Maritime Transport Research 2 (2021) 100008

Contents lists available at ScienceDirect

Maritime Transport Research

journal homepage: www.elsevier.com/locate/martra

Scheduling ships with uncertain arrival times through the Kiel Canal

Tina Andersen[a], Joakim Høgset Hove[a], Kjetil Fagerholt[a], Frank Meisel[b,*]

[a] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway
[b] School of Economics and Business, Kiel University, Germany

ARTICLE INFO

Keywords:
Ship scheduling
Matheuristic
Simulation
Uncertainty
Time corridors

ABSTRACT

The Kiel Canal is a two-way waterway that connects the Baltic Sea and the North Sea. The canal consists of an alternating sequence of narrow transit segments and wider siding segments. This calls for solving a ship scheduling problem to decide which ships have to wait in sidings to let opposing traffic pass through such that the total traversing time of all ships is minimized. This paper extends previous studies on scheduling ships through the Kiel Canal by considering that the arrival times of the ships at the entrance to the canal are subject to uncertainty. This is a major challenge in the planning as it gives frequent need of replanning to make the schedules feasible. We propose a mathematical formulation for the problem to mitigate the negative effects of the uncertainty. This formulation incorporates time-corridors, so that the schedule will still be valid as long as the ships arrive within their given time-corridors. To solve real-sized instances of the problem, we adapt a matheuristic that adds violated constraints iteratively to the problem. The matheuristic was tested within a rolling horizon simulation framework to study the effect of arrival time uncertainty. We show by experiment that solutions of the matheuristic for different time-corridor widths can be used to identify a suitable corridor width that trades off the average traversing time of ships and the number of reschedules required in the planning. A simple myopic heuristic, reflecting the current scheduling practice, was used to generate benchmark results, and tests on real data showed that the matheuristic provides solutions with significantly less need of replanning, while at the same time keeping the total traversing times for the ships short. We also provide simulations to gain insight about the effect on the ships' average traversing time from upgrading the narrow transit segments.

1. Introduction

The seaborne cargo volumes are constantly growing along with the concern about the environmental consequences of greenhouse gas emissions and air pollution (UNCTAD, 2018). Shipping companies therefore look for ways to reduce their fuel costs and emissions. The Kiel Canal cuts through the Northern part of Germany between the North Sea and Baltic Sea (blue route in Fig. 1), allowing ships to save on average 250 nautical miles, and reduce the fuel consumption accordingly, compared to sailing around the Jutland Peninsula in Denmark (red route), see Heitmann et al. (2013). The Kiel Canal has an annual traffic of about 30 000 ships (UCA, 2019) making it the most trafficked canal in the world. The canal is 98.7 kilometers long and has two sets of locks, one at each end, in the portal towns Kiel and Brunsbüttel. The shipping companies also value the proximity to the port of Hamburg, which is the third largest port in Europe (UNCTAD, 2018) and can easily be accessed from the Kiel Canal's exit in Brunsbüttel.

* Corresponding author.
E-mail addresses: kjetil.fagerholt@ntnu.no (K. Fagerholt), meisel@bwl.uni-kiel.de (F. Meisel).

https://doi.org/10.1016/j.martra.2020.100008
Received 5 September 2020; Received in revised form 7 December 2020; Accepted 21 December 2020

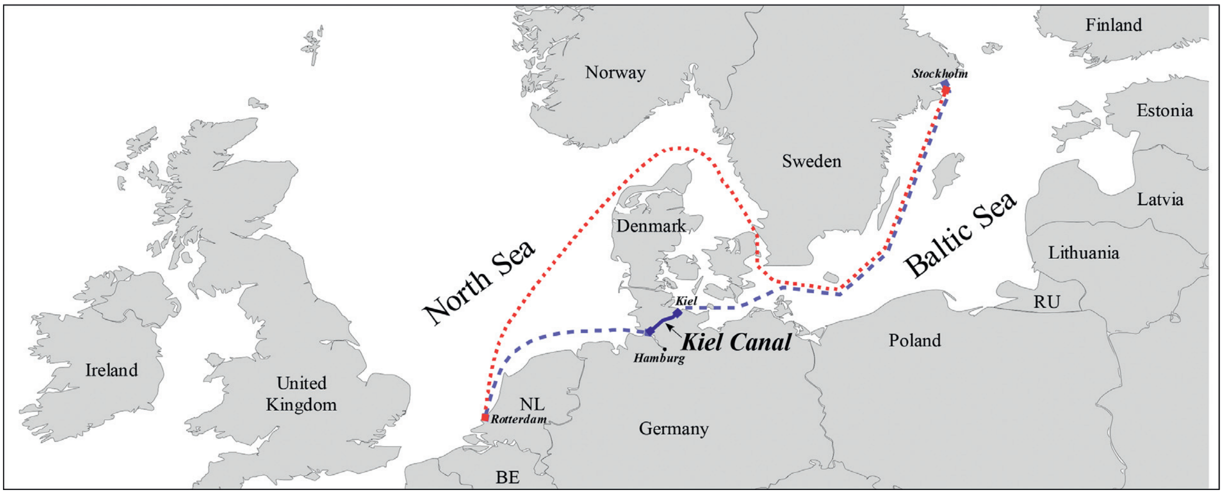


Fig. 1. Sailing route around the Jutland Peninsula and Kiel Canal shortcut. Meisel and Fagerholt (2019).

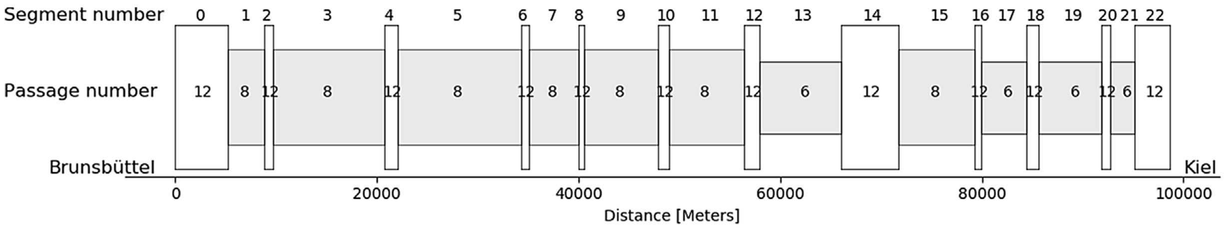


Fig. 2. Overview over sidings (white) and transit segments (grey) in the Kiel Canal together with passage number and segment number.

The daily operations of the Kiel Canal consist of planning safe ship schedules for ships traversing the canal. The Kiel Canal is made up of 23 different segments. It alternates between wide siding segments and narrow transit segments. Fig. 2 illustrates the sidings and transit segments over the length of the canal. Since the transit segments are narrow, large ships can only pass each other in the wider sidings. More precisely, for managing the traffic of the Kiel Canal, each segment is assigned a *passage number* that expresses its width and each ship is assigned a *traffic group number* that expresses its size. The passage number of sidings is 12. The passage number of very narrow transit segments is 6 and for somewhat wider transit segments it is 8, see Fig. 2. The traffic group numbers of ships range from 1 (small) to 6 (large). The rule for managing the traffic is that opposing ships cannot meet in a transit segment if the sum of their traffic group numbers exceeds the passage number of the segment. In such a case, one of the ships has to wait in a siding to let the opposing ship pass through.

Furthermore, the ships' travel speeds are given by speed limits varying with the size of the ship. The largest ships must travel at lower speeds than smaller ships. This, in turn, may lead to smaller ships queuing up behind larger ships in transit segments as ships can only overtake other ships in sidings due to safety reasons.

Scheduling ships through the Kiel Canal has previously been studied by Lübbecke et al. (2019) and Meisel and Fagerholt (2019), which propose different mathematical formulations and heuristic methods that provide solutions within an acceptable time. Ships that are going through the canal announce their arrival a few hours before their Expected Time of Arrival (*ETA*). All previous studies considered this information to be given with certainty and solved the ship scheduling as a deterministic problem. In contrast to the these previous studies, we consider that the ships' actual arrival times are subject to uncertainty and, especially, that ships are often delayed. This is a major practical challenge in the planning as it gives frequent need of replanning to keep the schedules feasible. Frequent rescheduling is time consuming and forces the planners to use simple rules of thumb, which may yield travel plans that are far from optimal with the consequence of longer traversal times through the canal for the ships. We have named the resulting scheduling problem *Ship Scheduling on the Kiel Canal with Uncertain Arrival Times* (*SSKC-UAT*).

Our contributions through this paper are as follows. We introduce the *SSKC-UAT* and propose a mathematical formulation for the problem to mitigate the negative effects of the uncertain arrival times. This formulation incorporates time-corridors, so that a schedule will still be valid as long as the ships arrive within their given time-corridors. In other words, the time-corridor serves as a buffer that hedges against delayed arrival times of ships at the canal. If possible, this corridor is preserved throughout a ship's canal journey such that a ship who arrives early within its time-corridor benefits in terms of an early canal exiting time. However, planned waiting times of a ship in a siding consume the corridor as such waiting obviously delays the earliest possible canal exiting time of a ship. The presented model takes care of these interdependencies. To ensure fast solutions to real-sized instances of the problem, we adapt the well-working matheuristic of Meisel and Fagerholt (2019) to our problem. The matheuristic has been tested within a rolling horizon simulation framework to study the effect of arrival time uncertainty and to identify suitable widths of the time-corridors by experiment. A simple myopic heuristic, reflecting the current scheduling practice, is used to generate benchmark results. Tests on realistic data show that the matheuristic provides solutions with significantly less need of replanning, while at the same time keeping the total traversing times for the ships at an acceptable level. We also provide simulations to gain insight about the effect on the ships' average traversing time from upgrading the narrow transit segments. It should be noted that even though this study is for scheduling ships through the Kiel Canal, the proposed model and matheuristic can be adapted for other waterways or for traffic systems that face similar problems.

The outline of this paper is as follows. Section 2 reviews the relevant literature, while Section 3 gives a detailed description of the *SSKC-UAT* together with a mathematical formulation for the problem. The matheuristics is described in Section 4, while Section 5 presents a simulation framework for evaluating schedules. The computational study is given in Section 6. Concluding remarks are drawn in Section 7.

2. Literature review

A number of studies has investigated traffic management problems for waterways all around the world. In particular, Griffiths (1995) present queuing models for building convoys of ships that go through the Suez Canal, Ulusçu et al. (2009) give an optimization model for sequencing ships that pass the Strait of Istanbul, Sluiman (2017) presents methods for sequencing ships at the Strait of Istanbul too as well as at the Sunda Strait, and Lalla-Ruiz et al. (2018) present an optimization model and a heuristic for scheduling ships at the Yangtze river delta. Further studies address ship scheduling for inland waterways without referring to a particular real-world case, see e.g. Alfandari et al. (2019). All those studies that address a particular real-world waterway take into account the specific layout of the considered waterway. Anyhow, what all these waterways have in common is that a ship has to go through only one or at most two narrow transit segments. Therefore, the decision making is typically to find a sequence in which the ships arriving at either side of such a segment are allowed to travel through this bottleneck. In contrast, the Kiel Canal consists of 23 alternating narrow and wide segments where each ship going through the canal has to pass all of these segments. For this reason, the ship scheduling decisions of the different segments are interdependent and require advanced methods for traffic management.

