Multi-vessel placement in multi-chamber inland waterway transport locks using switching max-plus algebra

Segovia, Pablo; Ummels, Raphael; van den Boom, Ton; Reppa, Vasso

DOI
10.1016/j.oceaneng.2025.123414

Publication date
2026

Document Version
Final published version

Published in
Ocean Engineering

Ocean Engineering 343 (2026) 123414

Contents lists available at ScienceDirect

Ocean Engineering

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

Research paper

Multi-vessel placement in multi-chamber inland waterway transport locks using switching max-plus algebra

Pablo Segovia [a,b,*], Raphael Ummels [c,d], Ton van den Boom [d], Vasso Reppa [c]

[a] Department of Automatic Control, Universitat Politècnica de Catalunya - BarcelonaTECH, Barcelona, Spain
[b] Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
[c] Department of Maritime and Transport Technology, Delft University of Technology, Delft, The Netherlands
[d] Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Keywords:
Inland waterborne transport
Intelligent transportation systems
Lock scheduling
Multi-chamber locks
Multi-vessel chambers
Max-plus algebra
Switching max-plus linear systems.

ABSTRACT

This paper presents a novel scheduling approach for inland waterway transport (IWT) vessels that must pass through multi-chamber locks. A *switching max-plus linear* (SMPL) model is built to determine, for each vessel, the most appropriate route, the arrival times at relevant waypoints as well as at the destination, the relative order in which it moves through the network with respect to other vessels, its assignment to certain lock chambers together with other vessels, and its position inside each chamber. The SMPL constraints are translated to mixed integer linear programming (MILP) constraints for the optimization problem to be solvable, and objectives minimizing arrival times or arrival time offsets are defined. The proposed approach is tested on a multi-lock waterway, and its performance is compared to the current state of practice using relevant key performance indicators (KPIs), which allows to demonstrate the superior performance of the proposed approach.

1. Introduction

Inland waterborne transport (IWT) is a low CO_2 emission alternative to road transport (Agency, 2021), leading to reduced noise pollution and relieved road and railway congestion (European Commission, 2001). Moreover, marine vessels are expected to lower carbon intensity and greenhouse gas emissions in the near future to comply with IMO regulations (Kougiatsos et al., 2025). However, Dutch and European IWT have never been exploited to their full capacity, only making up a small percentage of the EU modal share despite repeated objectives set for growth (Eurostat, 2022). Deficient infrastructure, particularly locks, has been identified as a factor holding back that growth (Soveges et al., 2015). Locks are ship-passable divisions between different water levels on a waterway, and consist of one or multiple chambers in which the water level can be regulated, so vessels can be brought from one water level to another (Liu et al., 2021) by means of a *lockage* . These chambers have a limited capacity and their operation takes time, potentially causing delays for vessels (Passchyn et al., 2016b). The same holds true about vessels using infrastructure other than locks, e.g., movable bridges (Segovia et al., 2025), dams (Liu et al., 2025), quays (Zhang et al., 2024) and ports (Dong et al., 2025).

Expensive and time-consuming physical expansion of locks is not always an option. Moreover, projects that get executed often miss their targets (Soveges et al., 2015). Another possibility to improve vessel navigation is to optimize their passage through the locks. Optimization of lock operations, also referred to as lock scheduling, is an important and well studied problem. Table 1 summarizes existing approaches to solve the lock scheduling problem for a wide variety of settings. The entries are arranged considering that the lock scheduling problem consists of three interrelated subproblems: chamber assignment, lockage operation scheduling, and ship placement (Verstichel et al., 2014a). However, as some of the approaches do not address the multi-chamber lock case, the chamber assignment and the lockage operation scheduling subproblems are considered in a single group for classification purposes.

Existing lock scheduling approaches are first examined from the standpoint of the inland waterway network topology. In the context of the IWT problem, network topology can be viewed as the manner in which locks are connected to one another within the waterway system. Hence, there exists a direct connection between network topology and the number of locks that are scheduled. Ji et al. (2020a, 2019), Zhang et al. (2023), Hermans (2014), Verstichel et al. (2011), Ji et al. (2020b), Verstichel et al. (2015) schedule vessel passage through a single lock,

* Corresponding author.
E-mail addresses: pablo.segovia@upc.edu (P. Segovia), raphael.ummels@upcmail.nl (R. Ummels), a.j.j.vandenBoom@tudelft.nl (T. van den Boom), v.reppa@tudelft.nl (V. Reppa).

https://doi.org/10.1016/j.oceaneng.2025.123414
Received 21 June 2025; Received in revised form 11 October 2025; Accepted 1 November 2025
Available online 17 November 2025
0029-8018/

Table 1
Classification of existing lock scheduling solutions.

Reference	Chamber assignment + lockage operation scheduling					Ship placement problem		Method**
	Network topology*	Number of locks	Navigation direction	Number of parallel chambers	Number of vessels per chamber	Chamber dimension	Placement rules	
Ji et al. (2020a)	N/A	1	Unidirectional	1	> 1	2	2D bin packing	MILP + NSGA-II
Ji et al. (2019)	N/A	1	Unidirectional	1	> 1	2	2D bin packing	MILP + BB
Zhang et al. (2023)	N/A	1	Unidirectional	1	> 1	2	2D bin packing	MOMA + FCE
Hermans (2014)	N/A	1	Bidirectional	1	1	N/A	—	MILP + BS + FS
Verstichel et al. (2011)	N/A	1	Bidirectional	> 1	> 1	1	Simplified capacity (width)	MILP + SRMH
Ji et al. (2020b)	N/A	1	Bidirectional	> 1	> 1	2	2D bin packing + mooring	MILP + ALNS
Verstichel et al. (2015)	N/A	1	Bidirectional	> 1	> 1	2	2D bin packing + mooring	MILP + BD
Wang et al. (2013)	Linear	> 1	Unidirectional	1	> 1	1	Simplified capacity (area)	MILP + ACO
Yuan et al. (2016)	Linear	> 1	Unidirectional	> 1	> 1	2	2D bin packing	MILP + ABC
Passchyn et al. (2016a)	Linear	> 1	Bidirectional	1	> 1	1	Simplified capacity (vessels)	MILP (CPLEX)
Zhao et al. (2020)	Linear	> 1	Bidirectional	1	> 1	2	2D bin packing	MILP + BQIGSA + MOBFTS
Prandtstetter et al. (2015)	Linear	> 1	Bidirectional	> 1	> 1	1	Simplified capacity (length)	FF + VNS
Ji et al. (2021a)	Linear	> 1	Bidirectional	> 1	> 1	2	2D bin packing + mooring	MILP (Gurobi)
Ji et al. (2021b)	Linear	> 1	Bidirectional	> 1	> 1	2	2D bin packing + mooring	MILP (Gurobi)
Segovia et al. (2022)	SOSD	> 1	Bidirectional	1	1	N/A	—	SMPL (Gurobi)
Ji et al. (2017)	MOMD	> 1	Bidirectional	1	> 1	2	2D bin packing	MINLP + ONSGA-III
This paper	**Arbitrary**	**> 1**	**Bidirectional**	**> 1**	**> 1**	**2**	**2D bin packing + mooring**	**SMPL (Gurobi)**

* SOSD: Single-Origin, Single-Destination; MOMD: Multiple-Origin, Multiple-Destination.
** MILP: Mixed Integer Linear Programming; NSGA-II: Nondominated Sorting Genetic Algorithm II; BB: Binary Borg; BS: Backscheduling; FS: Forward Scheduling; SRMH: Simple Random Meta Heuristic; ALNS: Adaptive Large Neighbourhood Search; ACO: Ant Colony Optimization; ABC: Artificial Bee Colony; BQIGSA: Binary Quantum-Inspired Gravitational Search Algorithm; MOBFTS: Multi-Order Best-Fit Tabu Search; FF: First-Fit; VNS: Variable Neighbourhood Search; BD: Benders' Decomposition; SMPL: Switching Max-Plus Linear; B&B: Branch and Bound; MINLP: Mixed Integer Nonlinear Programming; ONSGA-III: Orthogonal Nondominated Sorting Genetic Algorithm.

hence the concept of network topology is not applicable. The linear case, in which locks are connected in a linear sequence, is most commonly studied in the literature. Wang et al. (2013), Yuan et al. (2016), Passchyn et al. (2016a), Zhao et al. (2020), Prandtstetter et al. (2015), Ji et al. (2021a,b) all consider waterways with cascaded locks, meaning that all locks must be passed, one after the other, from origin to destination. Two exceptions to the linear topology case can be found. On the one hand, Segovia et al. (2022) address the single-origin, single-destination case with routing capabilities . All vessels travel between two endpoints that are connected by more than one waterway, and the route that each vessel follows is a decision variable that allows to optimize travel time. On the other hand, Ji et al. (2017) develop an ad hoc solution for the multiple-origin, multiple-destination case considering a small number of endpoints and routes. Differently from Segovia et al. (2022), the route is fixed for each vessel and hence cannot be decided.

Inland waterways can generally be sailed in both directions, i.e., upstream and downstream, although unidirectional navigation has also been examined in Ji et al. (2020a, 2019), Zhang et al. (2023), Wang et al. (2013), Yuan et al. (2016). This setting requires an additional empty lockage after every lockage to restore the water level inside the chamber, so that the next group of vessels can enter the lock. This is not necessarily the case in bidirectional navigation, as upstream and downstream lockages may be scheduled alternately, eliminating the requirement to perform empty lockages.

An important feature of locks is the number of parallel chambers. The single-chamber lock case, which has been studied in Ji et al. (2020a, 2019), Zhang et al. (2023), Hermans (2014), Wang et al. (2013), Passchyn et al. (2016a), Zhao et al. (2020), Segovia et al. (2022), Ji et al. (2017), does not require to solve the chamber assignment problem, as there exists only one chamber vessels can be assigned to. On the other hand, the multi-chamber lock case, which can be modeled as a parallel machine scheduling problem (Verstichel et al., 2014a) and is addressed in Verstichel et al. (2011), Ji et al. (2020b), Yuan et al. (2016), Prandtstetter et al. (2015), Verstichel et al. (2015), Ji et al. (2021a,b), allows to assign vessels to different chambers (although limitations may apply to, e.g., vessels with special cargo). While this setting provides more flexibility than the single-chamber lock case (when the chambers can be operated independently), this benefit comes at the cost of an increased number of decision variables.

