Vessel passage scheduling through cascaded bridges using mixed-integer programming

Segovia, Pablo; Negenborn, R.R.; Reppa, Vasso

DOI
10.1016/j.ifacol.2022.09.032

Publication date
2022

Document Version
Final published version

Published in
IFAC-PapersOnline

Citation (APA)
Segovia, P., Negenborn, R. R., & Reppa, V. (2022). Vessel passage scheduling through cascaded bridges using mixed-integer programming. *IFAC-PapersOnline*, *55*(16), 248-253. https://doi.org/10.1016/j.ifacol.2022.09.032

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-16 (2022) 248–253

Vessel passage scheduling through cascaded bridges using mixed-integer programming ⋆

Pablo Segovia * **Rudy R. Negenborn** * **Vasso Reppa** *

* *Department of Maritime and Transport Technology, Delft University of Technology, Delft, the Netherlands (e-mail: {P.SegoviaCastillo, R.R.Negenborn, V.Reppa}@tudelft.nl)*

Abstract: This paper addresses the problem of guaranteeing efficient inland waterway transport in the presence of sequential movable bridges, which must be operated to grant vessel passage. The main contribution is the formulation of the vessel passage scheduling problem as a mixed-integer programming problem. The scheduling algorithm receives vessel widths and voyage plans, and determines feasible vessel passage through bridges that best matches vessel plans. Process optimization is carried out using a rolling horizon implementation to keep the problem computationally tractable in real-time applications. Finally, a realistic case study based on the Rhine-Alpine corridor is used to test the approach and demonstrate its effectiveness.

Keywords: Real-time transportation operations, intelligent transportation systems, scheduling.

1. INTRODUCTION

Inland waterways are large-scale systems consisting of artificial canals and natural rivers, and are mainly used for freight transportation (Segovia et al., 2019). Inland waterway transport (IWT) is one of the most cost-effective and environmentally friendly transport modes to move large amounts of cargo (Ji et al., 2019b), two features that account for its widespread utilization. For instance, the Rhine-Alpine corridor, which connects the North Sea to the Mediterranean basin, handles an annual throughput of more than a billion tonnes of freight (European Commission and Innovation and Networks Executive Agency, 2018). The Three Gorges Dam (Yangtze river, China) also processed over a billion tonnes in 2018 (Zhao et al., 2019).

IWT is a complex system that involves many interacting systems, most notably vessels and infrastructure. This constitutes one of the main differences with respect to overseas transport. Moreover, water level variability in inland waterways poses additional challenges for vessels in terms of navigability and interaction with infrastructure. As a consequence of this complexity, inland shipping has experienced a gradual change of paradigm, from decision making based on human experience to embracing the modern trend of big data era (Yan et al., 2018). On the other hand, recent advancements in communication technologies and computational intelligence have facilitated the advent of *smart* vessels (Du et al., 2021), which can process real-time information and optimize their voyage plans. Therefore, increased levels of digitalization and automation are expected for IWT in the near future (Ye et al., 2021).

The inherent complexity of modern IWT requires to adopt strategies that guarantee an adequate system coordination. In the absence of coordination, navigation delays may arise. Returning to the example of cargo transport along the Yangtze river, Zhao et al. (2019) report that more than two hundred vessels can accumulate at the Three Gorges Dam anchorage during peak time, with an average time to pass the dam of thirty hours. It can be concluded that there is room for operational performance improvement.

One possibility consists in formulating management problems in an optimization framework, as their resolution yields feasible decisions that maximize a profit function. There is a large body of literature on optimization-based IWT management. Lalla-Ruiz et al. (2018) build a mathematical model to assign waterways to incoming and outgoing port vessels to minimize waiting times. Hill et al. (2018) reformulate the previous problem to incorporate time-dependent resource capacities. Chen et al. (2020) schedule passage of vessel train formations through cascaded waterborne intersections.

One of the main bottlenecks in IWT arises from the interaction between vessels and infrastructure, e.g., locks and bridges, as these interrupt navigation frequently. Optimal vessel passage scheduling through infrastructure has thus received considerable attention. Passchyn et al. (2016) tackle the single-chamber serial lock scheduling while addressing the trade-off between sailing time and emissions. Ji et al. (2019a) also study the co-scheduling of cascaded locks, but address the multiple-chamber case. Ji et al. (2021) study the same problem, but their approach has its roots in job shop scheduling and multi-commodity flow.

Although lock operation is a complicated process and can impact vessel navigation, it does not affect other traffic systems. Road and rail traffic make use of available bridges to connect two portions of land that are separated by water flows. Naturally, the operation of movable bridges to

⋆ This research is supported by the project "Novel inland waterway transport concepts for moving freight effectively (NOVIMOVE)". This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 858508.