The particularities of traffic management for the Kiel Canal have been investigated in a number of recent studies. Lübbecke (2015) and Lübbecke et al. (2019) present a fundamental mixed-integer program that describes the combinatorics of the ship scheduling decisions over the various segments of the canal. This model is based on the assumption that ships travel at their maximum allowed speed at all time. This implies that the time it takes for a ship to traverse a segment is a given constant and that having ships wait in sidings is the only way to avoid conflicts between ships. The authors propose a labeling algorithm which creates a feasible solution by iteratively adding the ships in a given order, based on the estimated time of arrival, with the use of what the authors refer to as collision-free routing. A local search is then applied to search for alternative ship orders that yield improved schedules. We want to note that the labeling algorithm in these papers determines time windows within which a ship can enter a segment without causing conflicts with other ships. These time windows are similar to the time-corridors determined in our paper. However, the time windows in Lübbecke (2015) and Lübbecke et al. (2019) are merely a means for feasibly inserting a ship into a partial schedule within a deterministic problem setting. The proactive dimensioning of time-corridors as is proposed in our paper for deriving robust solutions under uncertain ship arrival times is not considered there. Meisel and Fagerholt (2019) suggest extensions for the mixed-integer program and an alternative heuristic solution method. The extensions enrich the model (1) by considering the speed of ships as a decision rather than a fixed parameter, (2) by restricting the waiting times of ships as an instrument for improved service quality, and (3) by including capacity constraints that ensure that ships waiting in a siding segment do not exceed the available space of this segment. The proposed solution method is a matheuristic, which first relaxes all model constraints that are responsible for avoiding conflicts among ships and, then, iteratively adds violated constraints to obtain a feasible solution. It is shown by experiment that the proposed matheuristic solves realistically sized instances within seconds even for the extended versions of the problem. Furthermore, the study of Luy (2011) investigates the operations planning of the locks that are located at both ends of the Kiel Canal. While lock operations and traffic management are interdependent and, thus, might ideally be considered as an integrated problem, the two sub-problems are in the responsibility of two distinct canal authorities, which is why they are treated individually in the different streams of research.

Traffic management problems similar to the scheduling of ships in narrow waterways are found in rail scheduling, in particular if trains traveling in the same or in opposite direction cannot meet (or overtake) in segments that have a single track only, see the surveys of Cordeau et al. (1998) and Lusby et al. (2011). Train scheduling problems have been treated in a deterministic setting in many studies like, for example, Castillo et al. (2011), Gafarov et al. (2015), Yang et al. (2016), Lamorgese et al. (2017) and Zhang et al. (2019). Anyhow, a large number of studies on (single track) rail scheduling has also addressed this problem from a robustness perspective. An overview of modeling approaches to robust train timetabling is provided by Cacchiani and Toth (2012). A recent survey of the literature in this field is given by Lusby et al. (2018). The authors distinguish different types of problems where 'timetabling' is the field that comes close to our research. There are many approaches in this field that consider train-related issues like uncertain dwell times in stations, travel delays from passenger perspectives, etc, which are not relevant here. Papers that come close to the traffic management of ships in a canal are, for example, the single track studies of Meng and Zhou (2011), Shafia et al. (2012), and Jovanović et al. (2017). Meng and Zhou (2011) determine a robust meet-pass plan for trains at the example of a 138 km single track route that connects 18 stations in China. In this problem, the running time of trains along the segments is uncertain and segments can have a capacity breakdown. The uncertainty is captured in a set of scenarios and the goal is to find a solution that minimizes expected schedule deviations with respect to an initial planning table that prescribes entering and leaving times at segments and minimum dwell times at stations. Shafia et al. (2012) consider a single track corridor that connects the cities of Tehran and Isfahan in Iran. They focus on determining a robust, periodic timetable that can be applied repetitively and absorb delays. The problem is modelled as a variant of the job shop scheduling problem that also captures features like station capacity and headway time constraints. Jovanović et al. (2017) consider a busy rail corridor in Sweden. They present a knapsack formulation that purposefully distributes an amount of buffer time in a given timetable such as to improve the reliability and on-time performance of the trains. To this end, the trains may be given different priorities to reflect their need for protection against uncertain events.

Eventually, these problems share similarities with the traffic management in artificial waterways, but they also differ with regard to central features. For example, in single track rail systems opposing trains can never meet on a track whereas the Kiel Canal allows that opposing ships can meet in a transit segment under certain conditions that must be carefully reflected in the corresponding optimization models and solution methods. Also, trains must stop at all stations according to their timetable whereas in the ship traffic management it needs to be decided purposefully which ship stops where and for how long to let opposing traffic pass by. With regard to the above papers on robust train scheduling, we have to state that these approaches differ substantially in terms of their scope, the nature of the uncertain parameters, the decisions made, and/or the pursued objective. More precisely, Meng and Zhou (2011) focus on schedule recovery after a major service disruption that comes along with a capacity breakdown of a rail track segment, whereas breakdowns of segments are not a source of disruption for the daily traffic management of the Kiel Canal. Shafia et al. (2012) generate periodic schedules that can be executed repetitively as is appropriate for timetabled train operations but irrelevant for waterways where ship traffic does not follow repetitive patterns. Furthermore, they focus on issues that are of particular relevance in train operations such as an explicit separation of train departures from a same station to avoid crowded platforms. Such separations are not needed and even undesirable when scheduling ships in a canal. Jovanović et al. (2017) face a tactical problem of inserting buffer times into a given train timetable such that the overall cycle time of the timetable does not change if train operations are delayed. In contrast, traffic management at the Kiel Canal is an operational problem that is about optimizing ship schedules (timetables) w.r.t. time-corridors (buffer times) where the uncertainty lies in the initial arrival time of the ships and the objective is to minimize the total canal exiting time of all ships instead of an overall cycle time. Due to these differences, robust traffic management for the Kiel Canal clearly requires specific models and methods on its own.

Of course, there are also studies on robustness issues in ship operations management. However, such papers typically focus on routing decisions for the considered ships (e.g. Norlund et al., 2015) but not on managing ship traffic within a particular waterway infrastructure. For this reason and because all ship traffic management papers mentioned above focus on deterministic problems too, our study's contribution is to present a first approach to robust traffic management for inland waterways like the Kiel Canal. In this sense, our paper belongs to the stream of traffic management research.

3. Problem definition and mathematical model

General aspects of scheduling on the Kiel Canal are described in Section 3.1, while the Ship Scheduling problem on the Kiel Canal with Uncertain Arrival Times (*SSKC-UAT*) is presented in Section 3.2. Modeling assumptions and notation are given in Section 3.3, before we present the mathematical model for the *SSKC-UAT* in Section 3.4.

3.1. General aspects of scheduling ships through the Kiel Canal

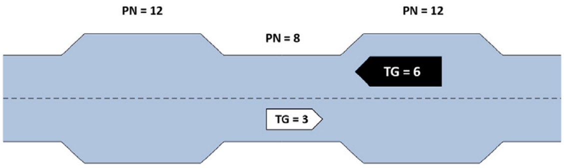
The Kiel Canal has bidirectional traffic. If a ship enters the canal on the Kiel-side it is *westbound* and if a ship enters in Brunsbüttel it is *eastbound*. Two ships that travel in the opposite direction are *opposing*, while ships traveling in the same direction are *aligned*.

The ships are categorized according to their size, and based on this they are given a *traffic group number* ranging from 1 to 6, where traffic group 6 consists of the largest ships. The maximum allowed traveling speed inside the canal depends on the traffic group number, and is 15 km/h for traffic group numbers 1–5 and 12 km/h for traffic group number 6. We assume that all ships follow this speed limit within the canal, except for when they have to wait in the sidings. The *traversing time* through a given segment is the time it takes for the ship to pass through that segment.

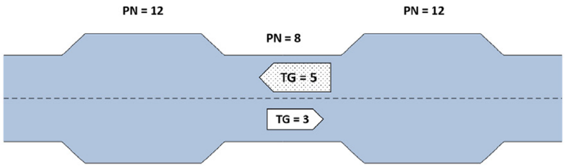
A feasible schedule needs to keep certain safety restrictions. Situations that might violate these safety restrictions are called conflicts. In the Kiel Canal, there are two types of conflicts. The first type is *aligning conflicts*. Each pair of aligned ships may pass each other in sidings, while it is not allowed for any aligned ships to overtake each other in transits. Therefore, there is a possible conflict between two aligned ships in every transit. In order to avoid such conflicts, the ships need to keep a minimum safety distance through the transit.

The other type of conflict is *opposing conflicts*. Two opposing ships may meet each other in a segment only if the sum of their traffic group number is less than or equal to the passage number of the segment. Therefore, ships will always be able to meet in sidings as these have a passage number of 12, see Fig. 2. For transit segments on the other hand, too large ships will not always be able to pass each other and a conflict arises, as demonstrated in Fig. 3. To avoid conflicts, one of the ships might have to wait in a siding so that the other ship has time to complete the transit segment before the other ship enters.

To ensure a safe passage of the ships and comply with the safety regulations, it is necessary to calculate a certain *safety time* that must be kept between the entering of two ships at a transit segment. Such a safety time must be calculated both for each pair of opposing and aligned ships. Details on how to calculate safety times and on how to derive set of ship pairs C_s^A and C_s^O that could have an aligning or opposing conflict in a segment s of the canal are given in Meisel and Fagerholt (2019).



(a) The dark ship must wait since the sum of traffic group (TG) numbers exceeds the passage number (PN) of the transit segment $(3 + 6 > 8)$.



(b) Both ships can meet in the transit segment since the sum of their traffic group numbers does not exceed the passage number $(3 + 5 \leq 8)$.

Fig. 3. Demonstration of legal ship traffic.

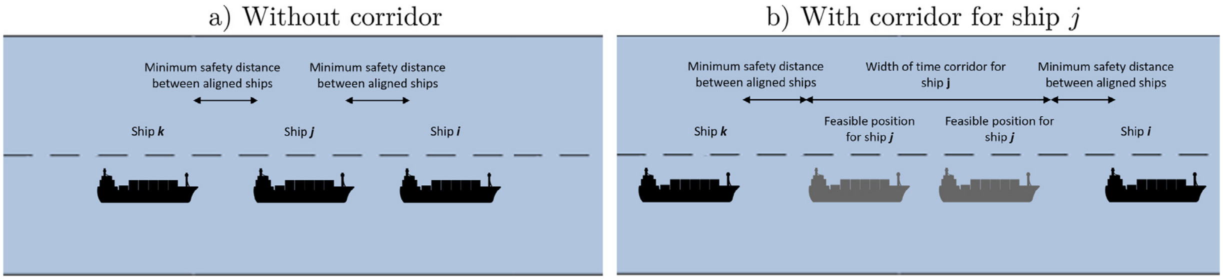


Fig. 4. a) The only allowed position for ship j is the position the ship is currently placed in, since there is no corridor in the schedule. If ship j is delayed, the schedule will become infeasible as the following ship k has to slow down or overtake ship j. b) Ship j has a corridor and is now allowed to be in all positions inside this corridor. The two extreme points, the earliest and latest allowed position, are marked with the two grey ships. The increased flexibility in the schedule comes at the cost of forcing ship k to travel further behind ship i.