Lock chambers can generally fit multiple vessels in the same lockage, although this depends on the specific size of the vessels and how these are arranged inside the chambers (Lan et al., 2024). However, Hermans (2014) and Segovia et al. (2022) simplify the problem by considering that only one vessel can be transferred in a lockage, hence the ship placement problem need not be addressed. On the other hand, the multi-vessel case requires to solve the ship placement problem, whereby the ships that are transferred in the same lockage and their position inside the chamber must be determined (Verstichel et al., 2014b). While this requires to observe chamber dimensions, there exist approaches that encode this information in the form of a one-dimensional parameter. Verstichel et al. (2011) use chamber width; Prandtstetter et al. (2015), chamber length; Wang et al. (2013), chamber area; and Passchyn et al. (2016a) consider that a maximum number of vessels can be placed in the same lockage, hence disregarding the dimensions of both the vessels and the chambers. The approaches presented in Ji et al. (2020a, 2019), Zhang et al. (2023), Ji et al. (2020b), Yuan et al. (2016), Zhao et al. (2020), Verstichel et al. (2015), Ji et al. (2021a,b, 2017) address the ship placement problem explicitly, which is often compared to the 2D bin packing problem in that a set of vessels that cannot rotate or overlap must be placed inside as few lock chambers as possible (Verstichel et al., 2014b). However, the mooring subproblem, whose resolution determines whether vessels are moored to the sides of the chambers or to other vessels, is only solved in Verstichel et al. (2015), Ji et al. (2020b, 2021a,b). Although the explicit consideration of this subproblem renders the solution more interesting from a practical standpoint, the reduced number of references that include this feature is probably due to the additional number of decision variables and constraints that are created, and whose presence complicates problem resolution even further.

Lock scheduling problems are usually formulated using different variations of mixed-integer programming (MIP). Mixed-integer linear programming (MILP) stands out as prevalent, although Ji et al. (2017) employ mixed-integer nonlinear programming (MINLP). The reason behind the success of MIP is mainly due to its modeling flexibility and the existence of state-of-the art linear programming (LP) based solvers (Vielma, 2015). Prandtstetter et al. (2015) take a different course and use the first-fit algorithm to generate an initial feasible solution by assigning vessels to lockages in chronological order. These problems are then generally solved approximately given their large dimensions, using metaheuristic algorithms to guide the search process. Another common option is to solve the problem directly using dedicated optimization software such as Gurobi (Ji et al., 2021b; Segovia et al., 2022; Ji et al., 2017) and CPLEX (Passchyn et al., 2016a), which, like the rest of state-of-the-art MIP solvers, are based on the branch-and-bound algorithm (Vielma, 2015). Verstichel et al. (2015) use Benders' decomposition to divide the problem into smaller instances to facilitate its resolution.

A more recent research line to schedule the operation of IWT vessels and locks was proposed in Segovia et al. (2022), where the system was modeled in the switching max-plus linear (SMPL) framework. Max-plus algebra lends itself well to the scheduling of event-based systems (Cohen et al., 1999). Moreover, the use of max-plus linear (MPL) systems as basic models for scheduling offers several advantages, such as the existence of several system-theoretical results, notably max-plus eigenvalues and eigenvectors, the growth-rate of the SMPL system, and controllability of the SMPL system in the case of a controlled system (Cohen et al., 1999; Heidergott et al., 2014; van den Boom et al., 2020), and the possibility to find bottlenecks and adequate initial scheduling values (Kersbergen et al., 2014). Moreover, SMPL systems are endowed with the capability to switch between different modes, a feature that aligns well with the features of IWT, as vessels can choose among different routes while overtaking other vessels. As such, SMPL systems offer more versatility than conventional MPL systems, which are characterized by a fixed model structure.

It can be concluded that the literature on lock scheduling is rich. Different approaches have been proposed, each focusing on some important aspects of the problem. Direct MILP algorithms offer high optimization quality but struggle with scalability and real-time responsiveness. Job-shop scheduling is flexible but can be complex to model and solve. Rule-based dispatching is fast and intuitive but often leads to suboptimal network-wide outcomes. Scheduling using SMPL models excels in dynamic, delay-prone environments where real-time rescheduling is critical. The algebraic structure and switching capabilities of SMPL make it particularly well-suited for modern transportation networks that demand both robustness and adaptability. All these arguments make SMPL a compelling choice for infrastructure or large-scale systems prone to frequent disruptions (Garrisi and Cervelló-Pastor, 2020; Kersbergen, 2015; van den Boom et al., 2020).

The main objective of this paper is the novel design of a general lock scheduling approach under realistic operational conditions, building on the advantages offered by SMPL models. More precisely, this paper presents the design of an SMPL system to schedule vessel passage through multi-chamber locks for any arbitrary IWT network, each chamber being able to fit multiple vessels. The main contribution of the paper is twofold. First, the proposed design can be applied to any arbitrary IWT layout, without any restrictions on the number of origins and destinations of the vessels. Second, by incorporating realistic two-dimensional ship placement capabilities using the SPP rules outlined in Verstichel et al. (2011), a ship placement solution which borrows ideas from the Tetris model presented in Bartels et al. (2021) is proposed to manage the passage of multiple vessels assigned to multiple chambers.

The remainder of this paper is organized as follows. Relevant max-plus algebra definitions and concepts are presented in Section 2. Section 3 describes the IWT problem in the presence of locks, and a solution to address the problem is presented in Sections 4 and 5. Section 6 introduces the case study and the tests that are used to validate the approaches, and the results obtained are discussed on the basis of key performance indicators (KPIs). Conclusions and future research directions are given in Section 7.

2. Preliminaries

Max-plus algebra is an algebra where the $+$ and $\times$ operators are replaced with the maximization operator $\oplus$ and the plus operator $\otimes$, respectively (Heidergott et al., 2014). The unit element is represented by $e = 0$ and the zero element is represented by $\varepsilon = -\infty$. The max-plus space of real numbers is defined as $\mathbb{R}_{\max} \triangleq \mathbb{R} \cup \{\varepsilon\}$. Let $a, b, c \in \mathbb{R}_{\max}$. Then:

$$c = a \oplus b = \max(a, b), \tag{1a}$$

$$c = a \otimes b = a + b, \tag{1b}$$

$$c = a \oplus \varepsilon = a. \tag{1c}$$

Matrix operators in max-plus algebra are equivalent to those in conventional algebra, with the operations replaced with their max-plus counterparts. Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}_{\max}^{n\times m}$, $\boldsymbol{C} \in \mathbb{R}_{\max}^{m\times p}$, and $[\boldsymbol{A}]_{ij}$ be entry a_{ij} of $\boldsymbol{A}$. Then:

$$[\boldsymbol{A} \oplus \boldsymbol{B}]_{ij} = \max(a_{ij}, b_{ij}),\ \forall i \in \mathcal{N},\ \forall j \in \mathcal{M}, \tag{2a}$$

$$[\boldsymbol{A} \otimes \boldsymbol{C}]_{il} = \bigoplus_{j=1}^{m} a_{ij} \otimes c_{jl} = \max_{j\in\mathcal{M}}(a_{ij} + c_{jl}),\ \forall i \in \mathcal{N},\ \forall l \in \mathcal{P},$$

with $\mathcal{N} = \{1, 2, \ldots, n\}$, $\mathcal{M} = \{1, 2, \ldots, m\}$ and $\mathcal{P} = \{1, 2, \ldots, p\}$.

These operations are used in MPL systems to model the behavior of discrete-event systems (DES). A general MPL system is defined by

$$\boldsymbol{x}(k) = \bigoplus_{\mu=\mu_{min}}^{\mu_{max}} \boldsymbol{A} \otimes \boldsymbol{x}(k-\mu) \oplus \boldsymbol{B} \otimes \boldsymbol{r}(k), \tag{3}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^n$ and $\boldsymbol{r}(k) \in \mathbb{R}^m$ are the state and input vectors at event k, respectively, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the state and input matrices at event k, respectively, and $\mu_{\min}$ and $\mu_{\max}$ denote the minimum and maximum value of μ, respectively.

Let v and $\bar{v}$ be a max-plus binary variable and its adjoint, respectively. Then, the following definitions apply:

$$v = \begin{cases} e & \text{(true)} \\ \varepsilon & \text{(false)} \end{cases} \rightarrow \bar{v} = \begin{cases} \varepsilon & (v = e) \\ e & (v = \varepsilon) \end{cases}. \tag{4}$$

3. Problem statement

The problem considered in this paper consists in determining vessel voyage plans from origin (departure node) to destination (destination node), all being required to pass through locks during their trips. These plans include, for each vessel, the selection of the most appropriate route, its arrival times at relevant waypoints along the trip as well as the destination, the relative order in which the vessel moves through the network with respect to other vessels (due to, e.g., overtaking manoeuvres), its assignment to certain lock chambers, and its position inside each chamber. Moreover, the problem is characterized by the existence of routing, ordering, synchronization, placement and mooring constraints, which further complicate its resolution.

In view of the above, the problem can be formally stated as follows. An **IWT network** is described by a topology graph consisting of a set of nodes $\mathcal{N} = \{0, 1, \ldots, n_{nodes} - 1\}$, which represent relevant physical locations within the waterway (e.g., possible origins and destinations, and lock waiting areas and chambers), and a set of arcs $\mathcal{D}$, which represent waterways connecting the nodes, with an arc from node i to node j, denoted by (i, j). Note that the existence of (i, j) implies the existence of (j, i), as bidirectional traffic is explicitly considered. Node i is given an x and y location, with position parameters $p_{x,i}$ and $p_{y,i}$. The travel distance between nodes i and j, denoted by $d(i, j)$, is taken to be the Euclidean distance.