10.1016/j.ifacol.2022.09.032

enable vessel passage disrupts other traffic, especially given the current state of practice, which prioritizes vessel passage over seamless road and rail traffic (Rijkswaterstaat, Center for Water, Transport and Environment (WVL), 2020). Main operational principles dictate that bridge operators try to fulfill passage requests from vessels as soon as possible, which is granted after making sure that there is no congestion on the bridge deck. Therefore, waterborne, road and rail transport are closely intertwined as a result of bridge operations. Increased infrastructure operation is desirable from the standpoint of IWT, but deteriorates the performance of the other transport modes, aside from resulting in increased wear and tear of infrastructure. On the contrary, limited bridge opening regimes are likely to result in increased waiting times for vessels.

Improved operational performance can be achieved by introducing vessel-to-infrastructure (V2I) communication, which allows vessels to share their voyage plans with bridges in advance. In turn, bridges can process information from multiple vessels, anticipate demand and assign modified arrival times that are best aligned with vessel voyage plans. This approach can be used to tackle vessel passage scheduling through cascaded bridges, whereby vessels are scheduled through a sequence of movable bridges to reach the final destination before the deadline.

The main contribution is the formulation of the vessel passage scheduling problem as a mixed-integer programming problem. The optimization criterion consists in minimizing deviations between vessel voyage plans and assigned bridge passage times by the scheduler, and is formulated as a sum of convex piecewise-linear functions. Moreover, a rolling horizon implementation is proposed to render the approach amenable to real-time implementations.

The remainder is organized as follows: the vessel passage scheduling problem is described in Section 2. Section 3 presents a scheduling approach based on mixed-integer programming. A case study based on the Rhine-Alpine corridor is tested in Section 4, which allows to draw conclusions and outline future research in Section 5.

2. PROBLEM STATEMENT

The vessel passage scheduling problem through cascaded bridges can be formulated as follows. A set of vessels $\mathcal{V} = \{1, ..., N_v\}$ must pass a set of movable bridges $\mathcal{B} = \{1, ..., N_b\}$ on their way from origin to destination. Moreover, $\mathcal{V} = \mathcal{V}_d \cup \mathcal{V}_u$, where $\mathcal{V}_d$ and $\mathcal{V}_u$ denote vessels traveling downstream (from inland towards the sea) and upstream (from the sea towards inland), respectively, and $\mathcal{V}_u \cap \mathcal{V}_d = \emptyset$. A discrete problem setting is adopted, and thus time is divided into a set of time steps $\mathcal{K} = \{1, ..., N\}$ of equal length. Vessel i is characterized by its width w_i [m] and its voyage plans, i.e., earliest and optimal arrival instants at bridge j, $\tau_{ij}^e, \tau_{ij}^o \in \mathbb{Z}_+$, respectively, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}$, with $\mathbb{Z}_+$ the set of positive integers. These arrival instants can be determined considering the maximum speed and the speed that minimizes fuel consumption, respectively. On the other hand, bridge j is described by its passage width b_j [m] and a schedule $s_j \in \mathbb{Z}_+^{n_j}$, i.e., a vector of n_j opening instants during the day, $\forall j \in \mathcal{B}$. Fixed bridge timetables are assumed, and are provided as input data.

The assumptions of the problem are listed below:

- A single waterway is considered, and therefore all vessels $\mathcal{V}$ must pass all bridges $\mathcal{B}$.
- Clearance under bridges (measured from water surface to bridge underside) is not sufficient for vessels to sail below bridges while these are closed.
- Choice of time step size is sufficiently large for vessel i to pass bridge j in one time step with zero dwell time, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}$, and also allows to consider that bridges are either open or closed.

Then, the decision variables are as follows:

- $T_{ij} \in \mathbb{Z}_+$ denotes the integer time instant at which vessel i is scheduled to pass bridge j, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}$.
- $\delta_{ijk} = \{0, 1\}$ is a binary variable that equals 1 if vessel i passes bridge j at time instant k, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}, \forall k \in \mathcal{K}$, and 0 otherwise.

The objective of the vessel passage scheduling problem is to determine T_{ij}, δ_{ijk} so as to satisfy the optimal passing times provided by vessels as much as possible. The design of the problem is tackled in the next section.

3. PROPOSED APPROACH

Vessel passage scheduling can be modeled using ideas from job shop scheduling. Broadly speaking, jobs (vessel passage) must be assigned to machines (bridges) so as to optimize a performance criterion (Ku and Beck, 2016).

A centralized decision entity—the *scheduler*—receives τ_{ij}^e, τ_{ij}^o, w_i, and determines suitable T_{ij}, δ_{ijk}, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}, \forall k \in \mathcal{K}$. It is assumed that all vessels adhere to the scheduler solution using local vessel controllers, which ensure that the scheduler solution is fulfilled. The design of such local controllers is out of the scope of the paper.

The performance criterion is formulated first. Then, physical and operational constraints are defined. Finally, a rolling horizon implementation is considered to account for real-time provision of information.