3.2. Ship scheduling in the Kiel Canal with uncertain arrival times

In order to facilitate the traffic management at the Kiel Canal, ship captains are asked to announce their *Expected Time of Arrival* (*ETA*) at the canal a few hours in advance. Clearly, the ETA is subject to a certain degree of uncertainty and ships may arrive later than expected within the canal, for example because of lower actual traveling speed due to weather conditions, heavy ship traffic or queuing time before entering the locks.

Such delays might render planned schedules infeasible and force the traffic operators to reschedule to obtain new conflict-free travel schedules. Frequent rescheduling is time consuming for the operators and might force them to use either fast heuristic approaches or simple rules of thumb. This may yield travel plans that are far from optimal with the consequence of longer traversal times through the canal.

In the *SSKC-UAT*, we consider this uncertainty in arrival times and try to reduce the amount of rescheduling while at the same time keeping the total traversing time for all ships through the canal as low as possible. The problem then consists of generating a feasible travel schedule for all ships that to some extent hedges against arrival time delays, while minimizing the total traversing time for all ships. Our approach to achieve this is by introducing a *time-corridor* for each ship (i.e., a time buffer), which means that space is temporarily reserved for that ship in the canal segments. As long as the ship enters the canal within its time-corridor and stays within this corridor throughout its journey, the scheduled travel plan will remain feasible. If for example, it was planned that a ship enters the canal at an ETA of 10:30 and is given a time-corridor of 20 minutes, the schedule will be feasible and valid as long as the ship enters the canal in the interval [10:30, 10:50].

The use of time-corridors is demonstrated in Figs. 4 and 5. In Fig. 4a, ship j has no corridor and is forced to remain at its relative position between ships i and k without any flexibility. In Fig. 4b, ship j received a distance corridor which gives more flexibility of the actual ship position in-between ships i and k. This distance corridor corresponds directly to a time-corridor. A better illustration of time-corridors is obtained from representing the ship traveling in a time-space diagram as is done in Fig. 5. Fig. 5a represents the situation without time-corridors where all three ships move at a same speed (that corresponds to the slope of the arcs) through the depicted canal segments. The separation of the lines represents the required safety distances between the ships. Fig. 5b represents the situation where ship j receives a time-corridor. Here, the actual position of the ship is no longer relevant but the schedule remains feasible as long as the ship stays within its corridor. In other words, the corridor can compensate for uncertain entering times of the ship. The figure also shows that ship k needs to be postponed in order to provide a corridor for ship j.

3.3. Modeling assumptions and notation

When modeling this problem, some assumptions have been made. Firstly, it is assumed that each ship traverses the canal at a constant speed (except for when waiting in the sidings) equal to the maximum speed based on the ship's traffic group number.

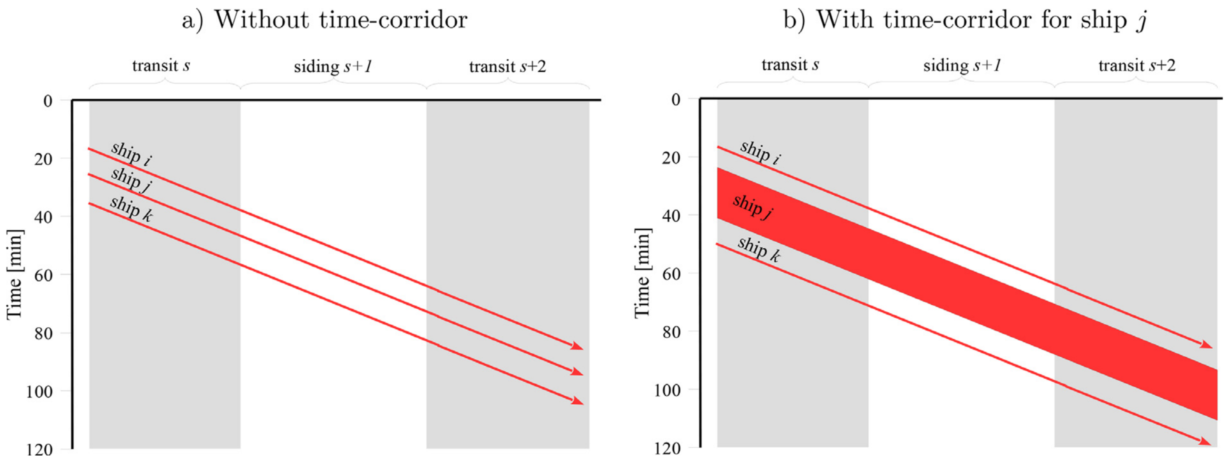


Fig. 5. a) The time-space representation shows that ship j has to keep its relative position in-between ships i and k. If ship j is delayed, the schedule will become infeasible as the following ship k has to slow down or overtake ship j. b) Ship j has a time-corridor and is now allowed to enter the segment at any time within the corridor without making the schedule infeasible. The increased flexibility in the schedule comes at the cost of forcing ship k to travel further behind ship i than without time-corridors.

Furthermore, extra traversing time due to acceleration or deceleration is neglected which means that travel times per segment are deterministic and can be calculated as part of the pre-processing. Secondly, capacity limits in the sidings are not considered here as experiments in Meisel and Fagerholt (2019) showed that these limits are hardly binding in any solution to the traffic management problem. Lastly, we assume that all ships travel through the entire canal.

The notation that we use in the mathematical formulation for the *SSKC-UAT* is presented and explained in the following.

Sets

$\mathcal{S}$: Set of sidings

$\mathcal{T}$: Set of transit segments

$\mathcal{E} = \mathcal{S} \cup \mathcal{T}$: Set of all segments, both sidings $\mathcal{S}$ and transit segments $\mathcal{T}$

$\mathcal{V}$: Set of ships

$\mathcal{V}^E$: Set of eastbound ships, i.e. ships that enter the canal in westmost segment 0 (Brunsbüttel) and exit through the eastmost segment $\bar{s} = 22$ (Kiel), see Fig. 2

$\mathcal{V}^W$: Set of westbound ships, i.e. ships that enter the canal in eastmost segment $\bar{s} = 22$ and exit through the westmost segment 0

C_s^A : Set of all possible aligning conflicts on segment s, given as a set of pairs of ships

C_s^O : Set of all possible opposing conflicts on segment s, given as a set of pairs of ships

Parameters

TC_i : Initial time-corridor width given to ship i

ETA_i : Estimated time of arrival at first canal segment for ship i

$\Delta_{i,j,s}$: Safety time between ship i and j in segment s

$D_{i,s}$: Traversing time for ship i in segment s

M : Big M-parameter

Decision Variables

$z_{i,j,s}$: Binary variable that takes value 1 if ship i enters segment s before ship j, 0 otherwise

$w_{i,s}$: The waiting time for ship i in siding segment s

$\underline{t}_{i,s}$: Planned entering time for ship i into segment s

$\bar{t}_{i,s}$: Latest entering time for ship i into segment s

3.4. Mathematical model

The model aims at making a schedule that can to some extent withstand delays in arrival times by introducing time-corridors. We take up and extend the base model formulation of Meisel and Fagerholt (2019), which assumed a deterministic setting without a need to mitigate against uncertainty in the arrival times. Model extensions that were investigated in Meisel and Fagerholt (2019) like, for example, considering ship speed as a decision variable, are not included here for reasons of brevity. The time-corridor of a ship is defined to be the difference between a planned entering time and a latest entering time at each segment and is used to retain space temporarily in the canal, creating a free path for the ship as long as it stays within the given time-corridors. Initially, at the begin of the canal journey, the *ETA* defines the earliest time in the corridor. The latest allowed entering time follows from a preset corridor width TC_i that is added to the earliest time.

The model uses the following decision variables: The binary variable $z_{i,j,s}$ decides which of two conflicting ships $(i,j) \in C_s^A \cup C_s^O$ gets to enter segment s first, where $z_{i,j,s} = 1$ if ship i enters before ship j into segment s and $z_{i,j,s} = 0$ if ship j enters before ship i into segment s. Variable $w_{i,s}$ denotes the waiting time for ship i in siding segment s. The planned entering time for ship i into segment s is denoted by $t_{i,s}$ whereas $\bar{t}_{i,s}$ denotes the latest entering time for ship i into this segment. $t_{i,s}$ and $\bar{t}_{i,s}$ span the time-corridor for ship i in segment s, meaning that the schedule remains feasible as long as the ship enters the segment within this time span. The optimization model is then formulated as follows.

$$\min TTT = \sum_{i \in \mathcal{V}^E} (t_{i,\bar{s}} + D_{i,\bar{s}} - t_{i,0}) + \sum_{i \in \mathcal{V}^W} (t_{i,0} + D_{i,0} - t_{i,\bar{s}}) \tag{1}$$

The objective function (1) minimizes the total transit time (TTT) for the ships that traverse the canal. The exiting time for a ship is when the ship leaves the last segment, which is $t_{i,\bar{s}} + D_{i,\bar{s}}$ for eastbound ships, while it is $t_{i,0} + D_{i,0}$ for westbound ships.

$$t_{i,s} + D_{i,s} = t_{i,s+1} \qquad i \in \mathcal{V}^E, s \in \mathcal{T} \tag{2}$$

$$t_{i,s} + D_{i,s} = t_{i,s-1} \qquad i \in \mathcal{V}^W, s \in \mathcal{T} \tag{3}$$

$$t_{i,s} + D_{i,s} + w_{i,s} = t_{i,s+1} \qquad i \in \mathcal{V}^E, s \in \mathcal{S} \backslash \{\bar{s}\} \tag{4}$$

$$t_{i,s} + D_{i,s} + w_{i,s} = t_{i,s-1} \qquad i \in \mathcal{V}^W, s \in \mathcal{S} \backslash \{0\} \tag{5}$$

Constraints (2) - (5) handle the flow of the ships. Constraints (2) set the planned entering time at the subsequent segment for each eastbound ship and for each transit segment. Constraints (3) do the same for westbound ships. Constraints (4) and Constraints (5) handle the flow for the sidings for eastbound and westbound ships, respectively. Note that waiting is allowed only in sidings.

