this reason, $\Lambda = 20$ is used in the following experiments as a compromise with low gaps and short cpu time.

Next, all instance sets are solved for the base problem, the individual extensions, and combinations of the extensions. The results are shown in Table 3. The results to the five instance sets '20w' to '30d' are analyzed first. Since the heuristic finds feasible solutions for all instances, column 'f' is omitted in the table. The only values reported are the δ-gap to the Cplex-solutions for model 'base' and the runtime of the heuristic. It is observed that the solution quality is very good and almost identical for all settings with very low gaps of $\delta \leq 1.0\%$. Also the average runtimes are extremely low with at most 1.8 seconds for configurations 'base', 'service', 'capacity' and 'service+capacity' and somewhat higher (up to 16.1 seconds) for configurations that include the speed extension. Interestingly, the heuristic can respect the capacities of sidings without difficulty, solving all instances feasibly, quickly, and with a low optimality gap. The same holds for

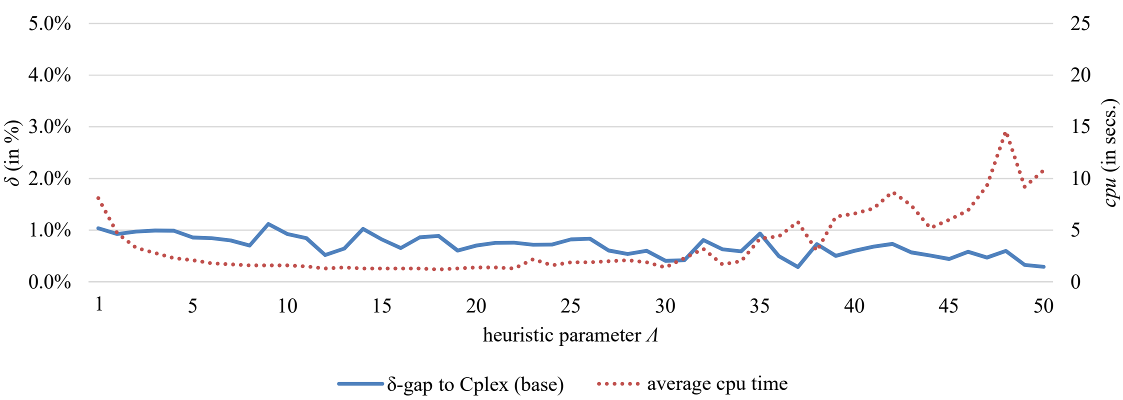

Figure 8: Results of pretest for heuristic parameter Λ with instances '40w'.

Table 3: Results of the matheuristic for all instance sets.

	base		speed		service		capacity		speed+ service		speed+ capacity		service+ capacity		speed+ service+ capacity	
set	δ	*cpu*	δ	*cpu*	δ	*cpu*	δ	*cpu*	δ	*cpu*	δ	*cpu*	δ	*cpu*	δ	*cpu*
20w	0.4	0.4	0.5	2.4	0.8	0.5	0.4	0.4	0.5	2.9	0.6	2.3	0.8	0.5	0.5	2.9
30w	0.4	0.7	0.3	4.5	0.6	0.9	0.4	0.7	0.3	7.1	0.4	4.4	0.6	0.9	0.3	7.1
40w	0.6	1.4	0.6	10.9	1.0	1.7	0.6	1.6	0.7	15.5	0.7	11.9	1.0	1.8	0.8	16.1
20d	0.6	0.4	0.5	2.4	0.5	0.5	0.6	0.4	0.3	3.0	0.5	2.5	0.5	0.5	0.3	3.1
30d	0.5	1.0	0.7	8.9	0.7	1.0	0.5	1.0	0.7	10.6	0.6	6.4	0.6	1.1	0.7	8.3
110	-7.5	12.1	-7.6	106.5	-6.5	15.7	-7.5	19.3	-7.3	151.8	-7.8	107.1	-6.8	27.6	-7.5	155.0
avg.	-0.8	2.7	-0.8	22.6	-0.5	3.4	-0.8	3.9	-0.8	31.8	-0.8	22.4	-0.5	5.4	-0.8	32.1

the service extension. Again, adding the speed extension to support the service extension or the capacity extension does not pay off, although the heuristic can solve these problems in all cases.