3.1 Cost function design

The main operational goal consists in minimizing the *scheduling error*, i.e., the mismatch between τ_{ij}^o and T_{ij}, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}$. A cost function J defined as the sum of $N_v \cdot N_b$ convex piecewise-linear functions—one per vessel and bridge—is employed in this paper. This cost function description is well suited to describe real-life conditions, e.g., unequal penalties on earliness and tardiness.

Cost function J is defined as $J \triangleq \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{B}} J_{ij}(T_{ij})$, where $J_{ij}(T_{ij})$ quantifies the penalty corresponding to vessel i passing bridge j, and is formulated as

$$J_{ij}(T_{ij}) = \begin{cases} a_{ij}^{(1)} T_{ij} + b_{ij}^{(1)} & \text{if } T_{ij} \in \left[l_{ij}^{(1)}, u_{ij}^{(1)}\right], \\ a_{ij}^{(2)} T_{ij} + b_{ij}^{(2)} & \text{if } T_{ij} \in \left(u_{ij}^{(1)}, u_{ij}^{(2)}\right], \\ \vdots & \\ a_{ij}^{(n_{ij})} T_{ij} + b_{ij}^{(n_{ij})} & \text{if } T_{ij} \in \left(u_{ij}^{(n_{ij}-1)}, u_{ij}^{(n_{ij})}\right], \end{cases} \quad (1)$$

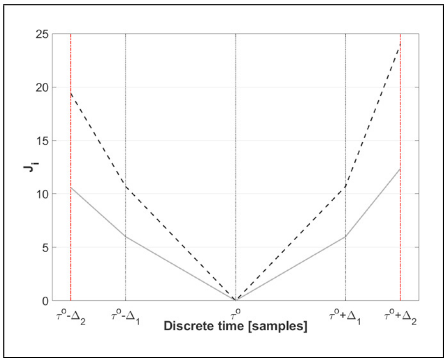

Fig. 1. Piecewise-linear functions for vessels 1 (gray solid) and 2 (black dashed) classes, with $w_1 < w_2$. Vertical black dotted and red solid lines delimit intervals and admissible scheduling interval, respectively.

where $a_{ij}^{(m)}$ and $b_{ij}^{(m)}$ characterize interval m. Moreover, $l_{ij}^{(1)}$ and $u_{ij}^{(m)}$ are the lower and upper bounds of intervals 1 and m, respectively, $\forall m \in \{1, ..., n_{ij}\}$, and n_{ij} is the total number of intervals into which J_{ij} is divided.

The large degree of parametrization can be used to reflect multiple real-life features:

- Larger mismatches between T_{ij} and τ_{ij}^{o} should be increasingly penalized. Then, the farther the interval from τ_{ij}^{o}, the steeper the sub-function slope should be.
- If the exact optimal arrival times cannot be satisfied, scheduling policies favor earliness over tardiness using suitable slope values—although both incur costs.
- While all vessels should be equally treated for the sake of fairness, wide vessels take up more bridge width than narrow vessels. Thus, wider vessels should incur larger scheduling error penalties, e.g., steeper slopes.
- Scheduling error may also be tailored to each bridge based on, e.g., road and/or rail traffic condition on its deck and safety concerns.

These ideas are illustrated by means of Figure 1. Convex piecewise-linear functions are depicted for two vessels of different width with the same optimal arrival times at the same bridge, hence the subscripts i and j have been dropped. Moreover, Δ_1 and Δ_2 are parameters introduced to denote the interval sizes in (1), and have been chosen as vessel- and bridge-independent in this case.

Cost function (1) is then reformulated as follows:

$$J_{ij}(T_{ij}) = \begin{cases} \infty & \text{if } T_{ij} \leq \tau_{ij}^{o} - \Delta_{ij,2}, \\ a_{ij}^{(1)} T_{ij} + b_{ij}^{(1)} & \text{if } \tau_{ij}^{o} - \Delta_{ij,2} < T_{ij} \leq \tau_{ij}^{o} - \Delta_{ij,1}, \\ a_{ij}^{(2)} T_{ij} + b_{ij}^{(2)} & \text{if } \tau_{ij}^{o} - \Delta_{ij,1} < T_{ij} \leq \tau_{ij}^{o}, \\ a_{ij}^{(3)} T_{ij} + b_{ij}^{(3)} & \text{if } \tau_{ij}^{o} < T_{ij} \leq \tau_{ij}^{o} + \Delta_{ij,1}, \\ a_{ij}^{(4)} T_{ij} + b_{ij}^{(4)} & \text{if } \tau_{ij}^{o} + \Delta_{ij,1} < T_{ij} \leq \tau_{ij}^{o} + \Delta_{ij,2}, \\ \infty & \text{if } T_{ij} \geq \tau_{ij}^{o} + \Delta_{ij,2}, \end{cases} \tag{2}$$

and $a_{ij}^{(m)}$ and $b_{ij}^{(m)}$ can be computed using desired slope values and $\Delta_{ij,1}, \Delta_{ij,2}$, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}$, and $m = 1, 2, 3, 4$. Numerical values may be chosen based on costs associated to vessel earliness and tardiness, e.g., mooring costs.