$$t_{i,0} = ETA_i \qquad i \in \mathcal{V}^E \tag{6}$$

$$t_{i,\bar{s}} = ETA_i \qquad i \in \mathcal{V}^W \tag{7}$$

$$\bar{t}_{i,0} = ETA_i + TC_i \qquad i \in \mathcal{V}^E \tag{8}$$

$$\bar{t}_{i,\bar{s}} = ETA_i + TC_i \qquad i \in \mathcal{V}^W \tag{9}$$

$$\bar{t}_{i,s} - t_{i,s} \geq \bar{t}_{i,s+1} - t_{i,s+1} \qquad i \in \mathcal{V}^E, s \in \mathcal{E} \backslash \{\bar{s}\} \tag{10}$$

$$\bar{t}_{i,s} - t_{i,s} \geq \bar{t}_{i,s-1} - t_{i,s-1} \qquad i \in \mathcal{V}^W, s \in \mathcal{E} \backslash \{0\} \tag{11}$$

Constraints (6) and (7) set the planned canal entering times, for eastbound and westbound ships, respectively, equal to a ship's estimated time of arrival. For eastbound ships this will be the entering time at segment 0 while for westbound it will be at segment $\bar{s}$. The latest entering time is set to be the estimated arrival time ETA_i plus the time-corridor TC_i given to the ship, as stated in Constraints (8) and (9). These constraints establish the initial width of each ship's time-corridor. As a ship travels through the canal, its given time-corridor may be consumed if it has to wait for another ship. This is because the time-corridor is a buffer that catches up variations of ship arrival times when entering the canal. If a ship actually arrives at its ETA_i, it can follow the time-corridor at the earliest segment entering times $t_{i,s}$ whereas if it just arrives at the time $ETA_i + TC_i$ it can follow the time-corridor at the latest entering times $\bar{t}_{i,s}$. However, if a ship enters the canal already at ETA_i but then has a planned waiting at some segment within the canal, the earliest entering times $t_{i,s}$ at subsequent segments are obviously postponed by the waiting time. This effects that the corridor shrinks according to the planned waiting time. Then, if the waiting at some segment completely consumed the time corridor of a ship, this ship has to follow a path without a time-corridor (as shown in Fig. 5a) for the rest of the canal journey. A further explanation of time-corridor consumption is provided below at the example of Fig. 6. Eventually, the time-corridor at a subsequent segment is at most as large as the time-corridor at the current segment, see Constraints (10) and (11).

$$t_{i,s} \leq \bar{t}_{i,s} \qquad i \in \mathcal{V}, s \in \mathcal{E} \tag{12}$$

$$\bar{t}_{i,s} + D_{i,s} \leq \bar{t}_{i,s+1} \qquad i \in \mathcal{V}^E, s \in \mathcal{S} \backslash \{\bar{s}\} \tag{13}$$

$$\bar{t}_{i,s} + D_{i,s} \leq \bar{t}_{i,s-1} \qquad i \in \mathcal{V}^W, s \in \mathcal{S} \backslash \{0\} \tag{14}$$

The time-corridor shrinks by the amount of time the ship has to wait. If the entire time-corridor is consumed, the ship will no longer have a corridor. Instead, it will have a strict travel schedule similar to as in a deterministic planning environment, such as in Lübbecke et al. (2019) and Meisel and Fagerholt (2019). Clearly, as the time-corridor cannot become negative, Constraints (12) ensure that the latest entering time is bounded by the planned entering time. Constraints (13) and (14) then propagate the latest entering time from a siding s to the next segment.

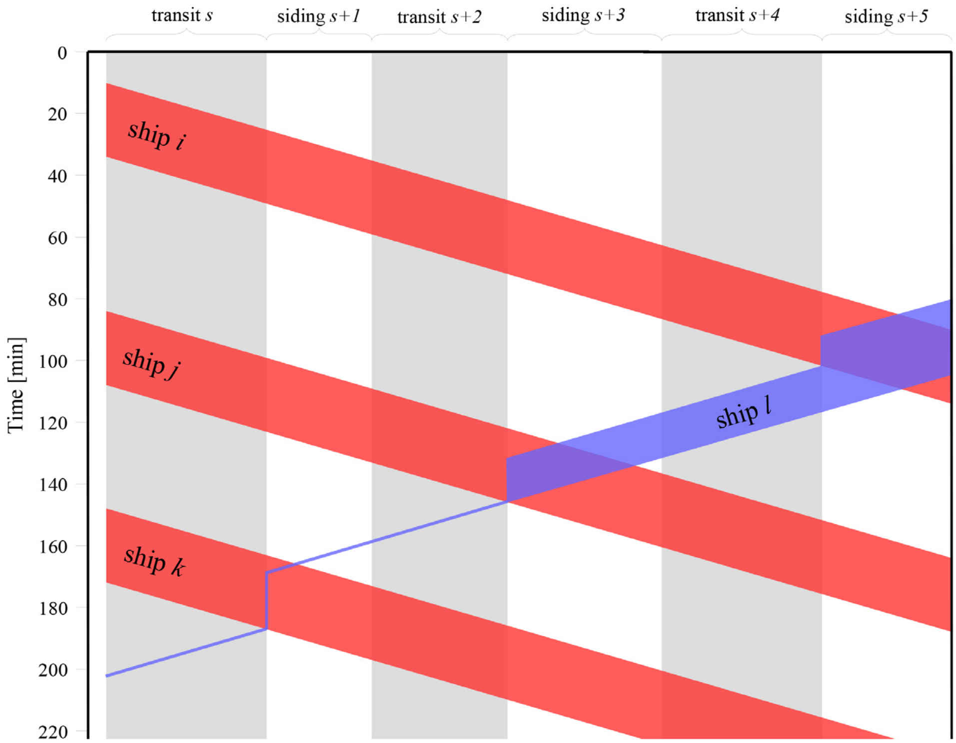


Fig. 6. An example of the consumption of time-corridors.

Fig. 6 shows an example of how waiting affects the time-corridor. There are three eastbound ships i, j and k as well as one westbound ship l. Initially, all four ships have time-corridors of 20 minutes each. In the depicted solution, the westbound ship l has to give priority to all three eastbound ships. For this reason, it has to wait in each of the sidings $s+5$, $s+3$, and $s+1$. In siding $s+5$, it has to wait until ship i's corridor has fully left the transit segment $s+4$. This shrinks l's time-corridor by about 7 minutes. In siding $s+3$, it has to wait until ship j's corridor has fully left the transit segment $s+2$. This waiting fully consumes l's remaining time-corridor, which is then of width 0. Eventually, in siding $s+1$, ship l has to wait until ship k's corridor has fully left the transit segment s. Here, the time-corridor of l is no more consumable, which is why the 0-width corridor is now shifted along the time axis, as is handled by Constraints (12) to (14). The depicted solution remains completely feasible as long as all four ships arrive within their respective time-corridors.

$$\bar{t}_{i,s} + D_{i,s} = \bar{t}_{i,s+1} \qquad i \in \mathcal{V}^E, s \in \mathcal{T} \tag{15}$$

$$\bar{t}_{i,s} + D_{i,s} = \bar{t}_{i,s-1} \qquad i \in \mathcal{V}^W, s \in \mathcal{T} \tag{16}$$

Since a ship cannot wait inside a transit segment, the time-corridor does not shrink when traveling from a transit to a siding. This is taken care of by Constraints (15) and (16).

$$\bar{t}_{i,s} + \Delta_{i,j,s} \le t_{j,s} + M \cdot (1 - z_{i,j,s}) \qquad s \in \mathcal{T}, (i,j) \in C_s^A \cup C_s^O \tag{17}$$

$$\bar{t}_{j,s} + \Delta_{j,i,s} \le t_{i,s} + M \cdot z_{i,j,s} \qquad s \in \mathcal{T}, (i,j) \in C_s^A \cup C_s^O \tag{18}$$

Constraints (17) and (18) are the precedence constraints and are only defined for the transit segments as traveling in sidings does not raise conflicts. If ship i enters segment s before ship j, i.e. $z_{i,j,s} = 1$, Constraints (17) force the entering time for ship j to be later than the latest entering time of ship i (i.e. the end of i's time-corridor) plus the safety time that has to elapse to make sure that ships i and j can safely traverse the segment. If ship j gets priority over ship i, Constraints (18) ensure that ship j enters first.

$$t_{i,s}, \bar{t}_{i,s} \ge 0 \qquad i \in \mathcal{V}, s \in \mathcal{E} \tag{19}$$

$$w_{i,s} \ge 0 \qquad i \in \mathcal{V}, s \in \mathcal{S} \tag{20}$$

$$z_{i,j,s} \in \{0,1\} \qquad s \in \mathcal{T},\ (i,j) \in C_s^A \cup C_s^O \tag{21}$$

Constraints (19) and (20) are the non-negative constraints for the decision variables. Constraints (21) set the precedence variables to be binary.

4. Iterative conflict-Adding matheuristic

The mixed integer programming (MIP) model provided in Section 3.4 is hard to solve due to the large number of binary z-variables and Constraints (17) and (18), which are needed to resolve all potential conflicts. The size of the problem increases strongly with an increase in the number of ships and it also becomes harder to solve with increased width of time-corridors. As a result of this, a MIP-solver can only solve small problem instances. We therefore propose a matheuristic, which we denote as the Iterative Conflict-Adding Matheuristic (*ICAM*) for solving the *SSKC-UAT*.

The basic framework of *ICAM* is taken from Meisel and Fagerholt (2019). We present this in Algorithm 1 to make the paper self-contained. We want to note that the width of time-corridors is given as an input parameter to the heuristic (and respected when solving the embedded optimization model). A planner can rerun the heuristic for several different widths to identify the one that best suits the arrival time uncertainty he/she is faced with. We illustrate this later in our computational experiments. By introducing time-corridors, the complexity of the algorithm increases since the exact position of each ship is no longer fully known. This requires a more detailed conflict detection through Algorithm 2 , which is later explained. The idea behind *ICAM* is to first solve the problem with empty sets C_s^A and C_s^O of aligned and opposing conflicts, as described in step 2 in Algorithm 1. Without precedence constraints, the problem solves quickly with a commercial solver as it is a continuous linear program only. All conflicts that make the solution

Input: A problem instance
1. Initialization: $C_s^O \leftarrow \emptyset$ and $C_s^A \leftarrow \emptyset$ for all segments s
2. Solve the optimization model using a standard MIP-solver.
3. Identify the set of *ConflictsFound* in the current solution using Algorithm 2.
4. **while** *ConflictsFound* $\neq \emptyset$ **do**
    5. *FirstConflicts* $\leftarrow$ up to the v first conflicts in *ConflictsFound*
    6. **forall the** conflicts *fc* $\in$ *FirstConflicts* **do**
        7.**if** *fc* is a conflict of opposed ships on segment s **then**
            8. $C_s^O \leftarrow C_s^O \cup fc$
        end
        9. **if** *fc* is a conflict of aligned ships on segment s **then**
            10. $C_s^A \leftarrow C_s^A \cup fc$
        end
    end
    11. Fix feasible part of current solution up to the earliest conflict τ.
    12. Solve the optimization model using a standard MIP solver.
    13. Identify the set of *ConflictsFound* in the current solution using Algorithm 2.
end
return The obtained solution which is a feasible travel plan for all ships.