Also the extremely large instances '110' are solved fast by the heuristic. For models 'base' and 'capacity', runtimes are about a quarter of a minute which is as fast as the approach of Lübbecke et al. (2018). By considering the δ-gaps, it is found that the matheuristic produces much better solutions compared with what Cplex achieves within its runtime limit. The method can also handle all other extensions presented in this paper, where speed decisions again increase solution times. Adding the hard waiting time limits from the service extension rendered two of the '110' instances infeasible. Therefore the corresponding results reported in columns 'service', 'service+capacity', and 'speed+service+capacity' of Table 3 are averages over eight solutions only.

Eventually, the proposed matheuristic quickly provides high quality solutions for all model variants presented in this paper.

Next to problem size and density, a further property that impacts solving the problem are the sizes of the ships, i.e. their traffic group numbers g_i. To analyze this property, instances from set '40w' were solved repeatedly with traffic group numbers of all ships being fixed to $1, 2, \ldots 6$. Table 4 reports the δ-values and *cpu*-times averaged over the 10 instances in set '40w' under the varied traffic group numbers. If all ships belong to traffic group 1, there cannot be any conflicts as all ships go at same speed and opposing ships are small enough to pass each other in all segments of the canal. Therefore, the instances are solved instantaneously in the first iteration of the heuristic and the total transit times are clearly below the transit times observed for the actual ship sizes that were considered in the previous experiments ($\delta = -5.7\%$). Identical results are observed if all ships belong to traffic group 2 or 3 as these ships can still travel without any

Table 4: Results with identical traffic group numbers g_i of all ships for instances '40w'.

g_i	1	2	3	4	5	6
δ	-5.7	-5.7	-5.7	-0.3	10.6	41.4
cpu	0.1	0.1	0.1	1.9	3.4	4.7

conflict. This changes if all ships belong to traffic group 4, where conflicts arise in the narrow transit segments with passage number $p_s = 6$. If all ships have an identical traffic group number of 5, they suffer from substantial waiting time as even the wider transit segments ($p_s = 8$) are insufficient for a passing of opposing traffic. If all ships have a traffic group number of 6, they all have to go at the reduced default speed of $\overline{v}_i = 12$ km/h, which effects a further strong increase of the δ-gap. The *cpu*-times show that the problem becomes more difficult for traffic group numbers larger than 3. Note that the average traffic group number over all ships in the real world instances 'w40' is 3.7. For these ships, the $\delta = 0.6$ (see Table 3) is larger than the gap observed here for identically-sized ships of traffic group 4. This is because the real world instances contain a number of large ships of traffic groups 5 and 6 whose scheduling strongly affects the solution quality, see also subsequent experiment in Section 5.4.

5.4. Computational Experiment 3: Waiting Time Limits

Adding the service extension to the base model guarantees that the total transit time of ship $i \in \mathcal{V}$ for traversing the canal will be at most $\frac{\sum_{s \in \mathcal{E}} l_s}{\overline{v}_i} + W^{\max}_{g_i}$. In the previous experiments, the problems with default waiting time limits $W^{\max}_{g_i}$ could be solved without much difficulty. In this experiment, it is attempted to further reduce the limits $W^{\max}_{g_i}$, as this would allow the canal authority to guarantee even lower maximum transit times for ships using the canal. To analyze this potential, instance set '40w' is solved by the heuristic with model 'service' under various maximum total waiting times of $W^{\max}_g = 180$, 160, 140 and 120 minutes for traffic group numbers $g = 1$ to 5. For ships with traffic group number 6, the default value of $W^{\max}_6 = 120$ is kept, see Section 3.5. Note that the differing default values for $W^{\max}_g$ are the only discrimination among ships in the current traffic management process. Bringing the limits $W^{\max}_g$ closer together for all six traffic groups can therefore be considered a further step to a truly equal treatment of ships.

Figure 9 shows the maximum waiting times and average waiting times of ships per traffic group number g in the solutions for the base setting without any waiting time limits ($W^{\max}_{1-6} = \infty$), for the default service-setting ($W^{\max}_{1-5} = 180$), and in the more restricted service-settings ($W^{\max}_{1-5} =$ 160, 140, and 120). It is observed that the maximum waiting times are effectively restricted. The average waiting time becomes smaller for the largest ships ($g = 6$) but stays almost the same for all other traffic groups. The reason for this is that there are relatively few large ships ($g = 6$) in the instances, where cutting the maximum total waiting time of one ship effects a substantial reduction in the average waiting time for this group. However, although the service extension effectively limits the maximum total waiting time of ships, Figure 9 shows that waiting times are not distributed fairly among the ships. In all considered settings the average waiting times increase in the traffic group number although the decisions are made under an unweighted objective function. The explanation is that smaller ships can slip through the canal more or less