3.2 Formulation of operational constraints

Restrictions to be satisfied during system operation are:

- Vessel passage cannot be scheduled before earliest times of arrival at bridges:
$$T_{ij} \geq \tau_{ij}^{e}, \ \forall i \in \mathcal{V}, \forall j \in \mathcal{B}. \tag{3}$$
- Vessel passage can only take place if bridges are open:
$$T_{ij} \in s_j, \ \forall i \in \mathcal{V}, \forall j \in \mathcal{B}. \tag{4}$$
- Vessels pass each bridge exactly once:
$$\sum_{k \in \mathcal{K}} \delta_{ijk} = 1, \ \forall i \in \mathcal{V}, \forall j \in \mathcal{B}. \tag{5}$$
- Multiple vessels may pass the bridge simultaneously as long as the total width (plus safety distances, included in w_i) do not exceed bridge passage width:
$$\sum_{i \in \mathcal{V}} w_i \delta_{ijk} \leq b_j, \ \forall j \in \mathcal{B}, \forall k \in \mathcal{K}. \tag{6}$$
- Vessels must pass all bridges in the correct order:
$$T_{i(j+1)} \geq T_{ij}, \forall i \in \mathcal{V}_d, \forall j \in \mathcal{B} \setminus \{N_b\}, \tag{7a}$$
$$T_{i(j+1)} \leq T_{ij}, \forall i \in \mathcal{V}_u, \forall j \in \mathcal{B} \setminus \{N_b\}. \tag{7b}$$
Bridge numbering is such that first and last are at the upstream and downstream ends, respectively.
- Passing times of vessels are bounded:
$$T_{ij} \leq N, \ \forall i \in \mathcal{V}, \forall j \in \mathcal{B}. \tag{8}$$
- Decision variables T_{ij} and δ_{ijk} are linked as follows: $(\delta_{ijk} = 1) \rightarrow (T_{ij} = k)$. This can be re-written as
$$\left.\begin{aligned} T_{ij} - k &\leq (1 - \delta_{ijk})\, M \\ k - T_{ij} &\leq (1 - \delta_{ijk})\, M \end{aligned}\right\} \forall i \in \mathcal{V}, \forall j \in \mathcal{B}, \forall k \in \mathcal{K}, \tag{9}$$
where M is a scalar that must be large enough to ensure correctness of (9). Following the reasoning of Ku and Beck (2016), $M \triangleq N_v \cdot N_b \cdot N$ is a suitable constant, as it is infeasible that all vessels are scheduled to pass all bridges at the last time instant.

3.3 Vessel scheduling: a rolling horizon implementation

The vessel passage scheduling problem through cascaded bridges can be formulated as follows:

$$\min_{T_{ij},\, \delta_{ijk}} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{B}} J_{ij}(T_{ij}) \tag{10a}$$

subject to:

$$\text{Constraints (3)–(9)}, \tag{10b}$$

$$T_{ij} \in \mathbb{Z}_{+}, \ \delta_{ijk} = \{0, 1\}, \ \ \forall i \in \mathcal{V}, \forall j \in \mathcal{B}, \forall k \in \mathcal{K}. \tag{10c}$$

Problem (10) can be recast in the mixed-integer programming framework by introducing auxiliary binary and real variables (Bemporad and Morari, 1999). The necessary steps are reported in detail by Ferrari-Trecate et al. (2001). The main steps are sketched below:

- $n_{ij} - 1$ binary variables $\epsilon_{ij}^{(l)}$ are introduced for J_{ij}, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}$, $l = 1, ..., n_{ij} - 1$. Then, $\epsilon_{ij}^{(l)} = 1$ if $T_{ij} > u_{ij}^{(l)}$ following notation in (1), and 0 otherwise.
- $n_{ij} - 1$ variables $z_{ij}^{(l)} \in \mathbb{R}$ are introduced for J_{ij}, and $\tilde{z}_{ij} \triangleq \sum_{l=1}^{n_{ij}-1} z_{ij}^{(l)}$. Each combination of $\epsilon_{ij}^{(l)}$ activates a single $z_{ij}^{(l)}$, thus selecting a different interval of J_{ij}.