A set of **vessels** $\mathcal{K} = \{0, 1, \ldots, n_{vessels} - 1\}$ travel on the network, such that departure and destination nodes, denoted by $b(k)$ and $d(k)$, respectively, are any two graph nodes, which are assumed to be always connected. Vessel k is assumed to maintain a constant speed on arc (i, j) while possibly overtaking other vessels. Moreover, vessel k has a known minimum departure time $u(k)$, an optional scheduled arrival time $\hat{x}_d(k)$ (not all arrivals of vessels at their destination may be scheduled), and a known and achievable maximum velocity $v_{max,(i,j)}(k)$. This maximum velocity results in a minimum travel time $\tau_{(i,j)}(k)$ that can be computed

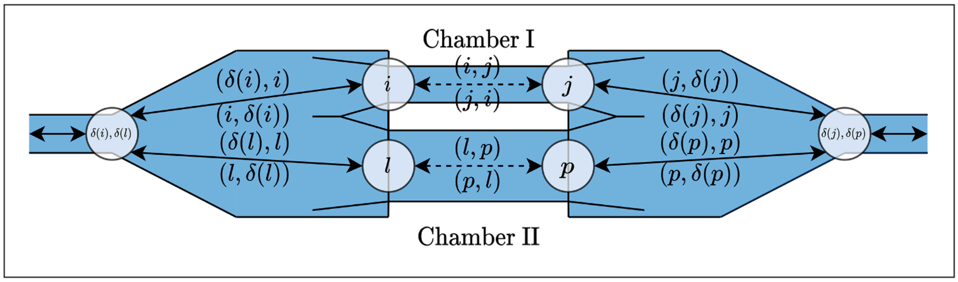

Fig. 1. Schematic representation of a multi-chamber lock with two chambers.

as

$$\tau_{(i,j)}(k) = \frac{d(i,j)}{v_{max,(i,j)}(k)}. \tag{5}$$

These vessels must pass through a set of **locks** $\mathcal{L} = \{0, 1, \ldots, n_{locks} - 1\}$ while sailing from departure to destination node. Locks are represented using a special construction of nodes and arcs, such that lock chambers are defined by two nodes i and j connected by arc (i,j), with $(i,j) \in \mathcal{D}_{locks} \subset \mathcal{D}$. Lock chamber arc $(i,j) \in \mathcal{D}_{locks}$ is characterized by a minimum operation time, which is denoted by $\tau_{(i,j)}(k)$. This minimum operation time corresponds to the minimum amount of time needed to perform a lockage operation of that chamber, and includes the chamber filling and emptying times. Moreover, for the case of non-empty lockages, it also includes the time required to position and moor a vessel before the operation, and the time required for the vessel to exit the chamber after the operation. Note that $\tau_{(i,j)}(k)$ can be generally defined as the minimum time for vessel k to travel from node i to j on arc (i,j), hence this definition is in agreement with the regular arc travel time defined by (5). Moreover, let $\overline{\tau}_{(i,j)}$ be the extra vessel processing time of chamber (i,j), which is assumed to be independent of the extra vessel itself and the number of vessels already in the chamber. To clarify this, consider the following situation: if a lockage on lock chamber arc (i,j) has three vessels in it, its operation time is then at worst $\tau_{(i,j)}(k) + 2\overline{\tau}_{(i,j)}$, with vessel k being one of the vessels in the lockage. Chamber (i,j) is discretized into $N_{\chi,(i,j)}$ equally sized horizontal bins, with $N_{\gamma,(i,j)}$ [m] its vertical length. Moreover, vessel shapes are approximated by bounding rectangles, such that $W_{\chi}(k)$ is the width of vessel k in number of discrete bins and $W_{\gamma}(k)$ [m] is the length of vessel k. Maximum lock chamber capacity must be observed, which can be achieved by constraining how vessels are stacked both in height and width.

A schematic representation of a multi-chamber lock with two chambers is shown in Fig. 1. Note that all lock chambers share the same two waiting areas, which are assumed to have infinite capacity, hence $\delta(i) = \delta(l)$ and $\delta(j) = \delta(p)$. While separate chambers may have separate waiting areas for mooring in reality, the entry point to the run-out zone—and thus the waiting area entry node—is assumed to be the same for all lock chambers. Fig. 2 provides a schematic representation of the geometrical parameters of lock chamber (i,j) and vessel k.

Lockages are complex operations in practice, and for this reason the following considerations are introduced in this paper, following the guidelines established by the Government (2017). Lockages can start as soon as all vessels assigned to the same chamber have entered the lock in a synchronized manner, and the duration of a lockage is assumed to depend on the number of vessels inside the chamber. Vessel $k \in \mathcal{K}$ must always be moored to the sides of the chamber or another vessel $k - \mu \in \mathcal{K} \backslash \{k\}$, as long as vessel k can be fully alongside vessel $k - \mu$. Special rules apply to vessels transporting flammable goods, denoted by $\mathcal{K}_{\nabla} \subset \mathcal{K}$, and explosive and/or toxic goods, denoted by $\mathcal{K}_{\nabla\nabla} \subset \mathcal{K}$. On the one hand, vessel $k \in \mathcal{K}_{\nabla}$ can be synchronized with other vessels in the same lockage, but is required to keep at least 10 m of distance to other vessels (this distance is included in the bounding rectangle sizes used for placement), hence vessel $k \in \mathcal{K}_{\nabla}$ cannot be moored to any other vessel. On the other hand, vessel $k \in \mathcal{K}_{\nabla\nabla}$ cannot be synchronized with other vessels in the same lockage. Fig. 3 shows a number of acceptable and unacceptable configurations. Vessels moored to another vessel are indicated by a black arrow. Note that no restriction for vessels $k \in \mathcal{K}_{\nabla} \cup \mathcal{K}_{\nabla\nabla}$ on arcs $(i,j) \in \mathcal{D} \backslash \mathcal{D}_{locks}$ is considered in this paper.

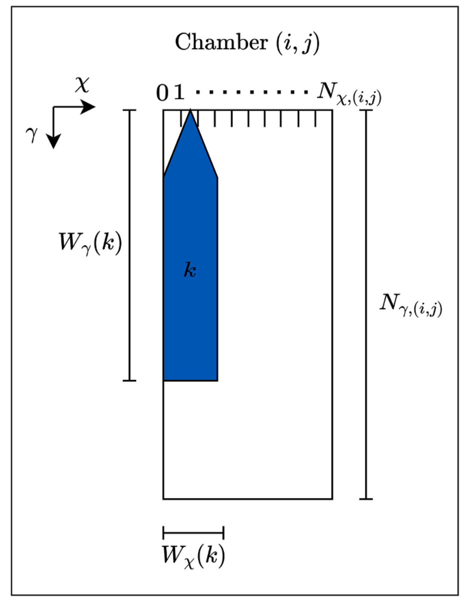

Fig. 2. Geometrical parameters of lock chamber (i,j) and vessel k.

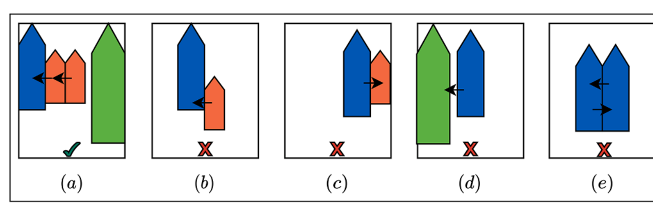

Fig. 3. Vessel mooring configurations. (a) Acceptable configuration. (b) The smaller vessel is not fully alongside the larger vessel, and the larger vessel is not moored to anything. (c) The larger vessel cannot be fully alongside the smaller vessel. (d) The vessels are not adjacent. (e) The vessels are both fully adjacent to each other, but neither is moored to the chamber side.

Finding a solution for the IWT problem considering locks requires to determine the values of the decision variables:

- $x_i(k)$ denotes the arrival time of vessel k at node i, $\forall k \in \mathcal{K}$, $\forall i \in \mathcal{N}$.
- $w_{(i,j)}(k)$ represents the routing of vessel k through arc (i,j), $\forall k \in \mathcal{K}$, $\forall (i,j) \in \mathcal{D}$, such that

$$w_{(i,j)}(k) = \begin{cases} e & \text{if vessel } k \text{ travels along arc } (i,j), \\ \varepsilon & \text{otherwise.} \end{cases} \tag{6}$$

- $z_{(i,j)}(k, k-\mu)$ represents the ordering of vessels k and $k-\mu$ in lock chamber arc (i,j), $\forall k \in \mathcal{K}$, $\forall k-\mu \in \mathcal{K}\backslash\{k\}$, $\forall (i,j) \in \mathcal{D}_{locks}$, such that

$$z_{(i,j)}(k, k-\mu) = \begin{cases} e & \text{if vessel } k-\mu \text{ uses arc } (i,j) \text{ before vessel } k, \\ \varepsilon & \text{otherwise.} \end{cases} \tag{7}$$

- $s_{(i,j)}(k, k-\mu)$ accounts for the synchronization of vessels k and $k-\mu$ in lock chamber arc (i,j), $\forall k \in \mathcal{K}$, $\forall k-\mu \in \mathcal{K}\backslash\{k\}$, $\forall (i,j) \in \mathcal{D}_{locks}$, such that

$$s_{(i,j)}(k, k-\mu) = \begin{cases} e & \text{if vessels } k \text{ and } k-\mu \text{ are processed in the same} \\ & \text{lockage on arc } (i,j), \\ \varepsilon & \text{otherwise.} \end{cases} \tag{8}$$

- $\alpha_{\chi,(i,j)}(k, g)$ represents the positioning of vessel k inside bin χ of lock chamber (i,j) at time step g, $\forall k \in \mathcal{K}$, $\forall (i,j) \in \mathcal{D}_{locks}$, $\forall \chi \in \{1, \dots, N_{\chi,(i,j)}\}$, $\forall g \in \mathcal{G}$, with $\mathcal{G} = \{0, \dots, n_{steps}-1\}$ and $n_{steps} \triangleq |\mathcal{K}|$, such that

$$\alpha_{\chi,(i,j)}(k, g) = \begin{cases} e & \text{if vessel } k \text{ is slid into bin } \chi \text{ of arc } (i,j) \text{ at time step } g, \\ \varepsilon & \text{otherwise.} \end{cases} \tag{9}$$

  The definition of n_{steps} potentially allows all vessels to be slid into the same lock chamber on the same lockage.
- $h_{\chi,(i,j)}(k, g)$ is the height of the vessel stack in bin χ of lock chamber (i,j) at time step g for vessel k, $\forall k \in \mathcal{K}$, $\forall (i,j) \in \mathcal{D}_{locks}$, $\forall \chi \in \{1, \dots, N_{\chi,(i,j)}\}$, $\forall g \in \mathcal{G}$.
- $m_{(i,j)}(k, k-\mu)$ represents the mooring of vessels k and $k-\mu$ in lock chamber arc (i,j), $\forall k \in \mathcal{K}$, $\forall k-\mu \in \mathcal{K}\backslash\{k\}$, $\forall (i,j) \in \mathcal{D}_{locks}$, such that

$$m_{(i,j)}(k, k-\mu) = \begin{cases} e & \text{if vessel } k \text{ is moored to vessel } k-\mu \text{ on arc } (i,j), \\ \varepsilon & \text{otherwise.} \end{cases} \tag{10}$$

The quality of the solution, which vessels and locks are assumed to comply with, must be assessed using relevant numerical indicators. The following KPIs are considered:

1. Cumulative arrival time, which accounts for the time elapsed from the departure of vessel k to its arrival at its final destination.
2. Arrival time offset, which measures the difference between the scheduled arrival of vessel k at its final destination and its actual arrival.
3. Number of lockages, which keeps track of the total number of lockages that are required so that all vessels can reach their destination. This is a common KPI in lock scheduling problems, as lock operations can unbalance upstream and downstream water levels—even if lock operations may be used to generate hydroelectricity (Pour et al., 2022). Occupied and empty lockages are both counted. A multi-vessel lockage counts as a single lockage.
4. Average delay, which gives an indication of how long vessels have been delayed on their route due to the presence of other vessels in the system.

The proposed approach is divided into two parts. The complete SMPL model of the system, which includes the full list of constraints, is derived in Section 4. The conversion of this model into an MILP model, expressed using conventional algebra, is presented in Section 5. This latter model is used to formulate an optimization problem whose resolution yields the optimal values of the decision variables.