Algorithm 1: The Iterative Conflict-Adding Matheuristic (*ICAM*).

Input: Current solution
ConflictsFound $\leftarrow \emptyset$
forall the *pairs of ships i and j* **do**
    forall the *transit segments* **do**
        if *precedence constraints (17) and (18) are violated* **then**
            if *ships i and j are aligning* **then**
                if *there already is a conflict between ships i and j* **then**
                    check time of conflicts and keep earliest one
                end
                else
                    add conflict to *ConflictsFound*
                end
            end
            if *ships i and j are opposing* **then**
                add conflict to *ConflictsFound*
            end
        end
    end
end
return *ConflictsFound*

Algorithm 2: Conflict detection procedure.

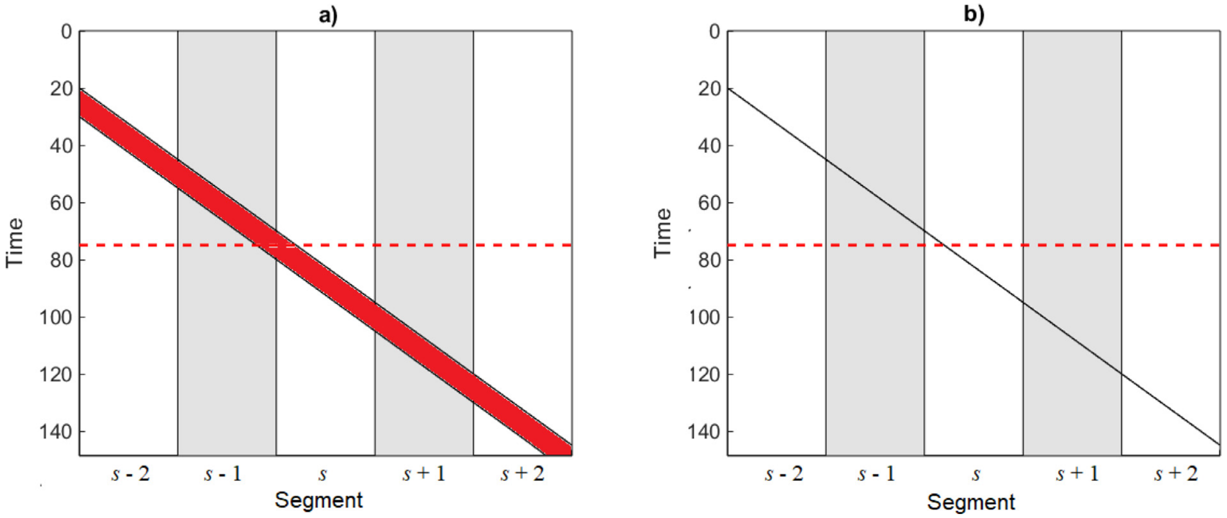


Fig. 7. a) When the formulation includes time-corridors, the ship can be in either the dark or the light segment at time 75. b) In a formulation without time-corridors, as used in Meisel and Fagerholt (2019), the ship can only be in the light segment at time 75.

infeasible are then identified and sorted in chronological order. As shown in steps 3 to 10 in Algorithm 1, the conflict set called *ConflictsFound* contains all conflicts found in the current solution. A defined number of conflicts are stored temporarily in the set *firstConflicts* and these are added to either the aligning or the opposing conflicts sets C_s^A and C_s^O of the corresponding segment s. The schedule is then fixed up to the time of the first new conflict, which is denoted by τ. This means that some ships cannot have parts of their schedule changed in later iterations, which will help reduce the complexity of the MIP-problem solved in each iteration. The problem is then resolved with the updated sets of aligned and opposing conflicts and fixation of all ships' schedules up to time τ.

Multiple Possible Ship Positions

When all ships are given an initial time-corridor, it is not clear in which particular segment a ships is at a given time. In Fig. 7a), the two black lines represent the upper and lower boundary for a ship's position. The ship has a time-corridor of ten time minutes, i.e. the lower line is ten time units below the upper line. As seen by the dotted horizontal line, which refers to point in time 75, the ship can be in one out of two different segments. It can be in the transit segment $s-1$ or in the siding segment s. This is different from the situation in Meisel and Fagerholt (2019), where a ship travels along an exactly defined path where it is in exactly one segment at a given time, see the example in Fig. 7b).

When the width of a ship's time-corridor increases, the number of possible locations for this ship may increase as well. This results in larger conflict sets and more binary precedence variables to resolve all potential conflicts.

How conflicts are detected in each iteration in the *ICAM* with time-corridors for the ships is outlined in Algorithm 2. Since each ship may have a time-corridor in our problem, both the earliest and latest planned time are used to identify conflicts. Each pair of ships i and j is checked to see if there is a conflict between them. This is done on all transit segments by checking which ship enters the segment first, and the precedence variables $z_{i,j,s}$ and $z_{j,i,s}$ are then temporarily set according to this. Furthermore, the precedence constraints (17) and (18) are then examined to identify possible constraint violations. If a constraint has been violated, there is a conflict, and the directions of the ships are used to determine if the conflict is aligning or opposing.

Before the conflict is added to the set *ConflictsFound*, it is checked if the same two ships have another conflict on another segment. In this case, only the conflict that occurs first is added. The reason for this is that if two aligning ships have a conflict on a segment (as they move too close to each other), they will most likely have conflicts on all the following segments too.

The conflict detection algorithm used by Meisel and Fagerholt (2019) detects conflicts by running through the ships' positions at certain time increments. Since the ships can be in multiple possible positions when applying time-corridor scheduling, the same detection algorithm was not suitable for our case. The main difference between Meisel and Fagerholt (2019) and the one proposed in this paper is that Meisel and Fagerholt (2019)'s detection algorithm runs chronologically through the conflicts and stops when it has found the predefined number of conflicts, v. In comparison our algorithm first finds all conflicts before sorting them in a chronological order and returns the first v conflicts.

Mechanisms for Speeding up the Solution Process

Meisel and Fagerholt (2019) have proposed three mechanisms for speeding up the solution process, which are also employed in our version of the algorithm. One approach is to add only a predefined number of conflicts in each iteration instead of all the conflicts that are detected. Parameter v in Step 5 of Algorithm 1 controls how far ahead in time conflicts should be added. As such, it controls the size and complexity of the MIP-problem that is solved per iteration. Here, low v-values result in small MIP-subproblems to solve but it may require a larger number of iterations to solve the problem completely. Larger values of v create larger subproblems but require less iterations. Furthermore, v affects the overall solution quality. It is therefore essential to choose it appropriately to obtain both low total solution time and satisfactory solution gaps.

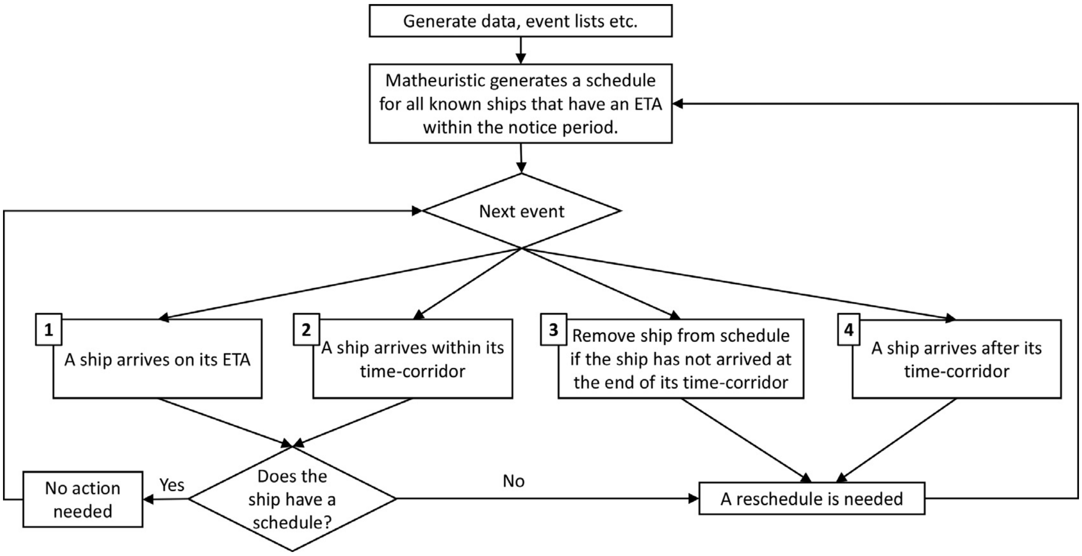


Fig. 8. Flowchart for the simulation framework where ship arrival times are uncertain.

Another way to reduce the solution time of the *ICAM* is to fix the schedule up to the time of the earliest conflict, see Step 11 of Algorithm 1. The conflict time is denoted τ and determined when the conflicts of the current solution are identified. The time is set to be the time of the earliest entering time of the segment where the conflict occurs for the two ships. As we introduce time-corridors into the problem, we fix in our version of the algorithm both entering times, the earliest $t_{i,s}$ and the latest $\bar{t}_{i,s}$ that are earlier than τ for any ship i and segment s. By fixing the part of the solution up to the earliest conflict, the runtime per iteration will stay almost constant, since previous binary precedence variables have already been fixed.

Finally, when solving the MIP-model in Step 12 of Algorithm 1, one can prescribe a maximum runtime as it is not mandatory to solve the subproblems to optimality within a heuristic framework.

5. Simulation framework

The *SSKC-UAT* is in Section 3.4 formulated as a static optimization problem. However, in practice it is dynamic and must be solved over and over again as new information (e.g. about delays leading to conflicts) appears over the course of time. Therefore, we want to test how the *ICAM* heuristic works within a rolling horizon simulation framework to study the effect of arrival time uncertainty. This section describes the simulation framework in which this testing is done.

The current practice at the canal is that ship captains are asked to call in their estimated *ETA* some time prior to the arrival at the canal. A typical announcement lead time of the *ETA*s is around two hours ahead of the arrival. This time is denoted the *notice period*. For example, if the notice period is exactly two hours, a ship with an ETA of 17:23 becomes known to the canal operators at time 15:23 and, therefore, this ship will be part of any planning that is conducted at this time or later.

There are several different aspects that must be considered in order to create a simulation process that realistically resembles the real-life scheduling on the canal. A flowchart for the simulation process can be found in Fig. 8. There are four different events that can occur, numbered from 1 to 4 in Fig. 8. Some of these event lead to a rescheduling, i.e. that a revised schedule is obtained from solving the *SSKC-UAT* for all ships known at the current time with the most up-to-date information available.