Algorithm 1 Vessel passage scheduling through cascaded bridges using a rolling horizon implementation

Require: at time instant k, (i) all data for vessels with $\tau_{ij}^e = k$, $\forall i \in \mathcal{V}$, $\forall j \in \mathcal{B}$, and (ii) bridge schedule s_j and available bridge width b_{jk}, $\forall j \in \mathcal{B}$, $\forall k \in \mathcal{K}$

```
Initialization: b_jk ← b_j, ∀j ∈ B, ∀k ∈ K
for k = 1 : N do
    if τ^e_ij = k, ∀i ∈ V, ∀j ∈ B then
        Solve (11) for vessels s.t. τ^e_ij = k, ∀i ∈ V, ∀j ∈ B
        b_jk ← b_jk − Σ_{i∈V} w_i δ_ijk, ∀j ∈ B, ∀k ∈ K
    else
    end if
end for
```

- Additional constraints must be defined to connect the original decision variables T_{ij}, δ_{ijk} with the auxiliary variables $\epsilon_{ij}^{(l)}$, $z_{ij}^{(l)}$, $\tilde{z}_{ij}$ to reformulate (10) as a mixed-integer programming problem.
- Then, (10) is transformed into the following program:

$$\min_{T_{ij},\, \delta_{ijk},\, \epsilon_{ij}^{(l)},\, z_{ij}^{(l)},\, \tilde{z}_{ij}} \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{B}} \tilde{z}_{ij} \tag{11a}$$

subject to:

$$\text{Constraints (3)–(9)}, \tag{11b}$$

$$\text{Additional constraints connecting } T_{ij},\, \delta_{ijk}, \tag{11c}$$

$$\epsilon_{ij}^{(l)},\ z_{ij}^{(l)},\ \forall i \in \mathcal{V},\ \forall j \in \mathcal{B},\ \forall k \in \mathcal{K},\ l = 1, ..., n_{ij} - 1,$$

$$T_{ij} \in \mathbb{Z}_+,\ \delta_{ijk} = \{0,1\}, \epsilon_{ij}^{(l)} = \{0,1\}\ z_{ij}^{(l)} \in \mathbb{R}, \tag{11d}$$

$$\forall i \in \mathcal{V},\ \forall j \in \mathcal{B},\ \forall k \in \mathcal{K},\ l = 1, ..., n_{ij} - 1.$$

Both (10) and (11) are characterized by a combinatorial nature due to the existence of binary and discrete variables. These problems scale poorly, as the effort to determine optimal solutions increases drastically for increased problem sizes. In this case, the need to schedule the passage of new vessels through the bridges as time progresses may render the problem computationally intractable.

The rolling horizon framework is well suited to tackle the problem, as it uses a sliding window that is shifted in time to solve smaller subproblems that are bounded in size (Zhan et al., 2010). The benefits in the context of this problem are twofold. On the one hand, computational complexity issues can be kept at bay. On the other hand, real-time information can be processed as time progresses. Nevertheless, and while optimal decisions are obtained for each subproblem, the solution of the overall problem might be suboptimal, as the subproblems solved at each time step do not make use of future information (Silvente et al., 2015). Nevertheless, the practical advantages for the vessel passage scheduling problem outweigh the drawbacks.

Algorithm 1 sketches the steps to solve the vessel passage scheduling problem using a rolling horizon implementation. Subproblem resolution at time instant k yields the scheduling of vessels with $\tau_{ij}^e = k$, $\forall i \in \mathcal{V}$, $\forall j \in \mathcal{B}$, $\forall k \in \mathcal{K}$. Furthermore, b_{jk}, which denotes the available width of bridge j at time instant k, is initialized at passage width b_j, $\forall j \in \mathcal{B}$, $\forall k \in \mathcal{K}$. Then, available bridge widths are updated at each time instant after the corresponding solution has been obtained, thus connecting consecutive subproblems.

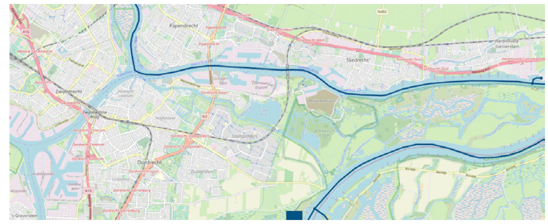

Fig. 2. Schematic representation of the Beneden Merwede

4. CASE STUDY

4.1 System description

The Rhine-Alpine corridor connects major economic centers such as Brussels and Antwerp, the Randstad region, the Rhine-Ruhr and Rhine-Neckar regions, and Milan and Genoa. This is one of the busiest European freight routes, as it connects the ports of Rotterdam and Antwerp to the Mediterranean basin, and its throughput represents 19% of EU's total GDP (European Commission and Innovation and Networks Executive Agency, 2018).

The Beneden Merwede is a river stretch in the Netherlands that runs between Dordrecht and Hardinxveld-Giessendam. A schematic representation obtained using a tool developed by Rijkswaterstaat [1] is provided in Figure 2. Data regarding road and railway movable bridges are provided in Table 1. Moreover, bridge schedules are assumed to be fixed and equal for all bridges in this work.