4. SMPL model of the multi-vessel multi-chamber scheduling problem

The SMPL model that will be presented next is formulated by establishing relationships among the max-plus variables defined in Section 3. For convenience, routing, ordering, synchronization, placement and mooring constraints are presented in different subsections.

4.1. Routing constraints

A route must be established between $b(k)$ and $d(k)$ for vessel $k \in \mathcal{K}$, and the scheduled arrival times at each of the nodes visited along the route must be constrained.

4.1.1. Minimum state value constraint

By convention, it is chosen that arrival times cannot be negative. This can be formulated as

$$x_i(k) \geq e, \ \forall k \in \mathcal{K}, \ \forall i \in \mathcal{N}. \tag{11}$$

4.1.2. Departure time constraint

Vessel k can only depart from its origin, i.e., $b(k)$, after its known minimum departure time, i.e., $u(k)$. This can be formulated as

$$x_{b(k)}(k) \geq u(k), \ \forall k \in \mathcal{K}. \tag{12}$$

4.1.3. Route start constraint

Vessel k must start its route by departing from $b(k)$ via an arc $(b(k), i)$ such that $i \in \mathcal{S}(b(k))$, where $\mathcal{S}(b(k))$ denotes the set of nodes connected to node $b(k)$. One, and at most one, routing variable $w_{(b(k),i)}(k)$ must be active. Moreover, vessel k cannot travel back to $b(k)$. This can be formulated as

$$\bigoplus_{i \in \mathcal{S}(b(k))} \left(w_{(b(k),i)}(k) \otimes \bigotimes_{j \in \mathcal{S}(b(k)) \backslash \{i\}} \overline{w}_{(b(k),j)}(k) \right) = e, \ \forall k \in \mathcal{K}. \tag{13}$$

4.1.4. Route end constraint

Vessel k must end its route by arriving at $d(k)$ via an arc $(i, d(k))$ such that $i \in \mathcal{S}(d(k))$, where $\mathcal{S}(d(k))$ denotes the set of nodes connected to node $d(k)$. One, and at most one, routing variable $w_{(i,d(k))}(k)$ must be active. Moreover, vessel k cannot travel away from $d(k)$. This can be formulated as

$$\bigoplus_{i \in \mathcal{S}(d(k))} \left(w_{(i,d(k))}(k) \otimes \bigotimes_{j \in \mathcal{S}(d(k)) \backslash \{i\}} \overline{w}_{(j,d(k))}(k) \right) = e, \ \forall k \in \mathcal{K}. \tag{14}$$

4.1.5. Transit constraint

The intermediate nodes that vessel k visits from origin to destination require active ingoing and outgoing arcs. Likewise, any intermediate node that is not visited by vessel k requires inactive ingoing and outgoing arcs. If there is an active arc (j,i) into node i, $\forall j \in \mathcal{S}(i)$, $\forall i \in \mathcal{N} \setminus \{b(k), d(k)\}$, then there must also be an active arc (i,l) out of node i, $\forall l \in \mathcal{S}(i) \setminus \{j\}$. This can be formulated as

$$\bigoplus_{j \in \mathcal{S}(i)} \bigoplus_{l \in \mathcal{S}(i) \backslash \{j\}} \left(w_{(j,i)}(k) \otimes w_{(i,l)}(k) \otimes \bigotimes_{p \in \mathcal{S}(i) \backslash \{j\}} \overline{w}_{(p,i)}(k) \otimes \bigotimes_{q \in \mathcal{S}(i) \backslash \{l\}} \overline{w}_{(i,q)}(k) \right)$$
$$\oplus \left(\bigotimes_{j \in \mathcal{S}(i)} \overline{w}_{(j,i)}(k) \otimes \bigotimes_{l \in \mathcal{S}(i)} \overline{w}_{(i,l)}(k) \right) = e, \ \forall k \in \mathcal{K}, \ \forall i \in \mathcal{N} \setminus \{b(k), d(k)\}. \tag{15}$$

4.1.6. Travel time constraint

Vessel k can only travel between node i and j at $v_{\max,(i,j)}(k)$, provided that arc (i,j) is active for vessel k. The arrival time at node j must be at least the arrival time at node i plus the minimum travel time on the arc for that vessel. This can be formulated as

$$x_j(k) \geq x_i(k) \otimes \tau_{(i,j)}(k) \otimes w_{(i,j)}(k), \ \forall k \in \mathcal{K}, \ \forall (i,j) \in \mathcal{D}. \tag{16}$$

Given the fact that $\tau_{(i,j)}(k) > 0 \ \forall k \in \mathcal{K}$, $\forall (i,j) \in \mathcal{D}$ is assumed in this paper, (16) has the nice property of preventing vessels from taking sub-tours. Sub-tours are cycles that, while not connected to the route between the origin and destination of vessel k, do fulfill the constraints. Fig. 4 displays a graph and a no longer valid routing solution for a vessel traveling from node 0 to node 2 due to the presence of (16).

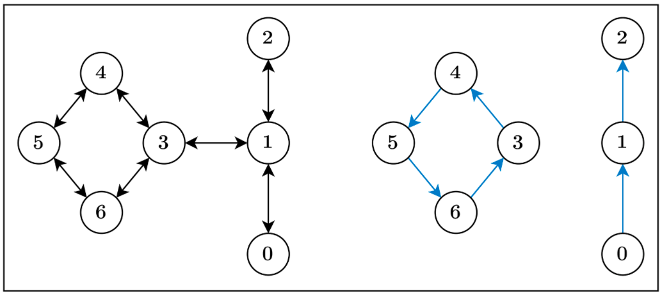

Fig. 4. Prohibited routing solution due to sub-tour (from node 3 to node 3).

4.2. Ordering constraints

Ordering constraints are required to resolve conflicts arising when multiple vessels approach the same lock, either traveling in the same or opposite directions. Note that ordering constraints are only defined for arcs $(i,j) \in \mathcal{D}_{locks}$ as overtaking is not permitted in locks, but it is on regular arcs $(i,j) \in \mathcal{D} \backslash \mathcal{D}_{locks}$.

4.2.1. Same travel direction ordering constraint

Let vessels k and $k-\mu$ travel on the same lock chamber arc (i,j). Let vessel $k-\mu$ be ordered before vessel k on arc (i,j) and assigned to a different lockage than vessel k. Then, vessel k may only enter arc (i,j) after vessel $k-\mu$ has left it and a lockage cycle has taken place (to restore water level). This can be formulated as

$$\begin{aligned} &x_i(k) \geq x_j(k-\mu) \otimes \tau_{(i,j)}(k) \otimes w_{(i,j)}(k-\mu) \otimes w_{(i,j)}(k) \\ &\quad \otimes z_{(i,j)}(k,k-\mu) \otimes \overline{s}_{(i,j)}(k,k-\mu), \\ &\forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \end{aligned} \tag{17}$$

4.2.2. Opposite travel direction ordering constraint

Let vessels k and $k-\mu$ travel on lock chamber arcs (i,j) and (j,i), respectively. Let vessel $k-\mu$ be ordered on arc (j,i) before vessel k is ordered on arc (i,j). Then, vessel k may only enter arc (i,j) after vessel $k-\mu$ has left it at node i. This can be formulated as

$$\begin{aligned} &x_i(k) \geq x_i(k-\mu) \otimes w_{(i,j)}(k) \otimes w_{(j,i)}(k-\mu) \otimes z_{(i,j)}(k,k-\mu), \\ &\quad \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \end{aligned} \tag{18}$$

4.2.3. Ordering variable consistency constraint

Consistency of Eqs. (17) and (18) must be ensured using an additional constraint that links appropriately the ordering of vessels k, $k-\mu \in \mathcal{K}$ on arcs (i,j), $(j,i) \in \mathcal{D}_{locks}$. This can be formulated as

$$\begin{aligned} &\left(z_{(i,j)}(k,k-\mu) \otimes z_{(j,i)}(k,k-\mu) \otimes \overline{z}_{(i,j)}(k-\mu,k)\right) \oplus \\ &\left(\overline{z}_{(i,j)}(k,k-\mu) \otimes \overline{z}_{(j,i)}(k,k-\mu) \otimes z_{(i,j)}(k-\mu,k)\right) = e, \\ &\forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \end{aligned} \tag{19}$$

4.3. Synchronization constraints

Synchronization constraints are required to model the behavior of vessels going through locks simultaneously. While ordering and synchronization principles might appear somewhat similar, only synchronization of lockages must be ensured, thus only affecting lock chamber arcs $(i,j) \in \mathcal{D}_{locks}$. On the other hand, ordering of vessels can potentially be done on all network arcs. However, vessel overtaking on arcs $(i,j) \in \mathcal{D} \backslash \mathcal{D}_{locks}$ is allowed, which overrides the need for ordering constraints on these arcs.

4.3.1. Lockage synchronization constraint

Synchronization of vessel k with vessel $k-\mu$ on lock chamber arc (i,j) has an effect on the starting time of their shared lockage, which can only happen once all vessels have entered the chamber. Therefore, vessel k may only leave arc (i,j) at node j after the latest arrival time of any vessel in the lockage plus the operation time $\tau_{(i,j)}(k)$ and the extra vessel processing time, given by $\overline{\tau}_{(i,j)}(k)$, times the number of vessels with which vessel k is synchronized. This can be formulated as

$$\begin{aligned} &x_j(k) \geq \left(x_i(k) \oplus \bigoplus_{k-\mu \in \mathcal{K}\backslash\{k\}} x_i(k-\mu) \otimes s_{(i,j)}(k,k-\mu) \right) \otimes \tau_{(i,j)} \\ &(k) \otimes \bigotimes_{k-\mu \in \mathcal{K}\backslash\{k\}} \left(\left(s_{(i,j)}(k,k-\mu) \otimes \overline{\tau}_{(i,j)}\right) \oplus e \right) \otimes w_{(i,j)}(k), \\ &\forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}. \end{aligned} \tag{20}$$

4.3.2. Synchronization variable symmetry constraint

Synchronization of vessel k with vessel $k-\mu$ on lock chamber arc (i,j) through variable $s_{(i,j)}(k,k-\mu)$ requires to set the value of $s_{(i,j)}(k-\mu,k)$ accordingly. This can be formulated as

$$\begin{aligned} &\left(s_{(i,j)}(k,k-\mu) \otimes s_{(i,j)}(k-\mu,k)\right) \oplus \left(\overline{s}_{(i,j)}(k,k-\mu) \otimes \overline{s}_{(i,j)}(k-\mu,k)\right) = e, \\ &\qquad \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \end{aligned} \tag{21}$$

4.3.3. Synchronization and routing consistency constraint

Synchronization of vessel k with vessel $k-\mu$ on lock chamber arc (i,j) through variable $s_{(i,j)}(k,k-\mu)$ requires to set the value of $w_{(i,j)}(k)$ accordingly. This can be formulated as

$$\begin{aligned} &\left(w_{(i,j)}(k) \otimes s_{(i,j)}(k,k-\mu)\right) \oplus \left(w_{(i,j)}(k) \otimes \overline{s}_{(i,j)}(k,k-\mu)\right) \oplus \left(\overline{w}_{(i,j)}(k)\right. \\ &\left.\otimes \overline{s}_{(i,j)}(k,k-\mu)\right) = e, \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\}, \forall (i,j) \in \mathcal{D}_{locks}. \end{aligned} \tag{22}$$

4.3.4. Explosive and/or toxic vessel synchronization constraint

A hazardous vessel $k \in \mathcal{K}_{\nabla\nabla}$ may not be processed in the same lockage as vessel $k-\mu$ on any lock chamber arc (i,j). This can be formulated as

$$\bigotimes_{(i,j) \in \mathcal{D}_{locks}} \bigotimes_{k-\mu \in \mathcal{K}\backslash\{k\}} \left(\overline{s}_{(i,j)}(k,k-\mu)\right) = e, \forall k \in \mathcal{K}_{\nabla\nabla}. \tag{23}$$

4.4. Placement constraints

Placement constraints dictate the proper placement sequence of vessels within lock chambers.