Referring to Fig. 8, if a ship arrives on its *ETA*, the event will be of type 1. If the ship arrives later than *ETA* but still within its time-corridor, the event will be of type 2. Neither of these event type requires a rescheduling if the ship is already included in the current solution as this plan remains then feasible in both cases. Anyhow, distinguishing both event types helps to analyze the value of time-corridors. More precisely, ships that trigger event 1 would be feasible only in a deterministic planning whereas ships that trigger event 2 would make a non-robust solution infeasible. It is therefore the introduction of time corridors that keeps the solution feasible in these situations.

Anyhow, if a ship is delayed even beyond its time-corridor, there will be two events of type 3 and 4. First, when a delayed ship has not arrived at the end of its time-corridor, an event 3 is triggered at that time (i.e., *ETA* plus the length of the time-corridor). The ship will then be removed from the current solution and a reschedule is initiated to re-optimize the routes of the other ships. When the delayed ship finally arrives, an event of type 4 will initiate a rescheduling to insert this ship into the solution.

For conducting the simulation studies, we generated ship data as follows. We have used a discrete distribution for assigning traffic group numbers to each arriving ship. Based on historical data, we have set the probabilities to 0.005, 0.03, 0.495, 0.25, 0.21, and 0.01 for ships' traffic group numbers 1, 2, 3, 4, 5, and 6, respectively. The *ETA* is assumed to be the time when the ship is expected to enter its first segment of the canal, i.e. after the ship has passed through the lock. From analysing real-world data that was provided by the canal authority, it was found that the arrival times of two consecutive ships are not independent. This is caused by how the

Table 1
The results of the *ICAM* compared to the results obtained with the Xpress MIP-solver under a time limit of 600 seconds. Each row in the table is based on the average of ten test instances.

		Xpress				ICAM		
# of ships in instance	TC [min]	Avg objective [min]	# of optimal solutions	Avg gap [%]	Solution time [sec]	Avgchange [%]	Avg gap [%]	Solution time [sec]
20	0	8257.91	9	0.15	85.60	0.16	0.31	2.26
	10	8412.19	9	0.21	142.42	0.19	0.40	3.07
	25	8663.10	8	0.40	239.17	0.27	0.67	3.40
	Average	**8444.40**	**8.7**	**0.25**	**155.73**	**0.21**	**0.46**	**2.91**
30	0	12342.52	7	0.19	266.32	0.19	0.38	3.89
	10	12606.40	2	0.99	519.39	0.09	1.07	5.13
	25	13008.58	0	1.95	600.00	0.55	2.49	6.76
	Average	**12652.50**	**3.0**	**1.04**	**461.90**	**0.28**	**1.31**	**5.26**
40	0	17064.40	0	3.61	600.00	-0.78	2.85	35.51
	10	17588.75	0	5.32	600.00	-1.70	3.68	33.96
	25	18319.81	0	7.08	600.00	-2.24	4.95	37.53
	Average	**17657.65**	**0.0**	**5.34**	**600.00**	**-1.57**	**3.83**	**35.66**

locks are operated, more specifically that several ships are allowed through the locks simultaneously. A group of ships that enters into the first canal segment at around the same time is referred to as a *batch*.

We use exponential probability distributions to create the inter-arrival times between two consecutive ships inside a batch, as well as the time between two batches. The distributions are chosen so as to give a realistic arrival pattern and a yearly traffic of around 30 000 ships. Based on the received historical data, the traffic flow in both directions accounts for approximately 50% each with an almost constant flow of ships in both directions. However, since the ships arrive in batches, it is not possible to randomly draw a direction for each ship, as this will not guarantee realistic batch behavior. We therefore draw the directions of the batches instead of the individual ships. We have also looked at the historical data to create realistic distributions on the batch size, which can vary from 1 to 4 ships.

Since the canal operators did not provide data on ship delays, we have used estimates for the simulation study. We have assumed that the arrival time delays are independent and identically distributed among the ships. The percentage of ships that enter the canal punctually at their *ETA* is denoted α, whereas 1-α percent of the ships are delayed. The lengths of these delays are drawn from an exponential distribution with an expected value of d minutes. The values of parameters α and d are later varied in our experimental study.

6. Computational study

The computational study is divided into two parts. In Section 6.1, we set the parameters of the *ICAM* matheuristic and compare its performance to using the commercial MIP-solver FICO Xpress for solving the model in Section 3.4 directly. Afterwards, in Section 6.2, the simulation framework is used to test the effect of introducing time-corridors in a stochastic environment. The objective is to study how time-corridors affect the total traversing time of the ships and how many *reschedules* need to be performed during the planning periods, i.e. the number of times the *SSKC-UAT* needs to be resolved as a schedule turned infeasible. We also study the importance of the length of the notice period, i.e., how early in advance the ships call in their *ETA*s, and the effect of extending certain canal segments on the traversing times of the ships.

All simulations are conducted on Lenovo M5 computers with 3.4 GHz processors and 512 Gb of memory. The BCL-libraries developed by FICO® Xpress have been used to implement the *ICAM* through C++. For the heuristic, we set the maximum runtime per iteration to be 5 seconds and the number of conflicts added per iteration to $\nu = 20$. These parameters were determined in a pretest like the one conducted in Meisel and Fagerholt (2019).

6.1. Evaluation of the iterative conflict-Adding matheuristic (ICAM)

We next test how well the *ICAM* performs and the commercial MIP-solver perform for test instances of different size and different initial time-corridor widths for the ships. Ten test cases with 20, 30 and 40 ships were generated based on real historical data. As the yearly traffic through the Kiel canal is about 30 000 ships and each ship spends about 8 hours in the canal, an average of 27 ships is in the canal at a time. Therefore, test cases with 30 ships represent an average traffic density whereas instances with 20 ships are low traffic density and 40 ships represent busy traffic situations. We consider the initial time-corridor widths TC of 0 (i.e., there are no time-corridors), 10 and 25 minutes per ship, resulting in a total of 90 test instances. The test data is provided to other researchers upon request.

The results are summarized in Table 1, where each row represents average results over ten instances of same size and corridor width. For the MIP-solver, the average objective value, the number of instances solved to optimality, the average gap, and the average solution time are shown. The gap is defined as the relative difference between the best solution and the best bound, where the Xpress solver is given a runtime of 600 seconds per instance. For the ICAM matheuristic, column 'Avg change' is calculated as the relative difference between the solution found by *ICAM* and the Xpress objective value (negative numbers means that the heuristic solution is better), while column 'Avg gap' gives the relative difference between the *ICAM* solution and the best bound obtained from the MIP-solver. In addition to conducting these tests with a runtime of 600 seconds, we ran two test instances with 40 ships each and time-corridors of 25 minutes, with the exact solution method (Xpress MIP-solver) for a total of 18 hours. The gaps for these two instances were reduced from 9.95% and 4.41% after 600 seconds to 6.19% and 2.89% after 18 hours, respectively. As the reduction of gaps after the extreme long runtime is very minor and mostly due to improvements of lower bounds rather than improvements of the objective function, it seems reasonable to conduct the comparisons of the *ICAM* and the exact solution method at a 600 seconds runtime. It can also be mentioned that since this is an operational planning problem that has to be solved every time a new ship is approaching the canal, 600 seconds can be considered as a practical upper limit on the maximum allowed solution time.

As shown in Table 1, the overall performance of the *ICAM* is very good. For the instances with 20 and 30 ships, the matheuristic gives solutions that are just slightly worse than the ones obtained by the MIP-formulation whereas for the cases with 40 ships, *ICAM* performs significantly better than MIP-Solver Xpress. Regarding the role of time-corridors, we observe that the objective values increase (as expected) and that gaps get somewhat larger for wider corridors. Furthermore, runtimes of Xpress clearly grow with larger time-corridors whereas runtimes of $ICAM$ stay almost constant for different widths of corridors. Although the runtimes of the heuristic grow substantially for larger instances here, it is much faster than Xpress in all cases. Due to this performance, *ICAM* is very well suited to be used in the simulation framework.

6.2. Simulations with arrival time uncertainty

In the subsequent experiments, we analyze the effect of introducing time-corridors on the total traversing time of the ships and the number of reschedules. We also investigate the effect of the length of the notice period (i.e., the time in advance the ships will call in their *ETA*s) to see if it may be beneficial to seek for longer notice periods. Finally, we conduct an analysis on how possible upgrades in the canal can improve the average traversing time.

For the simulations, we used a data set of about two months (63 days) of ships traffic. The first day is taken out of the performance analysis as it serves as warmup period. Thus each simulation has 62 days of full traffic.

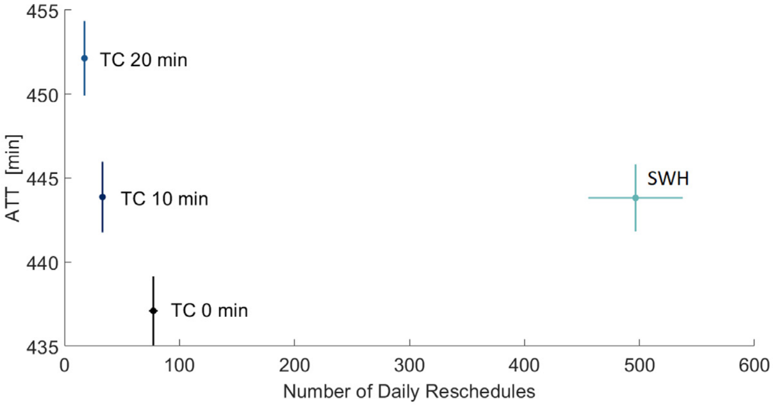
Four different simulations are conducted to analyze the effect of introducing time-corridors on the total traversing times for the ships and the number of reschedules. The first simulation is run using a heuristic that reflects the current scheduling practice to create some benchmarks to compare with. We denote this heuristic as the Simple Waiting Heuristic (*SWH*). The *SWH* solves conflicts chronologically as they occur by forcing one of the two ships to wait. It always selects the ship that gets the smallest waiting time without considering conflicts that appear later. The algorithmic complexity of this procedure is low, hence the computational time is very short per conflict resolution.

The three other simulations use the *ICAM* as the solution method with different time-corridor widths TC of 0, 10 and 20 minutes. The delay parameters used during these simulations are $\alpha = 0.5$ and $d = 10$, meaning that 50% of the ships are assumed to arrive later than their *ETA*s and that the expected delay among the delayed ships is ten minutes.