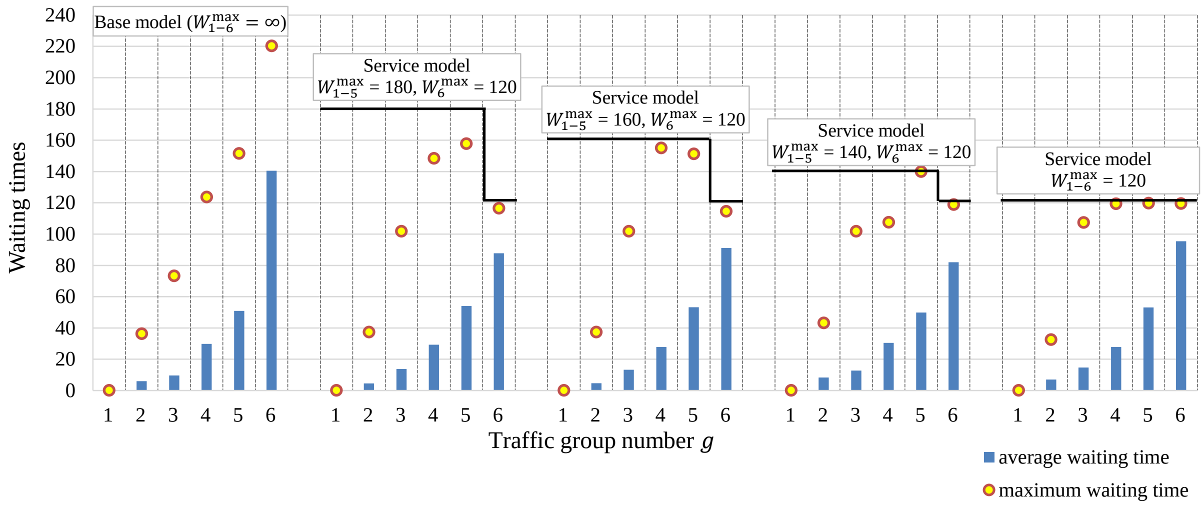

Figure 9: Average and maximum ship waiting time per traffic group under varied waiting limits for instances '40w'.

independently of the opposing traffic whereas larger ships get more likely in conflict with opposing traffic and, thus, have to wait at an increasing extent. This reveals an inherent disadvantage for larger ships that comes from their restricted passing capabilities in the transit segments. From these observations, the canal authority might be better off with a weighted objective function that gives priority to large ships. Although this looks like an unfair treatment of ships at first, the resulting plans could then have a more fair distribution of waiting times among ships of all sizes.

In Figure 9, values below $W_{1-5}^{\max} = 120$ were not considered as the service model can then not be solved feasibly for all instances. The issue can be tackled by also including speed decisions, which leads to model configuration 'speed+service'. This approach guarantees a less tight maximum transit time of $\frac{\sum_{s \in \mathcal{E}} l_s}{\underline{v}_i} + W_{g_i}^{\max}$ for ship i, but the problem can be solved feasibly more likely as the waiting time limitations can be met by slowing down ships purposefully. Alternatively, the heuristic might be allowed to relax Constraints (21) and (22) for a ship, if it can otherwise not provide a feasible solution for an instance. For such ships the transit times are no longer bounded, but the heuristic can then deliver feasible solutions in all cases. The quality of a solution is then measured in terms of the number of ships that do not meet the waiting limits in the final solution. This measure is called the service level SL. The corresponding extension is referred to as 'service (relax)'. Of course, the capacity extension can be added additionally.