4.4.1. Initial condition constraint

Lock chambers must be empty at the start of the chamber-filling process. This can be formulated as

$$\boldsymbol{h}_{(i,j)}(k,0) \geq \boldsymbol{e},\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}, \tag{24}$$

where $\boldsymbol{h}_{(i,j)}(k,g)$ is the ordered vector of all bin heights of vessel k at time step g such that $\boldsymbol{h}_{(i,j)}(k,g) = \begin{bmatrix} h_{0,(i,j)}(k,g) \\ h_{1,(i,j)}(k,g) \\ \vdots \\ h_{N_{\chi,(i,j)}-1,(i,j)}(k,g) \end{bmatrix}$, and $\boldsymbol{e}$ is the column vector of length equal to $N_{\chi,(i,j)}$ with all entries equal to e.

4.4.2. System update constraint

If vessel $k-\mu$ is synchronised with vessel k on lock chamber arc (i,j) and vessel $k-\mu$ is slid into bin χ of the chamber at time step g, then the length of vessel $k-\mu$ will add to the bin height values of vessel k in that chamber at time step $g+1$. This can be formulated as

$$\begin{aligned} &\boldsymbol{h}_{(i,j)}(k,g+1) \geq s_{(i,j)}(k,k-\mu) \otimes \alpha_{\chi,(i,j)}(k-\mu,g) \otimes \boldsymbol{A}_{\chi,(i,j)}(k-\mu) \otimes \boldsymbol{h}_{(i,j)}(k,g), \\ &\qquad \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}, \forall \chi \in \chi^{adm}_{k-\mu}, \end{aligned} \tag{25}$$

where $\chi^{adm}_{k-\mu} \triangleq \{0,1,\ldots,N_{\chi,(i,j)} :\ W_\chi(k-\mu)+\chi \leq N_{\chi,(i,j)}\}$ encompasses the admissible values for χ, such that vessel $k-\mu$ can be placed into without sticking out of the right side of the chamber, as shown in Fig. 5. Moreover, $\boldsymbol{A}_{\chi,(i,j)}(k-\mu)$ is the $N_{\gamma,(i,j)}$-by-$N_{\gamma,(i,j)}$ matrix for ship placement of vessel $k-\mu$ within bin χ of arc (i,j), defined as

$$\boldsymbol{A}_{\chi,(i,j)}(k) \triangleq \begin{bmatrix} \boldsymbol{E}(\chi,\chi) & \mathcal{E}(\chi,W_\chi(k)) & \mathcal{E}(\chi,N_{\gamma,(i,j)}-W_\chi(k)-\chi) \\ \mathcal{E}(W_\chi(k),\chi) & \overline{\boldsymbol{W}}_\gamma(k) & \mathcal{E}(W_\chi(k),N_{\gamma,(i,j)}-W_\chi(k)-\chi) \\ \mathcal{E}(N_{\gamma,(i,j)}-W_\chi(k)-\chi,\chi) & \mathcal{E}(N_{\gamma,(i,j)}-W_\chi(k)-\chi,W_\chi(k)) & \boldsymbol{E}(N_{\gamma,(i,j)}-W_\chi(k)-\chi,N_{\gamma,(i,j)}-W_\chi(k)-\chi)) \end{bmatrix},$$

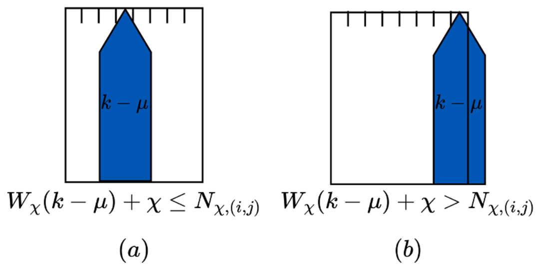

Fig. 5. (a): Vessel is not too wide to be placed in that bin. (b): Vessel is too wide to be placed in that bin.

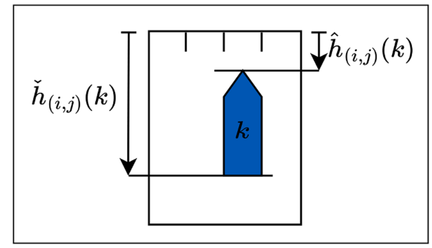

Fig. 6. Bow position $\hat{h}_{(i,j)}(k)$ and stern position $\check{h}_{(i,j)}(k)$ of vessel k in lock chamber arc (i,j).

where $\boldsymbol{E}$ and $\boldsymbol{\mathcal{E}}$ represent the unit and zero matrices of given dimensions, respectively, and $\overline{\boldsymbol{W}}_\gamma(k)$ is the $W_\chi(k)$-by-$W_\chi(k)$ matrix with all entries equal to $W_\gamma(k)$. Note that the ship placement matrix does not exist when the vessel is too large to be placed in that particular bin, either in width or in length.

Moreover, if vessel k itself is placed into the chamber, it should also update its own bin height states. This can be formulated as

$$\boldsymbol{h}_{(i,j)}(k,g+1) \ge \alpha_{\chi,(i,j)}(k,g) \otimes \boldsymbol{A}_{\chi,(i,j)}(k) \otimes \boldsymbol{h}_{(i,j)}(k,g),\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}, \forall \chi \in \chi_k^{adm}, \tag{26}$$

where $\chi_k^{adm} \triangleq \{0,1,\dots,N_{\chi,(i,j)} : W_\chi(k)+\chi \le N_{\chi,(i,j)}\}$.

System update constraints (25) and (26) ensure that vessels placed in the same lockage do not exceed lock chamber height or width.

4.4.3. Bin height synchronization constraint

If vessel k is synchronised with vessel $k-\mu$ on lock chamber arc (i,j), their bin heights must be the same on that chamber at all time steps g. This can be formulated as

$$\boldsymbol{h}_{(i,j)}(k,g) \ge s_{(i,j)}(k,k-\mu) \otimes \boldsymbol{h}_{(i,j)}(k-\mu,g),\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}, \tag{27a}$$

$$\boldsymbol{h}_{(i,j)}(k-\mu,g) \ge s_{(i,j)}(k-\mu,k) \otimes \boldsymbol{h}_{(i,j)}(k,g),\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}. \tag{27b}$$

4.4.4. Bin height increase constraint

The progress in building up bin height for vessel k must be maintained for time steps g at which no other vessel enters lock chamber arc (i,j). This can be formulated as

$$\boldsymbol{h}_{(i,j)}(k,g+1) \ge \boldsymbol{h}_{(i,j)}(k,g),\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}. \tag{28}$$

4.4.5. Routing and placement consistency constraint

If vessel k is routed through lock chamber arc (i,j), it then must be placed into one of the lock chamber bins χ at time step g, and exactly once. This can be formulated as

$$\bigoplus_{g\in\mathcal{G}} \bigoplus_{\chi\in\{0,1,\dots,N_{\chi,(i,j)}-1\,:\,\chi+W_\chi(k)\le N_{\chi,(i,j)}\}} \Big(w_{(i,j)}(k) \otimes \oplus \Big(\overline{w}_{(i,j)}(k) \otimes \bigotimes_{g\in\mathcal{G}} \bigotimes_{\chi\in\{0,1,\dots,N_{\chi,(i,j)}-1\}} \overline{\alpha}_{\chi,(i,j)}(k,g) \Big) = e,\ \alpha_{\chi,(i,j)}(k,g) \otimes \bigotimes_{p\in\mathcal{G}\backslash\{g\}} \bigotimes_{l\in\{0,1,\dots,N_{\chi,(i,j)}-1\}\backslash\{\chi\}} \overline{\alpha}_{l,(i,j)}(k,p) \Big) \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{29}$$

Note that the use of $\{0,1,\dots,N_{\chi,(i,j)}-1\}$ for the adjoint variables ensures that the variables $\alpha_{\chi,(i,j)}(k,g)$ are never active for the bins where vessel width exceeds chamber width.

4.4.6. Single vessel per step constraint

Only a single vessel k may enter lock chamber arc (i,j) at time step g. This can be formulated as

$$\begin{aligned}
&\Big(s_{(i,j)}(k,k-\mu) \otimes \alpha_{\chi,(i,j)}(k,g) \otimes \bigotimes_{p\in\{0,1,\dots,N_{\chi,(i,j)}-1\}} \overline{\alpha}_{p,(i,j)}(k-\mu,g) \Big) \\
&\oplus \Big(s_{(i,j)}(k,k-\mu) \otimes \bigotimes_{p\in\{0,1,\dots,N_{\chi,(i,j)}-1\}} \overline{\alpha}_{p,(i,j)}(k,g) \otimes \alpha_{\chi,(i,j)}(k-\mu,g) \Big) \\
&\oplus \Big(s_{(i,j)}(k,k-\mu) \otimes \overline{\alpha}_{\chi,(i,j)}(k,g) \otimes \bigotimes_{p\in\{0,1,\dots,N_{\chi,(i,j)}-1\}} \overline{\alpha}_{p,(i,j)}(k-\mu,g) \Big) \oplus \\
&\Big(s_{(i,j)}(k,k-\mu) \otimes \bigotimes_{p\in\{0,1,\dots,N_{\chi,(i,j)}-1\}} \overline{\alpha}_{p,(i,j)}(k,g) \otimes \overline{\alpha}_{\chi,(i,j)}(k-\mu,g) \Big) \\
&\oplus \overline{s}_{(i,j)}(k,k-\mu) = e, \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}, \\
&\forall g \in \mathcal{G},\ \forall \chi \in \{0,1,\dots,N_{\chi,(i,j)}-1\}.
\end{aligned} \tag{30}$$