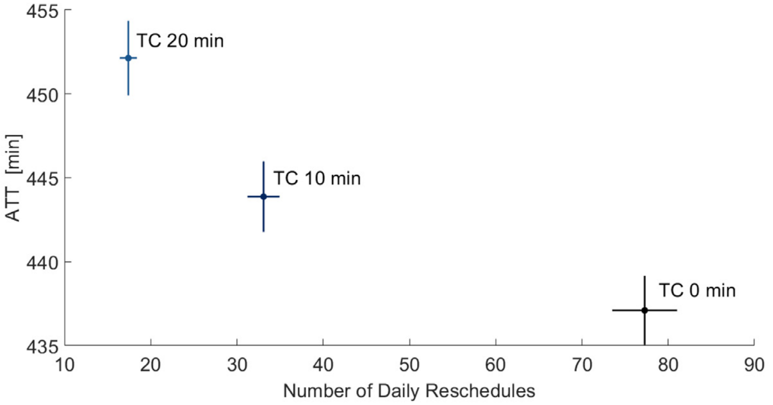
The results from these simulations are summarized in Fig. 9 a). This figure shows (with a 95% confidence interval) the average traversing time (ATT) for the ships and the daily number of reschedules from each of the four simulations. Comparing the two heuristics without any time-corridors (width 0), we see that *SWH* leads to an average traversing time of 444.82 minutes whereas *ICAM* achieves an average traversing time of 437.10 minutes. This is a reduction of almost 8 minutes. However, a much larger improvement is found in the number of reschedules, which goes from 496.61 down to 77.27. It should however be noted that the complexity of these two procedures are very different. The *ICAM* spends much longer time per full reschedule than what the *SWH* uses to resolve a specific conflict between two ships. Nevertheless, the *SWH* might be considered more tedious than the *ICAM* by the traffic managers working at the canal authority. Since the number of reschedules is so much larger for the *SWH* than for any of the *ICAM* results, we removed in Fig. 9 b) the *SWH* results to show more clearly the differences among the results obtained by the *ICAM* under different time-corridors.

As can be seen in Fig. 9 b), the average traversing time per ship increases with the width of the time-corridors, as a time-corridor given to one ship can restrict the traveling of other ships. This increase reflects the 'price of robustness' in our solutions. Time-corridors of 10 minutes lead to an increase of about six minutes compared to no time-corridors at all. With time-corridors of 20 minutes, the increase of ATT is about 15 minutes. Anyhow, the average traversing time achieved by *ICAM* with time-corridors of 10 minutes is still slightly smaller than the one obtained with the *SWH* and, at the same time, much more robust. In general, there is a significant decrease in the number of reschedules when the width of the time-corridors increases. The number of reschedules is reduced by more than 50% with time-corridors of 10 minutes compared to no time-corridors, and the number of reschedules is further decreased when the time-corridors are set to 20 minutes. Based on the results from these simulations, it might therefore be beneficial to include short time-corridors for each ship since this will decrease the need for reschedules and only give a modest increase in the average traversing time.

To further test the effect of the time-corridors, we have conducted additional simulations for three configurations of the delay parameters that represent different situations regarding the uncertainty in arrival times. More precisely, we consider situations (a) where 50% of the ships are delayed and the expected delay for delayed ships is 10 minutes, (b) where 50% of the ships are delayed and the expected delay for delayed ships is 20 minutes, and (c) where 20% of the ships are delayed and the expected delay for delayed ships is 20 minutes. For all three situations, we varied the time-corridor width from 0 to 25 minutes. Fig. 10 displays the simulation results.



(a) 95 % confidence interval for the four different simulations.



(b) 95 % confidence interval for the simulations where the $ICAM$ has been used with time-corridors TC of 0, 10 and 20 minutes.

Fig. 9. 95% Confidence Intervals for the number of daily reschedules and the average traversing time *ATT*. The *SWH* is excluded in b) to more clearly show the differences for the simulations with *ICAM* under different widths of time-corridors TC.

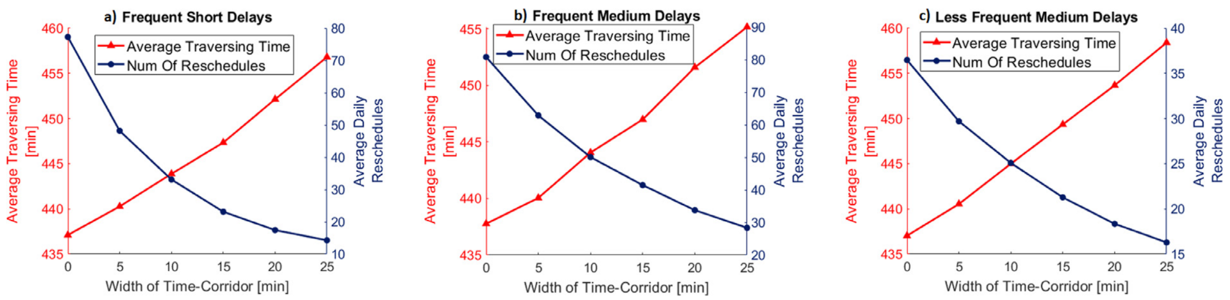


Fig. 10. Average traversing time and daily number of reschedules for different delay situations.

Both the average traversing time and the number of daily reschedules share the same properties in all the three plots in Fig. 10. The average traversing times increase with the width of the time-corridors and are quite similar among the three sets of delay parameters. With time-corridors of zero, we see that all three uncertainty settings give an average traversing time of about 437 minutes. The difference between them increases slightly as the width of the time-corridors increases, but this difference is still small. It therefore seems that the width of the time-corridors affects the average traversing time more than the different delay parameters do.

The number of daily reschedules decreases for larger time-corridors. Unlike for the average traversing time, the delay parameters affect the number of daily reschedules. In the case where 50% of the ships arrive on time (Figs. 10 a) and b)), around 80 reschedules will be needed each day if no time-corridors are applied. With 80% of the ships arriving on time (Fig. 10 c)), this number decreases to 36 reschedules. Based on these results, it seems that the preferred width of the time-corridors should be based on the expected delay. Which time-corridor width to select within this interval depends on how the operators value the average traversing time and the burden of rescheduling.

Table 2
Delay parameters for different notice periods lengths under two delay settings.

Length of notice period [min]	Setting 1 Expected delay for delayed ships d [min]	Setting 2 Expected delay for delayed ships d [min]
60	5	10
120	10	10
180	15	10
240	20	10

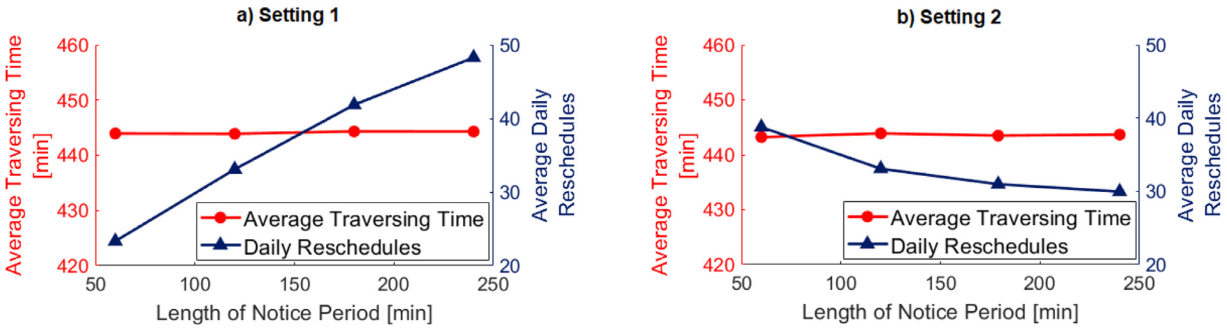


Fig. 11. Simulation results for different notice period lengths under two different delay assumptions.

All previous simulations were conducted with a notice period of two hours for the arrival of a ship. It is of interest to see if the average traversing time of the ships can be improved by extending the length of this notice period. To investigate this, we conducted simulations for four different notice period lengths of 60, 120, 180 and 240 minutes. Time-corridors of ten minutes are used for all these tests together with an α-value of 0.5, i.e. 50% of the ships arrive on time.

Two different settings regarding how the delays vary as the notice period changes have been tested. Setting 1 is based on the assumption that the expected delay is proportional to the length of the notice period. This assumption is built on the belief that the probability of experiencing a delay increases by the length of time the ship is exposed to unforeseen events before entering the canal. Setting 2 is based on the assumption that the expected delay is independent of the length of the notice period. This assumption follows the fact that certain delay-inducing events have the same probability of occurring regardless of how early the ship's *ETA* has been announced. For example, ships must occasionally wait before they are allowed to enter the locks and this waiting time is realized just upon arrival at the canal. The delay will in this case be independent of the notice period. Table 2 shows how the expected delays are set for the four considered notice periods under both settings. We furthermore keep $\alpha = 0.5$.

The average traversing times and the average number of daily reschedules for the simulations based on the two settings are shown in Fig. 11. It can be seen that the average traversing time remains relatively constant for the different lengths of notice periods under both settings. As for the number of daily reschedules, the two settings affect this number differently. Under setting 1, the number of reschedules increases with increased length of the notice period. This seems natural as the expected delay of the ships is modeled to increase proportionally to the length of the notice period. Still, the number of reschedules remains clearly below the numbers observed for the simple SWH heuristic, such that using a more advanced planning by $ICAM$ leads to much more robust schedules even under the challenging situation of delays that grow with longer notice periods.

As seen in Fig. 11 b), the effect of longer notice periods is positive with respect to the number of daily reschedules under setting 2. This shows that early announcements of ship arrivals together with relative constant actual delays is successfully exploited in our approach to obtain even more robust schedules.

We next analyse the waiting of ships in the canal. For this purpose, we simulated ship traffic for a 2-month period, which involves about 5000 ships. Fig. 12 shows for each siding of the canal, the total waiting time of all ships (measured in hours), the longest waiting time of any ship (measured in minutes), and the number of ships that waited in this segment. We report these results for *ICAM* with time-corridors of 0, 10 and 20 minutes and for *SWH* without time-corridors and for a notice period of 2 hours. It can be observed that the largest differences regarding the number of ships waiting among the different solution methods are in segments 0 and 22. The total waiting in these segments increases dramatically with the width of the time-corridors. This is because aligning ships that arrive almost at the same time must wait for the time-corridor for the ship in front of it to end. As the width of the time-corridors increase, more ships must wait for a longer time.