4.2 Experimental design

A fleet sails from Dordrecht to Hardinxveld-Giessendam, and must pass the movable bridges in Table 1. The following guidelines are observed to generate a fleet:

- All vessel widths must be smaller than the narrowest bridge. Widths must then be selected such that $w_i < \min(b_j)$, safety distances included, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}$.
- The vessel passage scheduling problem is solved for one day. Moreover, bridge opening regimes start at 6 a.m. and finish at 8 p.m., in accordance with Table 1.
- In order for the scheduling problem to be feasible, values for τ_{ij}^e, τ_{ij}^o that enable vessels to be scheduled in the same day must be considered, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}$.

A cost function (2) is defined for vessel i and bridge j, $\forall i \in \mathcal{V}, \forall j \in \mathcal{B}$. As discussed in Section 3.1, $a_{ij}^{(l)}$ and $b_{ij}^{(l)}$ are selected using empirical insight, and in such way that larger penalties are set for wider vessels:

- The following $a_{ij}^{(l)}$ are selected for the smallest vessel, denoted with $\underline{i}$: $a_{\underline{i}j}^{(2)} = 15°$, $a_{\underline{i}j}^{(3)} = a_{\underline{i}j}^{(4)} = 10°$ and $a_{ij}^{(5)} = 20°$. These are increased per additional meter of width for the rest of vessels: $a_{ij}^{(2)} = a_{\underline{i}j}^{(2)} + 3°/\text{m}$, $a_{ij}^{(3)} = a_{ij}^{(4)} = a_{\underline{i}j}^{(3)} + 2°/\text{m}$ and $a_{ij}^{(5)} = a_{\underline{i}j}^{(5)} + 4°/\text{m}$.
- The values for $b_{ij}^{(l)}$ can be computed using $a_{ij}^{(l)}$ and $\Delta_{ij,1}$, $\Delta_{ij,2}$. Dependency of the latter on vessel and bridge is dropped, and the values $\Delta_1 = 2$ hours and $\Delta_2 = 5$ hours are selected.

[1] https://vaarweginformatie.nl/

Table 1. Movable bridges in the Beneden Merwede

Bridge	Width [m]	Assumed fixed schedule	Approx. distance from previous bridge [m]
(1) Traffic bridge Dordrecht	44	6:50-7:00, 7:50-8:00, 8:50-9:00, 9:50-10:00,	—
(2) Railway bridge Grotebrug	44	10:50-11:00, 11:50-12:00, 12:50-13:00,	50
(3) Traffic bridge Papendrecht	30	13:50-14:00, 14:50-15:00, 15:50-16:00, 16:50-17:00,	4500
(4) Railway bridge Baanhoek	30	17:50-18:00, 18:50-19:00, 19:50-20:00	2500

4.3 Results

Based on the above guidelines, a fleet of vessels is generated, and their passing times through the four movable bridges are scheduled using Algorithm 1. Results are obtained in Matlab R2020b using Gurobi Optimization 9.1.2 and YALMIP (Löfberg, 2004). A time step size of five minutes is selected, which imposes an upper bound on the runtime of Algorithm 1. Its selection is deemed appropriate from the standpoint of system operation. Although discrete times are denoted with integers, these values are translated into corresponding five-minute time intervals.

Before proceeding with the analysis of the results, it must be noted that the first and second bridges are treated as a single bridge in practice given the inter-bridge distance reported in Table 1. Thus scheduling results are identical for both bridges., and are provided in the same plots.

Comparison of τ_{ij}^o (optimal vessel voyage plans) and T_{ij} (arrival times assigned by the scheduler) is provided in Figures 3, 4 and 5 for the first-second, third and fourth bridges, respectively. In the absence of a scheduling mechanism, vessels arrive at bridges at their optimal arrival times (depicted in red), which are outside the schedules for the most part. Conversely, the scheduler assigns arrival times (depicted in blue) within bridge opening schedules, thus fulfilling constraint (4). Note that the overlaps that occur when the scheduler can assign passing times T_{ij} (blue) that match optimal voyage plans τ_{ij}^o (red) can be identified in Figures 3–5 by single blue lines for a given vessel.

It must be noted that Figures 3–5 do not provide complete vessel voyage plans, as earliest arrival times are not depicted to avoid cluttered figures. Therefore, while it might appear as if bridges still admit more vessels to pass simultaneously, earliest arrivals might take place later than the closest bridge opening slot. An example of this is vessel 5 in Figure 5, whose optimal arrival time is closer to the 9:50–10:00 slot, but is actually assigned to pass at 10:50. In these cases, the scheduler assigns values of T_{ij} that belong to a future opening slot (with respect to τ_{ij}^e and τ_{ij}^o) to ensure satisfaction of constraint (3).