4.5. Mooring constraints

Mooring constraints define the rules for the proper mooring of vessels to each other or the sides of the chamber. These constraints require the introduction of auxiliary variables, shown in Fig. 6, to keep track of the bow and the stern position of each vessel. Let $\hat{h}_{(i,j)}(k)$ be the vertical position of the bow of vessel k in lock chamber arc (i,j), defined as

$$\hat{h}_{(i,j)}(k) \triangleq \bigoplus_{g\in\mathcal{G}} \bigoplus_{\chi\in\{0,1,\dots,N_{\chi,(i,j)}-1|\chi+W_\chi(k)\le N_{\chi,(i,j)}\}} \Big(\alpha_{\chi,(i,j)}(k,g) \otimes \bigoplus_{p\in\{0,1,\dots,W_\chi(k)-1|p+W_\chi(k)\le N_{\chi,(i,j)}\}} h_{\chi+p,(i,j)}(k,g) \Big), \tag{31}$$

and let $\check{h}_{(i,j)}$ be the vertical position of the stern of vessel k in lock chamber arc (i,j), defined as

$$\check{h}_{(i,j)}(k) \triangleq \hat{h}_{(i,j)}(k) \otimes W_\gamma(k). \tag{32}$$

4.5.1. Synchronization and mooring consistency constraint

Vessel k can only be moored to vessel $k-\mu$ on lock chamber arc (i,j) if the vessels are synchronized. This can be formulated as

$$\Big(m_{(i,j)}(k,k-\mu) \otimes s_{(i,j)}(k,k-\mu) \Big) \oplus \Big(s_{(i,j)}(k,k-\mu) \otimes \overline{m}_{(i,j)}(k,k-\mu) \Big) \oplus \Big(\overline{s}_{(i,j)}(k,k-\mu) \otimes \overline{m}_{(i,j)}(k,k-\mu) \Big) = e, \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{33}$$

4.5.2. Single vessel mooring constraint

Vessel k can be moored to at most one vessel in lock chamber arc (i,j). This can be formulated as

$$\Big(\bigotimes_{k-\mu\in\mathcal{K}\backslash\{k\}} \overline{m}_{(i,j)}(k,k-\mu) \Big) \oplus \bigoplus_{k-\mu\in\mathcal{K}\backslash\{k\}} \Big(m_{(i,j)}(k,k-\mu) \otimes \bigotimes_{p\in\mathcal{K}\backslash\{k,k-\mu\}} \overline{m}_{(i,j)}(k,p) \Big) = e, \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{34}$$

4.5.3. Chamber side adjacency constraint

If vessel k is placed adjacent to the sides of lock chamber arc (i,j), it may not be moored to another vessel, hence none of its mooring variables may be true. If vessel k is placed in (i,j), but is not adjacent to the sides of the chamber, one of its mooring variables must be true. This can be formulated as

$$\Big(\overline{w}_{(i,j)}(k) \otimes \bigotimes_{k-\mu\in\mathcal{K}\backslash\{k\}} \overline{m}_{(i,j)}(k,k-\mu) \Big) \oplus \Big(w_{(i,j)} \otimes \bigoplus_{k-\mu\in\mathcal{K}\backslash\{k\}}$$

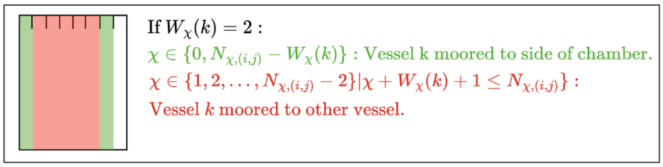

Fig. 7. Mooring possibilities for vessel k ($W_\chi(k) = 2$).

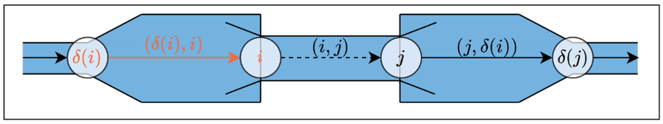

Fig. 8. Lock chamber arc (i, j) with node $\delta(i)$ and arc $(\delta(i), i)$ highlighted.

$$\bigoplus_{g \in \mathcal{G}} \bigoplus_{\chi \in \{1,2,\ldots,N_{\chi,(i,j)}-2\}:\ \chi + W_\chi(k) + 1 \leq N_{\chi,(i,j)}\}} \Big(\alpha_{\chi,(i,j)}(k,g) \otimes m_{(i,j)}(k, k-\mu)\Big)\Big) \oplus$$
$$\Big(w_{(i,j)}(k) \otimes \bigoplus_{k-\mu \in \mathcal{K} \setminus \{k\}} \bigoplus_{g \in \mathcal{G}} \bigoplus_{\chi \in \{0, N_{\chi,(i,j)} - W_\chi(k)\}} \alpha_{\chi,(i,j)}(k,g) \otimes \overline{m}_{(i,j)}(k, k-\mu)\Big)$$
$$= e,\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{35}$$

Note that the possible values of χ are split into two sets, one in which vessel k is adjacent to the side of the chamber, and another in which it is not. An example of the bins in which vessel k, with $W_\chi(k) = 2$, would be moored is provided in Fig. 7: in green, the bins in which it would be moored to the side of the chamber; in red, the bins in which it would have to be moored to another vessel. In this particular example, the top left corner of the vessel cannot be placed into the rightmost bin, as the vessel would be too wide for the chamber. Therefore, the chamber side adjacency constraint (35) is also required to ensure that mooring operations (to either one of the chamber sides or another vessel) observe total lock chamber width.

4.5.4. *Vessel adjacency constraint*

Vessel k can only be moored to vessel $k - \mu$ on lock chamber (i, j) arc if they are adjacent. However, it is not required that two vessels be moored to each other if they are adjacent. This can be formulated as

$$\Big(m_{(i,j)}(k, k-\mu) \otimes \bigoplus_{l \in \mathcal{G}} \bigoplus_{g \in \mathcal{G}} \bigoplus_{\chi \in \{1,2,\ldots,N_{\chi,(i,j)} - W_\chi(k) - 1\}} \bigoplus_{q \in \{\chi - W_\chi(k-\mu), \chi + W_\chi(k)\}}$$
$$\alpha_{\chi,(i,j)}(k,g) \otimes \alpha_{q,(i,j)}(k-\mu, l)\Big) \oplus \overline{m}_{(i,j)}(k, k-\mu) = e,\ \forall k \in \mathcal{K},$$
$$\forall k - \mu \in \mathcal{K} \setminus \{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{36}$$

The set $\{1, 2, \ldots, N_{\chi,(i,j)} - W_\chi(k) - 1\}$ checks all of the bins χ in which vessel k can be placed where mooring would be required, and the set $\{\chi - W_\chi(k-\mu), \chi + W_\chi(k)\}$ then checks if vessel $k - \mu$ is placed to the left or right of vessel k if it is placed in bin χ.

4.5.5. *Mooring anti-symmetry constraint*

If vessel k is moored to vessel $k - \mu$ on lock chamber arc (i, j), then vessel $k - \mu$ cannot be moored to vessel k on that same chamber. This can be formulated as

$$\big(m_{(i,j)}(k, k-\mu) \otimes \overline{m}_{(i,j)}(k-\mu, k)\big) \oplus \big(\overline{m}_{(i,j)}(k, k-\mu) \otimes m_{(i,j)}(k-\mu, k)\big) \oplus$$
$$\big(\overline{m}_{(i,j)}(k, k-\mu) \otimes \overline{m}_{(i,j)}(k-\mu, k)\big)$$
$$= e,\ \forall k \in \mathcal{K},\ \forall k - \mu \in \mathcal{K} \setminus \{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{37}$$

4.5.6. *Flammable vessel constraint*

Vessels $k \in \mathcal{K}_\nabla$ must keep 10 m of distance from all other vessels, so they may not be moored to other vessels. This means that they must then always be moored to the sides of the chamber. This can be formulated as

$$\bigoplus_{k-\mu \in \mathcal{K} \setminus \{k\}} m_{(i,j)}(k, k-\mu) = \varepsilon,\ \forall k \in \mathcal{K}_\nabla,\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{38}$$

5. Optimization problem using an MILP reformulation

Max-plus algebra is well suited to describe the evolution of event-based systems such as IWT. However, the SMPL constraints given in Section 4 must be transformed using conventional algebra to solve the optimization problem. The reason for this is twofold. On the one hand, MILP solvers cannot use ε, as $\varepsilon \triangleq -\infty$. On the other hand, the maximization operator cannot be directly used in an MILP problem (although it can be implemented using additional auxiliary variables and constraints).

This section begins with an overview of MILP problem formulation . The SMPL constraints derived in Section 4 are presented next, together with the non-SMPL constraints that must be added for completeness. Then, the operational objectives to be optimized are discussed.

5.1. *Overview of MILP problems*

MILP problems can be written as

$$\min_{\boldsymbol{x},\boldsymbol{\phi}} \quad J = c_x^T \boldsymbol{x} + \boldsymbol{c}_\phi^T \boldsymbol{\phi} \tag{39}$$
$$\text{s.t} \quad \boldsymbol{A}_x \boldsymbol{x} + \boldsymbol{A}_\phi \boldsymbol{\phi} \leq \boldsymbol{f}, \quad \boldsymbol{x} \geq 0, \quad \boldsymbol{\phi} \in \mathbb{B}^{n_\phi},$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ and $\boldsymbol{\phi} \in \mathbb{B}^{n_\phi}$ are the vectors of continuous and binary decision variables, respectively, with $\mathbb{B} \triangleq \{0, 1\}$, n_x and n_ϕ are the lengths of vectors $\boldsymbol{x}$ and $\boldsymbol{\phi}$, respectively, and $\boldsymbol{c}_x$, $\boldsymbol{c}_\phi$, $\boldsymbol{A}$ and $\boldsymbol{f}$ are arrays of coefficients of appropriate dimensions. Binary variables are defined such that ϕ_i, which denotes the i-th entry of $\boldsymbol{\phi}$, is equal to 0 and 1 when it is in the false and true state, respectively, $\forall i \in \{1, 2, \ldots, n_\phi\}$.

Max-plus binary variables, on the other hand, are either equal to e or ε as per Eq. (4). Let ϑ be a max-plus binary variable, with $\overline{\vartheta}$ its max-plus adjoint. Then, ϑ can be approximated in conventional algebra for inequality constraints as (van den Boom et al., 2020):

Max-plus algebra $\longleftrightarrow$	Conventional algebra
ϑ	$\beta(1-\vartheta)$
$\overline{\vartheta}$	$\beta\vartheta$
$\vartheta, \overline{\vartheta} \in \mathbb{B}_{max}$	$\vartheta \in \mathbb{B}$

where $\mathbb{B}_{\max} \triangleq \{e, \varepsilon\}$ and β is a large enough negative scalar, such that $\beta \approx \varepsilon$. The use of β is a form of big-M notation, and its value should be reduced to a reasonable bound to avoid numerical problems and reduce computation time by allowing for better problem relaxation (Trespalacios and Grossmann, 2015).