Based on Fig. 12, which displays the heatmap with different waiting measures, it seems like that transit segments number 13, 15, 17, 19 and 21 are restraining the ship flow. These transits have passage numbers equal to 6, while all other transits have passage numbers equal to 8. Note that segment 15 has passage number 8 in Fig. 2, but this extension was quite recently which is why it has passage number 6 in our experiments. Since the waiting is much higher at the adjacent sidings to transit segments with passage number 6, it can be of interest to analyze the effect of upgrading these segments. To analyse this, we conducted multiple simulations for the 2-month period with time-corridors of ten minutes where $ICAM$ was used as solution method. In each simulation *one* of the very narrow segments was changed to a passage number of 8. The results are presented in Table 3. Additionally, a simulation where

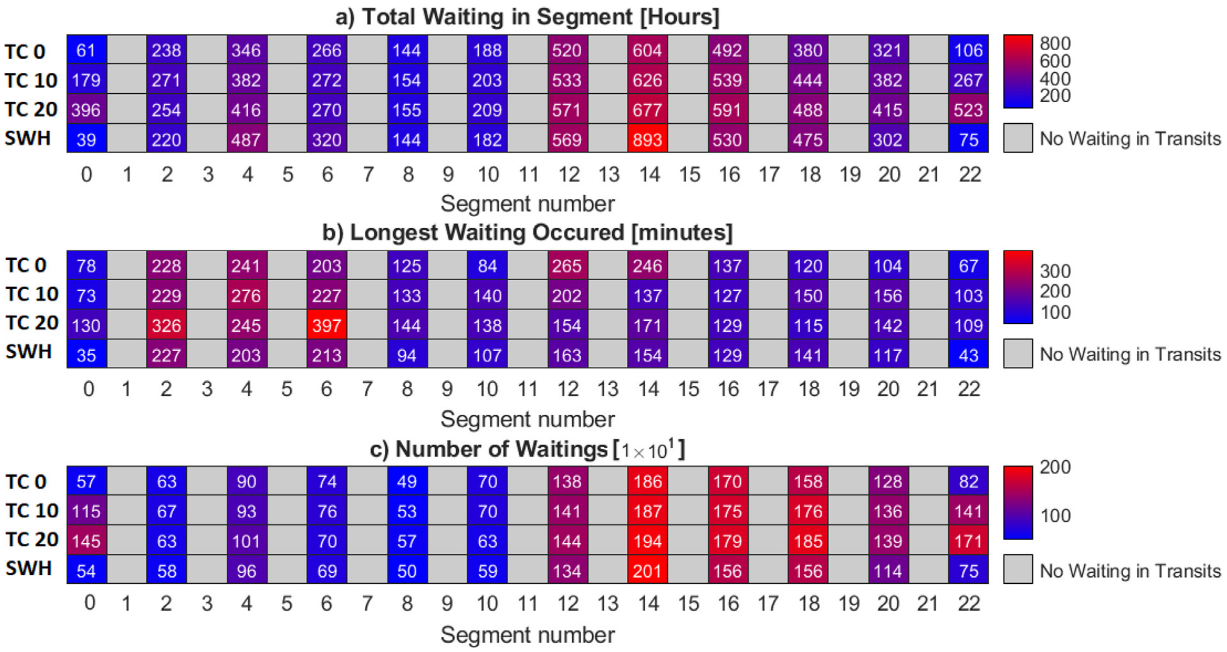


Fig. 12. Waiting time measures for simulated traffic schedules obtained by *ICAM* for three different time-corridor widths (0, 10 and 20 minutes) and obtained by *SWH* (without time-corridors). Please note that the units differ for the three different waiting measures.

Table 3
Improvement in traversing time by upgrading certain transits from passage number 6 to 8.

	Average traversing time $[min]$	Reduction in transit time $[\%]$	Transit length $[meter]$	Impact per kilometer $[min/km]$
Original topology	443.87	-	-	-
Transit 13	437.92	1.34%	8170	0.728
Transit 15	438.91	1.12%	7510	0.660
Transit 17	440.37	0.79%	4406	0.794
Transit 19	440.30	0.81%	6150	0.580
Transit 21	442.76	0.25%	2412	0.460
All transits upgraded	424.39	4.39%	28 648	0.680

all the transits have been upgraded to a passage number of 8 has been conducted. Both the average traversing time in the original layout and the average traversing times after the upgrades are presented in Table 3. The reduction in traversing times relative to the original canal layout are also shown. The length of the different segments varies extensively and since the upgrading cost may be proportional to the length of the segment, we also provide information about the minutes saved per kilometer of segment length.

The greatest improvement can be obtained by upgrading transit 13 that gives around six minutes reduction in average traversing time. This may seem small, but with an annual traffic of 30 000 ships, it will reduce the total traversing time for all ships by around 3 000 hours per year. Upgrading transit segment 17 yields the highest improvement per kilometer of upgrade. Upgrading transit segment 13 gives the second highest improvement per kilometer and the highest overall improvement per segment. Interestingly, transit segment 15, which was recently extended to passage number 8, only achieves the second or third rank, depending on which performance improvement measure one considers. Eventually, by jointly upgrading all considered transit segment to passage number 8, it is possible to save more than 19 minutes in the average traversing time. This corresponds to an estimated reduction in the total traversing time for all ships of almost 10 000 hours per year. Next to the time-saving analysis that is conducted here, upgrading segments is of course also an issue of construction costs etc., which, however, is beyond the scope of this paper.

7. Concluding remarks

We have studied the scheduling of ships through the Kiel Canal when the ships' arrival times at the canal are subject to uncertainty. Such uncertainties give frequent need of replanning to keep schedules feasible. To tackle this challenge, we have proposed a mathematical model that incorporates time-corridors, so that schedules stay valid as long as ships arrive within the given time-corridors. The problem is solved by an *Iterative Conflict Adding Matheuristic* (*ICAM*), which was shown to produce very good solutions for real-world instances in short time.

The *ICAM* has also been embedded into a rolling horizon simulation framework. We showed by experiment how to identify suitable widths of the time-corridors that reduce the number of reschedules without increasing the ships' transit times too much. For comparison, we simulated the current scheduling practice through a so-called simple waiting heuristic (*SWH*). The quality of the schedules was measured both on how many times a rescheduling is required per day and on the average traversing time of ships. By introducing the *ICAM* in the scheduling of ship traffic, both the average traversing time and the number of reschedules decrease compared to the schedules of *SWH*. The results also show that even with time-corridors of ten minutes, the average traversing time stays below the one achieved by the *SWH*, but that the number of daily reschedules is reduced dramatically from almost 500 to around 33. If we compare these results with the *ICAM* heuristic without time-corridors (i.e. time-corridors of ten vs. zero minutes), we obtain a reduction in the number of daily reschedules from around 77 to 33, though at the cost of an increase in the average traversing times of almost seven minutes per ship. Based on these results, it is clear that the use of time-corridors reduces the need for frequent reschedules and might therefore be preferred over solution methods without any protection against uncertainty in the arrival times.

We have also used the rolling horizon simulation framework to analyze how the average traversing times for the ships and the number of daily reschedules are affected by the notice time (i.e. the time in advance the ships call in their estimated times of arrival). It was shown that the average traversing times did not depend much on the notice periods, while the number of daily reschedules did. Furthermore, we investigated how the average traversing time can be decreased by upgrading transit segments along the canal. We found that upgrading narrow transit segments from passage number six to passage number eight, reduced the average traversing time per ship between one and six minutes. When upgrading all the narrow transit segments together, the average traversing time can be reduced by more than 19 minutes, which is a reduction of 4.4% of the total traversing times.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alfandari, L., Davidović, T., Furini, F., Ljubić, I., Maras, V., Martin, S., 2019. Tighter MIP models for barge container ship routing. Omega (Westport) 82, 38–54. doi:10.1016/j.omega.2017.12.002.

Cacchiani, V., Toth, P., 2012. Nominal and robust train timetabling problems. Eur J Oper Res 219 (3), 727–737. doi:10.1016/j.ejor.2011.11.003.

Castillo, E., Gallego, I., Ureña, J.M., Coronado, J.M., 2011. Timetabling optimization of a mixed double- and single-tracked railway network. Appl Math Model 35 (2), 859–878. doi:10.1016/j.apm.2010.07.041.

Cordeau, J.-F., Toth, P., Vigo, D., 1998. A survey of optimization models for train routing and scheduling. Transportation Science 32 (4), 380–404. doi:10.1287/trsc.32.4.380.

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. doi:10.1016/j.cie.2015.03.014.

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

Heitmann, N., Rehdanz, K., Schmidt, U., 2013. Determining optimal transit charges: the Kiel Canal in germany. J. Transp. Geogr. 26, 29–42.

Jovanović, P., Kecman, P., Bojović, N., Mandić, D., 2017. Optimal allocation of buffer times to increase train schedule robustness. Eur. J. Oper. Res. 256 (1), 44–54. doi:10.1016/j.ejor.2016.05.013.

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., Natvig, E., 2017. An exact micro-macro approach to cyclic and non-cyclic train timetabling. Omega (Westport) 72, 59–70. doi:10.1016/j.omega.2016.11.004.

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., 2019. Ship traffic optimization for the Kiel Canal. Oper. Res. 67(3), 791–812.

Lusby, R.M., Larsen, J., Bull, S., 2018. A survey on robustness in railway planning. Eur. J. Oper. Res. 266 (1), 1–15. doi:10.1016/j.ejor.2017.07.044.

Lusby, R.M., Larsen, J., Ehrgott, M., Ryan, D., 2011. Railway track allocation: models and methods. OR Spectrum 33 (4), 843–883. doi:10.1007/s00291-009-0189-0.

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

Meisel, F., Fagerholt, K., 2019. Scheduling two-way ship traffic for the Kiel Canal: model, extensions and a matheuristic. Computers and Operations Research 106, 119–132.

Meng, L., Zhou, X., 2011. Robust single-track train dispatching model under a dynamic and stochastic environment: a scenario-based rolling horizon solution approach. Transportation Research Part B: Methodological 45 (7), 1080–1102. doi:10.1016/j.trb.2011.05.001.

Norlund, E.K., Gribkovskaia, I., Laporte, G., 2015. Supply vessel planning under cost, environment and robustness considerations. Omega (Westport) 57, 271–281. doi:10.1016/j.omega.2015.05.006.

Shafia, M.A., Sadjadi, S.J., Jamili, A., Tavakkoli-Moghaddam, R., Pourseyed-Aghaee, M., 2012. The periodicity and robustness in a single-track train scheduling problem. Appl. Soft Comput. 12 (1), 440–452. doi:10.1016/j.asoc.2011.08.026.

Sluiman, F., 2017. Transit vessel scheduling. Nav. Res. Logist. 64 (3), 225–248. doi:10.1002/nav.21742.

UCA, 2019. United Canal Agency: Regulations. https://www.kiel-canal.de/regulations/. [Online; accessed 23-Jan 2020].

Ulusçu, Ö.S., Özbas, B., Altiok, T., Or, I., Yilmaz, T., 2009. Transit vessel scheduling in the strait of istanbul. J. Navig. 62 (1), 59–77.

UNCTAD, 2018. United nations conference on trade and development: Review of Maritime Transport 2018. https://unctad.org/en/pages/PublicationWebflyer.aspx?publicationid=2245. [Online; accessed 23-Jan 2020].

Yang, L., Qi, J., Li, S., Gao, Y., 2016. Collaborative optimization for train scheduling and train stop planning on high-speed railways. Omega (Westport) 64, 57–76. doi:10.1016/j.omega.2015.11.003.

Zhang, C., Gao, Y., Yang, L., Kumar, U., Gao, Z., 2019. Integrated optimization of train scheduling and maintenance planning on high-speed railway corridors. Omega (Westport) 87, 86–104.