It was discussed in Section 3.3 that problem resolution using a rolling horizon implementation might not yield optimal schedules. This can be realized by considering vessels 47 and 53 in Figure 4. While the optimal arrival time of vessel 47 is five minutes earlier than that of vessel 53, the latter is scheduled to pass before the former. Schedules are built in an incremental manner, and the rolling horizon scheduling strategy operates through provision of earliest arrival times. In this case, $\tau_{53,j}^e < \tau_{47,j}^e$, which is why it is scheduled to pass before. However, a rolling horizon implementation allows for real-time scheduling.

Bridge width occupancy considering the scheduler solution is provided in Figures 6, 7 and 8 for the first-second, third and fourth bridges, respectively. Green and blue

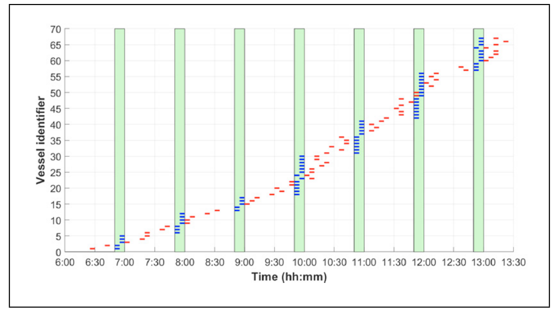

Fig. 3. Comparison of τ_{ij}^o (red) and T_{ij} (blue), first and second bridges

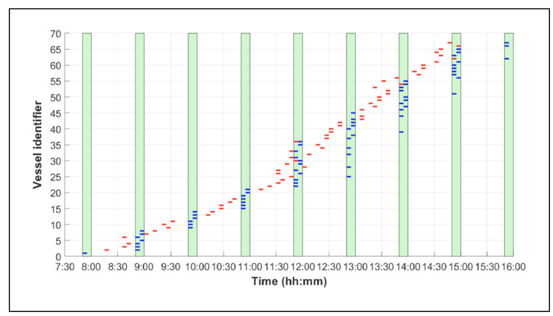

Fig. 4. Comparison of τ_{ij}^o (red) and T_{ij} (blue), third bridge

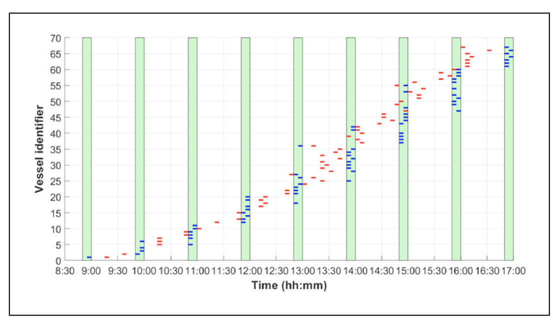

Fig. 5. Comparison of τ_{ij}^o (red) and T_{ij} (blue), fourth bridge

bars represent the maximum passage width b_j during opening windows and total bridge width usage by vessels, respectively. The scheduler succeeds in assigning feasible passing times from the standpoint of resource availability, thus fulfilling constraint (6). Moreover, smaller width of third and fourth bridges (with respect to first and second bridges) accounts for a larger resource usage by the former.

5. CONCLUSIONS AND FUTURE RESEARCH

This paper presented a formulation of the vessel passage scheduling problem through cascaded bridges using mixed-integer programming. Vessel passage was scheduled to minimize the mismatch with respect to optimal vessel voyage plans. To this end, vessel dimensions, and earliest and optimal arrival times were provided to the scheduler, which