Table 5 reports for the resulting settings the number of feasible solutions f found by the heuristic for instance set '40w', the δ-gap as defined before, and the service level SL for the relaxed service extension with wait limits $W_g^{\max}$ being set to 180, 160, ... 60 minutes. The results reveal that the speed extension leads to many more feasible solutions under very restricted waiting times and that adding siding capacity constraints is again handled without difficulty. Applying the service-relaxation produces feasible solutions also for very low values of $W_g^{\max}$ but some ships then have to accept longer waiting times ($SL > 0$). This measure can be drastically reduced by

Table 5: Performance of service-configurations under varied waiting limits for instances ’40w’.

	service		speed+ service		speed+ service+ capacity		service (relax)			speed+ service (relax)			speed+ service (relax)+ capacity		
W^{max}	f	δ	f	δ	f	δ	f	SL	δ	f	SL	δ	f	SL	δ
180	10	1.0	10	0.7	10	0.8	10	0.0	1.0	10	0.0	0.7	10	0.0	0.8
160	10	0.9	10	0.9	10	0.8	10	0.0	0.9	10	0.0	0.8	10	0.0	0.8
140	10	0.8	10	0.9	10	0.8	10	0.0	0.8	10	0.0	0.9	10	0.0	0.8
120	10	1.0	10	0.8	10	0.9	10	0.0	1.0	10	0.0	0.8	10	0.0	0.9
100	9	-	10	0.9	10	0.8	10	0.1	1.5	10	0.0	0.9	10	0.0	0.8
80	2	-	10	1.1	10	0.9	10	1.1	1.8	10	0.0	1.1	10	0.0	0.9
60	1	-	5	-	4	-	10	3.7	1.7	10	0.6	1.9	10	0.6	1.9
avg.	7.4	0.9	9.3	0.9	9.1	0.8	10.0	0.7	1.2	10.0	0.1	1.0	10.0	0.1	1.0

also adding the speed extension. Average runtimes for the matheuristic (not shown in Table 5) are like before with on average two seconds for the pure service extension and about 15 seconds for configurations that also include the speed extension. Also, there is no significant change in the δ-gaps for all settings. This means that low waiting times and, thus, low maximum transit times can be enforced without a relevant deterioration of the total transit times TTT. To summarize, applying solely the service extension proposed in this paper provides guaranteed maximum transit times to ships and can be solved quickly, but it might not lead to a feasible plan if waiting time limits are lowered too much. The traffic managers can then apply the speed extension and/or the service-relaxation to achieve feasible plans with an improved service quality.

6. Conclusions

This paper presented mathematical optimization models with several practical extensions for the traffic management problem of the Kiel Canal. The experiments show that the MIP solver Cplex has difficulties solving the problem, especially when one or more of the extensions are added. In contrast, the matheuristic proposed in this paper quickly obtains high quality solutions for all problem variants. Experiments show that the heuristic can respect the capacities of sidings without much computational effort. Furthermore, the heuristic can ensure additional service requirements by strictly limiting waiting times. This means that maximum transit times for ships can be guaranteed. By including speed decisions, ships can be purposefully slowed down which produces feasible plans under very strict waiting time limits. A service-relaxation can be applied for some of the ships if feasibility cannot be established otherwise. Since the heuristic is so fast, the traffic managers might gradually reduce waiting limits to find feasible solutions of best service-quality. An issue that will be tackled in future research is to consider the robustness of the plans to cope with uncertainties in ship travel times and other disturbances. Another topic is to apply the proposed heuristic in a rolling horizon framework as was briefly outlined in Section 4. An adoption of the ideas presented in this paper to other inland waterways is a further subject of future research.

Acknowledgments: *We thank the Waterways and Shipping Board Brunsbüttel for supporting this project. We thank in particular Dipl.-Nautiker Sven Krieg for providing many helpful inputs. Furthermore, we thank the two anonymous referees, whose valuable comments helped us improving the paper.*

Bibliography

Andersson, H., Fagerholt, K., Hobbesland, K., 2015. Integrated maritime fleet deployment and speed optimization: Case study from RoRo shipping. Computers & Operations Research 55, 233 – 240.

Campbell, J. F., Smith, L. D., Sweeney, D. C. I., Mundy, R., Nauss, R. M., Jan 2007. Decision tools for reducing congestion at locks on the upper mississippi river. In: Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07). pp. 56–56.

Castillo, E., Gallego, I., Ureña, J. M., Coronado, J. M., 2011. Timetabling optimization of a mixed double- and single-tracked railway network. Applied Mathematical Modelling 35 (2), 859 – 878.

Codato, G., Fischetti, M., 2006. Combinatorial benders' cuts for mixed-integer linear programming. Operations Research 54 (4), 756–766.

Corry, P., Bierwirth, C., 2018. The berth allocation problem with channel restrictions. Transportation Science, in press.

Gafarov, E. R., Dolgui, A., Lazarev, A. A., 2015. Two-station single-track railway scheduling problem with trains of equal speed. Computers & Industrial Engineering 85, 260 – 267.