In what follows, the constraints and cost function of the MILP problem described by (39) will be defined.

5.2. *MILP constraints*

Each of the constraints provided in Section 4 is expressed using conventional algebra as presented in Kersbergen (2015). Note that these are expressed as smaller-than-inequality constraints to fit into the general formulation introduced in Eq. (39). Additional auxiliary variables are defined where appropriate.

5.2.1. *Routing constraints*

The minimum value start constraint, given by (11), is transformed into

$$-x_i(k) \leq 0,\ \forall k \in \mathcal{K},\ \forall i \in \mathcal{N}. \tag{40}$$

The minimum state value constraint, given by (12), is transformed into

$$-x_{b(k)} \leq -u(k),\ \forall k \in \mathcal{K}. \tag{41}$$

The route start constraint, given by (13), is transformed into

$$\sum_{i \in S(b(k))} w_{(b(k),i)}(k) \leq 1,\ \forall k \in \mathcal{K}, \tag{42a}$$

$$-\sum_{i\in\mathcal{S}(b(k))} w_{(b(k),i)}(k) \leq -1,\ \forall k \in \mathcal{K}. \tag{42b}$$

The route end constraint, given by (14), is transformed into

$$\sum_{i\in\mathcal{S}(d(k))} w_{(i,d(k))}(k) \leq 1,\ \forall k \in \mathcal{K}, \tag{43a}$$

$$-\sum_{i\in\mathcal{S}(d(k))} w_{(i,d(k))}(k) \leq -1,\ \forall k \in \mathcal{K}. \tag{43b}$$

The transit constraint, given by (15), is transformed into

$$\sum_{j\in\mathcal{S}(i)} w_{(i,j)}(k) - \sum_{j\in\mathcal{S}(i)} w_{(j,i)}(k) \leq 0, \forall k \in \mathcal{K},\ \forall i \in \mathcal{N} \setminus \{b(k), d(k)\}, \tag{44a}$$

$$\sum_{j\in\mathcal{S}(i)} w_{(j,i)}(k) - \sum_{j\in\mathcal{S}(i)} w_{(i,j)}(k) \leq 0, \forall k \in \mathcal{K},\ \forall i \in \mathcal{N} \setminus \{b(k), d(k)\}, \tag{44b}$$

$$\sum_{j\in\mathcal{S}(i)} w_{(j,i)}(k) \leq 1,\ \forall k \in \mathcal{K},\ \forall i \in \mathcal{N} \setminus \{b(k), d(k)\}. \tag{44c}$$

The travel time constraint, given by (16), is transformed into

$$-x_j(k) + x_i(k) - \beta w_{(i,j)}(k) \leq -\beta - \tau_{(i,j)}(k),\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}. \tag{45}$$

5.2.2. Ordering constraints

The same travel direction ordering constraint, given by (17), is transformed into

$$- x_i(k) + x_j(k-\mu) - \beta w_{(i,j)}(k) - \beta w_{(i,j)}(k-\mu) - \beta z_{(i,j)}(k, k-\mu)$$
$$\leq -3\beta - \tau_{(j,i)}(k), \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{46}$$

The opposite travel direction ordering constraint, given by (18), is transformed into

$$- x_i(k) + x_i(k-\mu) - \beta w_{(i,j)}(k) - \beta w_{(j,i)}(k-\mu) - \beta z_{(i,j)}(k, k-\mu) \leq -3\beta,$$
$$\forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{47}$$

The ordering variable consistency constraint, given by (19), is transformed into

$$z_{(i,j)}(k, k-\mu) - z_{(j,i)}(k, k-\mu) \leq 0,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks} \tag{48a}$$

$$z_{(j,i)}(k, k-\mu) - z_{(i,j)}(k, k-\mu) \leq 0,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks} \tag{48b}$$

$$z_{(i,j)}(k, k-\mu) + z_{(i,j)}(k-\mu, k) \leq 1,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks} \tag{48c}$$

$$-z_{(i,j)}(k, k-\mu) - z_{(i,j)}(k-\mu, k) \leq -1,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{48d}$$

5.2.3. Synchronization constraints

Two auxiliary variables must be introduced to represent the SMPL synchronization constraints from Section 4.3 using conventional algebra. Define $\theta_{synch,(i,j)}(k, k-\mu)$ as

$$\theta_{synch,(i,j)}(k, k-\mu) \triangleq x_i(k-\mu) s_{(i,j)}(k, k-\mu),\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},$$
$$\forall (i,j) \in \mathcal{D}_{locks}, \tag{49}$$

and define $t_{synch,(i,j)}(k)$ as

$$t_{synch,(i,j)}(k) \triangleq \max\left(x_i(k), \max\left(\theta_{synch,(i,j)}(k, k-\mu)\right)\right),\ \forall k \in \mathcal{K},\ \forall k-\mu$$
$$\in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}, \tag{50}$$

both of which can be linearized as shown in Bemporad and Morari (1999).

The lockage synchronization constraint, given by (20), is transformed into

$$t_{synch(i,j)}(k) - x_j(k) - \beta w_{(i,j)}(k) + \sum_{k-\mu\in\mathcal{K}\backslash\{k\}} s_{(i,j)}(k, k-\mu)\overline{\tau}_{(i,j)} \leq -\beta$$
$$- \tau_{(i,j)}(k), \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{51}$$

The synchronization variable symmetry constraint, given by (21), is transformed into

$$s_{(i,j)}(k, k-\mu) - s_{(i,j)}(k-\mu, k) \quad \leq 0,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}, \tag{52a}$$

$$s_{(i,j)}(k-\mu, k) - s_{(i,j)}(k, k-\mu) \quad \leq 0,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{52b}$$

The synchronization and routing consistency constraint, given by (22), is transformed into

$$s_{(i,j)}(k, k-\mu) - w_{(i,j)}(k) \leq 0,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{53}$$

The explosive and/or toxic vessel synchronization constraint, given by (23), is transformed into

$$\sum_{(i,j)\in\mathcal{D}_{locks}} \sum_{k-\mu\in\mathcal{K}\backslash\{k\}} s_{(i,j)}(k, k-\mu) \leq 0,\ \forall k \in \mathcal{K}_{\nabla\nabla}. \tag{54}$$

5.2.4. Placement constraints

The initial condition constraint, given by (24), is transformed into

$$h_{\chi,(i,j)}(k) \leq 0,\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall \chi \in \{0, 1, \dots, N_{\chi,(i,j)} - 1\}, \tag{55a}$$

$$-h_{\chi,(i,j)}(k) \leq 0,\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall \chi \in \{0, 1, \dots, N_{\chi,(i,j)} - 1\}. \tag{55b}$$

The system update constraints, given by (25) and (26), can no longer be conveniently expressed using the original vector notation, and are instead written on a single-entry basis as

$$h_{\chi,(i,j)}(k, g) - h_{\chi+p,(i,j)}(k, g+1) - \beta s_{(i,j)}(k, k-\mu) - \beta \alpha_{\chi,(i,j)}(k-\mu) \leq -2\beta$$
$$- W_{\gamma}(k-\mu), \forall k \in \mathcal{K}, \forall k-\mu \in \mathcal{K} \setminus \{k\},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G},\ \forall \chi$$
$$\in \{0, 1, \dots, N_{\chi,(i,j)} - 1 :\ \chi + W_{\chi}(k-\mu) \leq N_{\chi,(i,j)}\}, \forall p$$
$$\in \{0, 1, \dots, W_{\chi}(k-\mu) - 1\} \tag{56}$$

and

$$h_{\chi,(i,j)}(k, g) - h_{\chi+p,(i,j)}(k, g+1) - \beta \alpha_{\chi,(i,j)}(k-\mu) \leq -\beta - W_{\gamma}(k),\ \forall k$$
$$\in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}, \forall \chi \in \{0, 1, \dots, N_{\chi,(i,j)} - 1 :\ \chi + W_{\chi}(k)$$
$$\leq N_{\chi,(i,j)}\},\ \forall p \in \{0, 1, \dots, W_{\chi}(k) - 1\}, \tag{57}$$

respectively.

The bin height increase constraint, given by (28), is also written on a single-entry basis as

$$h_{\chi,(i,j)}(k, g) - h_{\chi,(i,j)}(k, g+1) \leq 0,\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g$$
$$\in \mathcal{G},\ \forall \chi \in \{0, 1, \dots, N_{\chi,(i,j)} - 1\}. \tag{58}$$

The routing and placement consistency constraint, given by (29), is transformed into

$$\sum_{\chi\in\{0,1,\dots,N_{\chi,(i,j)-1}\}} \sum_{g\in\mathcal{G}} \alpha_{\chi,(i,j)}(k, g) - w_{(i,j)}(k) \leq 0,\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks} \tag{59a}$$

$$-\sum_{\chi\in\{0,1,\dots,N_{\chi,(i,j)-1}\}} \sum_{g\in\mathcal{G}} \alpha_{\chi,(i,j)}(k, g) + w_{(i,j)}(k) \leq 0,\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{59b}$$

The single vessel per step constraint, given by (30), is transformed into

$$\sum_{p\in\{0,1,\dots,N_{\chi,(i,j)-1}\}} \alpha_{p,(i,j)}(k-\mu, g) + s_{(i,j)}(k, k-\mu) + \alpha_{\chi,(i,j)}(k, g)+ \leq 2,\ \forall k$$
$$\in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\}, \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G},\ \forall \chi \in \{0, 1, \dots, N_{\chi,(i,j)-1}\}. \tag{60}$$

5.2.5. Mooring constraints

Two auxiliary variables must be introduced to represent the vertical position of the bow, given by (31), using conventional algebra. Define $t_{bow,\chi,(i,j)}(k, g)$ as

$$t_{bow,\chi,(i,j)}(k, g) \triangleq \max\left(\{h_{\chi+p,(i,j)}(k, g)\}\right),\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G},$$

$$\forall \chi \in \{0,1,\ldots,N_{\chi,(i,j)}-W_{\chi}(k)\},\ \forall p \in \{0,1,\ldots,W_{\chi}(k)-1\}\}, \quad (61)$$

and define $\theta_{bow,\chi,(i,j)}(k,g)$ as

$$\theta_{bow,\chi,(i,j)}(k,g) \triangleq \alpha_{\chi,(i,j)}(k,g)t_{bow,\chi,(i,j)}(k,g+1),\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}, \forall \chi \in \{0,1,\ldots,N_{\chi,(i,j)}-W_{\chi}(k)\}. \quad (62)$$

Then, the bow position constraint is given by

$$\hat{h}_{(i,j)}(k) = \max\left(\{\theta_{bow,\chi,(i,j)}(k,g)\}\right),\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}, \forall \chi \in \{0,1,\ldots,N_{\chi,(i,j)}-W_{\chi}(k)\}, \quad (63)$$

and (61)–(63) can be linearized as shown in Bemporad and Morari (1999).