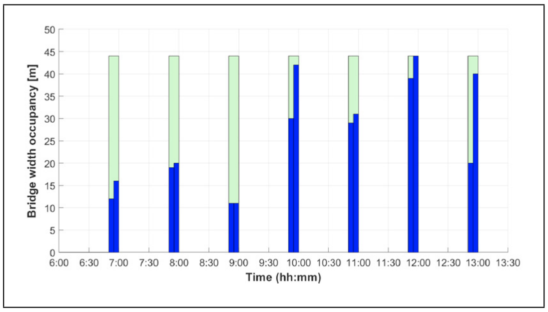

Fig. 6. Width occupancy of first and second bridges

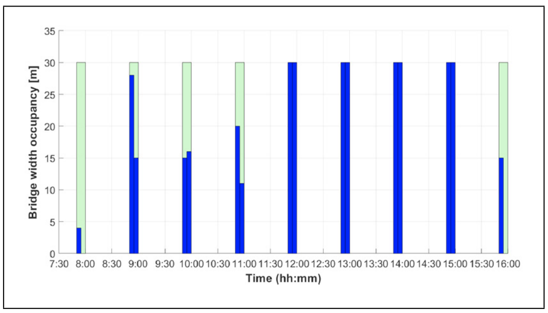

Fig. 7. Width occupancy of third bridge

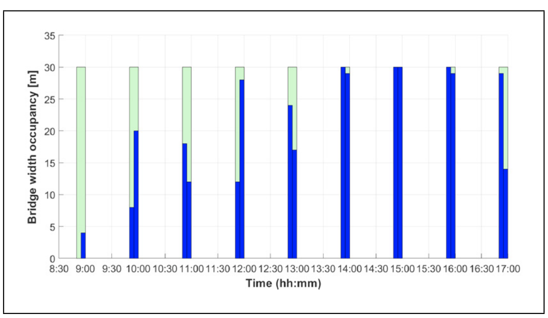

Fig. 8. Width occupancy of fourth bridge

processes them and sends an accepted or a modified arrival time back to each vessel. The optimality criterion was formulated as a sum of convex piecewise-linear functions.

Results reveal that the scheduler is constrained by given arrival times and bridge schedules. Extension to optimized bridge opening regimes will be examined to adapt bridge operation to navigation demand. Future work also regards the design of vessel controllers that guarantee arrival times at bridges in compliance with the scheduling solution.

REFERENCES

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

Chen, L., Huang, Y., Zheng, H., Hopman, H., and Negenborn, R. (2020). Cooperative multi-vessel systems in urban waterway networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(8), 3294–3307.

Du, Z., Negenborn, R.R., and Reppa, V. (2021). Cooperative Multi-Agent Control for Autonomous Ship Towing Under Environmental Disturbances. *IEEE/CAA Journal of Automatica Sinica*, 8(8), 1365–1379.

European Commission and Innovation and Networks Executive Agency (2018). *CEF support to Rhine - Alpine Corridor*. Publications Office. doi:doi/10.2840/480417.

Ferrari-Trecate, G., Letizia, P., and Spedicato, M. (2001). Optimization with piecewise-affine cost functions. Technical report, Citeseer.

Hill, A., Lalla-Ruiz, E., Voß, S., and Goycoolea, M. (2018). A multi-mode resource-constrained project scheduling reformulation for the waterway ship scheduling problem. *Journal of Scheduling*, 22(2), 173–182.

Ji, B., Yuan, X., and Yuan, Y. (2019a). A hybrid intelligent approach for co-scheduling of cascaded locks with multiple chambers. *IEEE Transactions on Cybernetics*, 49(4), 1236–1248.

Ji, B., Yuan, X., Yuan, Y., Lei, X., Fernando, T., and Iu, H.H. (2019b). Exact and heuristic methods for optimizing lock-quay system in inland waterway. *European Journal of Operational Research*, 277(2), 740–755.

Ji, B., Zhang, D., Yu, S.S., and Kang, C. (2021). Mathematical programming models for scheduling multiple cascaded waterway locks. *Computers & Industrial Engineering*, 156, 107289.

Ku, W. and Beck, J. (2016). Mixed integer programming models for job shop scheduling: A computational analysis. *Computers & Operations Research*, 73, 165–173.

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

Löfberg, J. (2004). YALMIP: a toolbox for modeling and optimization in MATLAB. In *IEEE International Symposium on Computer Aided Control Systems Design*.

Passchyn, W., Briskorn, D., and Spieksma, F.C. (2016). Mathematical programming models for lock scheduling with an emission objective. *European Journal of Operational Research*, 248(3), 802–814.

Rijkswaterstaat, Center for Water, Transport and Environment (WVL) (2020). Waterway guidelines 2020. Technical report, Rijkswaterstaat. URL `https://puc.overheid.nl/rijkswaterstaat/doc/PUC_632307_31/`.

Segovia, P., Rajaoarisoa, L., Nejjari, F., Duviella, E., and Puig, V. (2019). Model predictive control and moving horizon estimation for water level regulation in inland waterways. *Journal of Process Control*, 76, 1–14.

Silvente, J., Kopanos, G., Pistikopoulos, E., and Espuña, A. (2015). A rolling horizon optimization framework for the simultaneous energy supply and demand planning in microgrids. *Applied Energy*, 155, 485–501.

Yan, X., Wang, K., Yuan, Y., Jiang, X., and Negenborn, R.R. (2018). Energy-efficient shipping: An application of big data analysis for optimizing engine speed of inland ships considering multiple environmental factors. *Ocean Engineering*, 169, 457–468.

Ye, J., Reppa, V., Godjevac, M., and Negenborn, R.R. (2021). Construction mode detection for autonomous offshore heavy lift operations. *Safety Science*, 133, 104991.

Zhan, Z.H., Zhang, J., Li, Y., Liu, O., Kwok, S.K., Ip, W.H., and Kaynak, O. (2010). An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem. *IEEE Transactions on Intelligent Transportation Systems*, 11(2), 399–412.

Zhao, X., Lin, Q., and Yu, H. (2019). An Improved Mathematical Model for Green Lock Scheduling Problem of the Three Gorges Dam. *Sustainability*, 11(9).