Griffiths, J. D., 1995. Queuing at the Suez Canal. Journal of the Operational Research Society 46, 1299 – 1309.

Gu, Z., Nemhauser, G. L., Savelsbergh, M. W., Mar 2000. Sequence independent lifting in mixed integer programming. Journal of Combinatorial Optimization 4 (1), 109–129.

Heitmann, N., Rehdanz, K., Schmidt, U., 2013. Determining optimal transit charges: the Kiel Canal in Germany. Journal of Transport Geography 26, 29 – 42.

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

Jia, S., Li, C.-L., Xu, Z., 2018. Managing navigation channel traffic and anchorage area utilization of a container port. Transportation Science, in press.

Kraay, D. R., Harker, P. T., 1995. Real-time scheduling of freight railroads. Transportation Research Part B: Methodological 29 (3), 213 – 229.

Kraay, D. R., Harker, P. T., Chen, B., 1991. Optimal pacing of trains in freight railroads: Model formulation and solution. Operations Research 39 (1), 82 – 99.

Lalla-Ruiz, E., Shi, X., Voß, S., 2018. The waterway ship scheduling problem. Transportation Research Part D: Transport and Environment 60, 191 – 209.

Lamorgese, L., Mannino, C., 2015. An exact decomposition approach for the real-time train dispatching problem. Operations Research 63 (1), 48–64.

Lübbecke, E., 2015. On- and Offline Scheduling of Bidirectional Traffic. Logos Publisher, Berlin, PhD Thesis, Technical University Berlin.

Lübbecke, E., Lübbecke, M. E., Möhring, R. H., 2018. Ship Traffic Optimization for the Kiel Canal. Operations Research, in press.

Lusby, R. M., Larsen, J., Ehrgott, M., Ryan, D., 2011. Railway track allocation: models and methods. OR Spectrum 33 (4), 843 – 883.

Luy, M., 2011. Algorithmen zum Scheduling von Schleusenvorgängen: Verkehrsoptimierung am Beispiel des Nord-Ostsee-Kanals (in German). Diplomica Verlag, Hamburg, Master's Thesis at Technical University Berlin.

Passchyn, W., Coene, S., Briskorn, D., Hurink, J. L., Spieksma, F. C., Berghe, G. V., 2016. The lockmaster's problem. European Journal of Operational Research 251 (2), 432 – 441.

Righini, G., 2016. A network flow model of the Northern Italy waterway system. EURO Journal on Transportation and Logistics 5 (2), 99–122.

Sluiman, F., 2017. Transit vessel scheduling. Naval Research Logistics 64 (3), 225 – 248.

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

Szpigel, B., 1973. Optimal train scheduling on a single line railway. In: Ross, M. (Ed.), Operational Research. No. 72. pp. 343 – 352.

Tan, Z., Wang, Y., Meng, Q., Liu, Z., 2018. Joint ship schedule design and sailing speed optimization for a single inland shipping service with uncertain dam transit time. Transportation Science, in press.

Ulusçu, Ö. S., Özbas, B., Altiok, T., Or, I., Yilmaz, T., 2009. Transit vessel scheduling in the Strait of Istanbul. Journal of Navigation 62 (1), 59 – 77.

Verstichel, J., Kinable, J., Causmaecker, P. D., Berghe, G. V., 2015. A combinatorial benders' decomposition for the lock scheduling problem. Computers & Operations Research 54, 117 – 128.

WSV, 2018. Wasserstrassen- und Schifffahrtsverwaltung des Bundes (WSV), Der Nord-Ostsee-Kanal. `http://www.wsa-kiel.wsv.de/Nord-Ostsee-Kanal/`, accessed 03/19/2018.

Zhang, J., Santos, T. A., Soares, C. G., Yan, X., 2017. Sequential ship traffic scheduling model for restricted two-way waterway transportation. Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment 231 (1), 86–97.

Zhang, X., Lin, J., Guo, Z., Liu, T., 2016. Vessel transportation scheduling optimization based on channel-berth coordination. Ocean Engineering 112, 145 – 152.

Zhen, L., Liang, Z., Zhuge, D., Lee, L. H., Chew, E. P., 2017. Daily berth planning in a tidal port with channel flow control. Transportation Research Part B: Methodological 106, 193 – 217.

Zhou, X., 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.