The stern position constraint, given by (32), is transformed into

$$\check{h}_{(i,j)}(k) - \hat{h}_{(i,j)}(k) \le W_{\gamma}(k),\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}, \quad (64a)$$

$$-\check{h}_{(i,j)}(k) + \hat{h}_{(i,j)}(k) \le -W_{\gamma}(k),\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}. \quad (64b)$$

The synchronization and mooring consistency constraint, given by (33), is transformed into

$$m_{(i,j)}(k,k-\mu) - s_{(i,j)}(k,k-\mu) \le 0,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \quad (65)$$

The single vessel mooring constraint, given by (34), is transformed into

$$\sum_{k-\mu \in \mathcal{K}\backslash\{k\}} m_{(i,j)}(k,k-\mu) \le 1,\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}. \quad (66)$$

The chamber side adjacency constraint, given by (35), is transformed into

$$\sum_{\chi \in \{1,2,\ldots,N_{\chi,(i,j)}-2\,:\,\chi+W_{\chi}(k)+1 \le N_{\chi,(i,j)}\}} \sum_{g \in \mathcal{G}} \alpha_{\chi,(i,j)}(k,g) - \sum_{k-\mu \in \mathcal{K}\backslash\{k\}} m_{(i,j)}(k,k-\mu) \le 0, \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}. \quad (67)$$

Two auxiliary binary variables must be introduced to represent the vessel adjacency constraint, given by (36), using conventional algebra. Define $\psi_{\chi-W_{\chi}(k-\mu),(i,j)}(k,k-\mu)$ as

$$\psi_{\chi-W_{\chi}(k-\mu),(i,j)}(k,k-\mu) \triangleq \left(\sum_{g \in \mathcal{G}} \alpha_{\chi,(i,j)}(k,g)\right) \left(\sum_{g \in \mathcal{G}} \alpha_{\chi-W_{\chi}(k-\mu),(i,j)}(k-\mu,g)\right), \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}, \quad (68)$$

which only exists if $\chi - W_{\chi}(k-\mu) \ge 0$ for arc (i,j), and define $\psi_{\chi+W_{\chi}(k),(i,j)}(k,k-\mu)$ as

$$\psi_{\chi+W_{\chi}(k),(i,j)}(k,k-\mu) \triangleq \left(\sum_{g \in \mathcal{G}} \alpha_{\chi,(i,j)}(k,g)\right)\left(\sum_{g \in \mathcal{G}} \alpha_{\chi+W_{\chi}(k),(i,j)}(k-\mu,g)\right), \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}, \quad (69)$$

which only exists if $\chi + W_{\chi}(k) \le N_{\chi,(i,j)}$ for chamber (i,j).

Noting that (68) and (69) can be linearized as shown in Bemporad and Morari (1999), the vessel adjacency constraint is transformed into

$$-\sum_{\chi \in \{W_{\chi}(k-\mu),W_{\chi}(k-\mu)+1,\ldots,N_{\chi,(i,j)}-2\,:\,\chi+W_{\chi}(k)+1 \le N_{\chi,(i,j)}\}} \times \left(\psi_{\chi+W_{\chi}(k),(i,j)}(k,k-\mu) + \psi_{\chi-W_{\chi}(k-\mu),(i,j)}(k,k-\mu)\right) + m_{(i,j)}(k,k-\mu) \le 0,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \quad (70)$$

The mooring anti-symmetry constraint, given by (37), is transformed into

$$m_{(i,j)}(k,k-\mu) + m_{(i,j)}(k-\mu,k) \le 1,\ \forall k \in \mathcal{K}, \forall k-\mu \in \mathcal{K}\backslash\{k\},\ \forall (i,j) \in \mathcal{D}_{locks}. \quad (71)$$

The flammable vessel constraint, given by (38), is transformed into

$$\sum_{k-\mu \in \mathcal{K}\backslash\{k\}} m_{(i,j)}(k,k-\mu) = 0,\ \forall k \in \mathcal{K}_{\nabla},\ \forall (i,j) \in \mathcal{D}_{locks}. \quad (72)$$

5.2.6. Non-SMPL constraints

Two more sets of constraints on the bin height states are required to model the real lockage properly. Both require constraining the state variable of the system to be smaller than another quantity, which does not fit into the SMPL framework for state evolution equations. This is why they were not discussed in Section 4.

The chamber length constraint, which ensures that the bin height of vessel k does not exceed the length of lock chamber arc (i,j), can be formulated as

$$h_{\chi,(i,j)}(k,g) \le N_{\gamma,(i,j)},\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks},\ \forall g \in \mathcal{G}, \forall \chi \in \{0,1,\ldots,N_{\chi,(i,j)}\}. \quad (73)$$

The fully alongside constraint, which ensures that vessel k is fully alongside vessel $k-\mu$ in order for vessel k to be moored to vessel $k-\mu$, can be formulated as

$$\hat{h}_{(i,j)}(k) - \hat{h}_{(i,j)}(k-\mu) - \beta m_{(i,j)}(k,k-\mu) \quad \le -\beta,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\}, \forall (i,j) \in \mathcal{D}_{locks}, \quad (74a)$$

$$\check{h}_{(i,j)}(k-\mu) - \check{h}_{(i,j)}(k) - \beta m_{(i,j)}(k,k-\mu) \quad \le -\beta,\ \forall k \in \mathcal{K},\ \forall k-\mu \in \mathcal{K}\backslash\{k\}, \forall (i,j) \in \mathcal{D}_{locks}. \quad (74b)$$

where the value of β can be set as discussed in Section 5.1.

5.3. Cost function and resulting optimization problem

Two main objectives are considered in this paper. The first represents the situation wherein vessels want to arrive at their destination as quickly as possible. This is governed by the cumulative travel time objective, which can be computed as

$$J_{\mathcal{A}} = \sum_{k \in \mathcal{K}} \sigma_{\mathcal{A}}(k)\big(x_{d(k)}(k) - u(k)\big), \quad (75)$$

where $u(k)$ and $x_d(k)$ denote the departure and arrival time of vessel k at its destination, respectively, and $\sigma_{\mathcal{A}}(k)$, which is the cumulative travel time objective for vessel k, can be given different values to account for different priority levels.

The second main objective considers the situation in which vessels have an intended arrival time at their final destination, so arriving early or late can be undesirable. This is governed by the arrival time offset objective, which can be computed as

$$J_{\mathcal{H}} = \sum_{k \in \mathcal{K}} \sigma_{\mathcal{H}}(k)|\hat{x}_{d(k)}(k) - x_{d(k)}(k)|, \quad (76)$$

where $\hat{x}_{d(k)}(k)$ represents the planned arrival time at node $d(k)$ for vessel k and $\sigma_{\mathcal{H}}(k)$ is the arrival time offset weight for vessel k. This objective can be linearized using the auxiliary variable $t_{\mathcal{H}}(k) \triangleq |\hat{x}_{d(k)} - x_{d(k)}|$ and following the approach presented in Bemporad and Morari (1999).

It should be noted that a vessel can only be assigned one of the two main objectives, depending on whether the preference is to reach the destination as soon as possible or as close to its scheduled arrival as possible, respectively. The desired behavior can be selected by setting the corresponding objective weight $\sigma_{\mathcal{A}}(k)$ or $\sigma_{\mathcal{H}}(k)$ equal to a nonzero value, while the other weight is set equal to 0.

Two other auxiliary objectives, which are intended to enforce certain beneficial behavior, are added to the cost function. The first is a cumulative departure time objective, and ensures that vessel departure times are always equal to their minimum departure time. Although this could result in longer vessel travel times, vessels move at a slower pace on average, potentially reducing emissions. This auxiliary objective can be computed as

$$J_u = \sum_{k \in \mathcal{K}} x_{b(k)}(k) \cdot 10^{-3}, \quad (77)$$

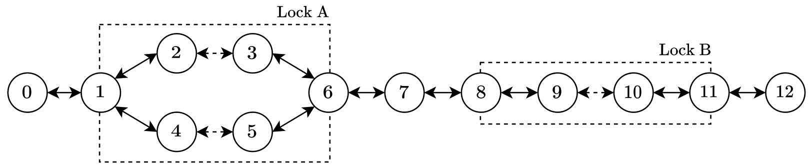

Fig. 9. Topology graph of the considered case study.

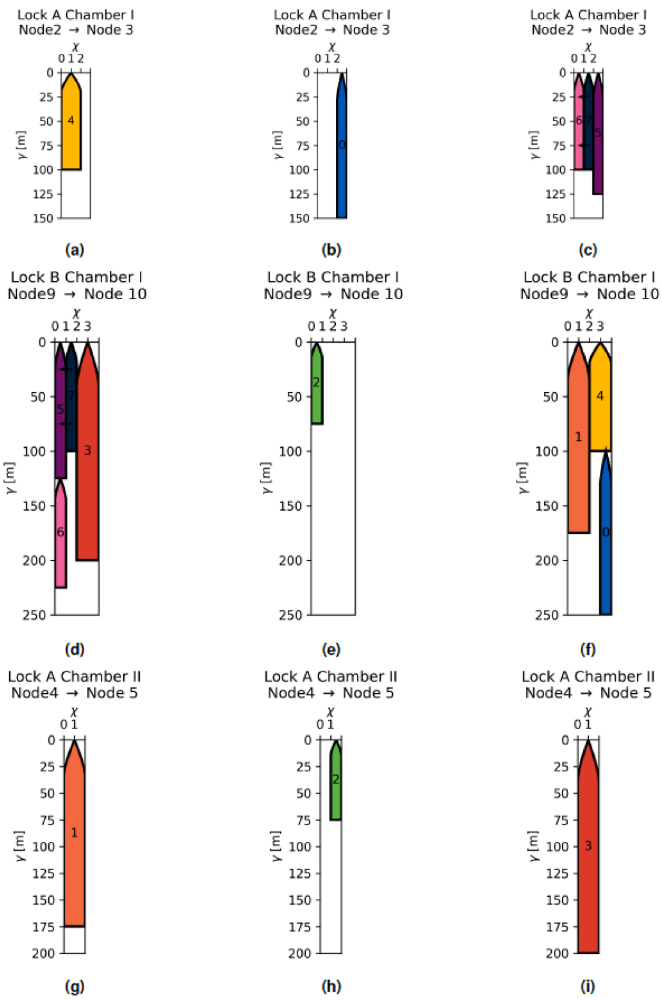

Fig. 10. Populated lockages for first test.

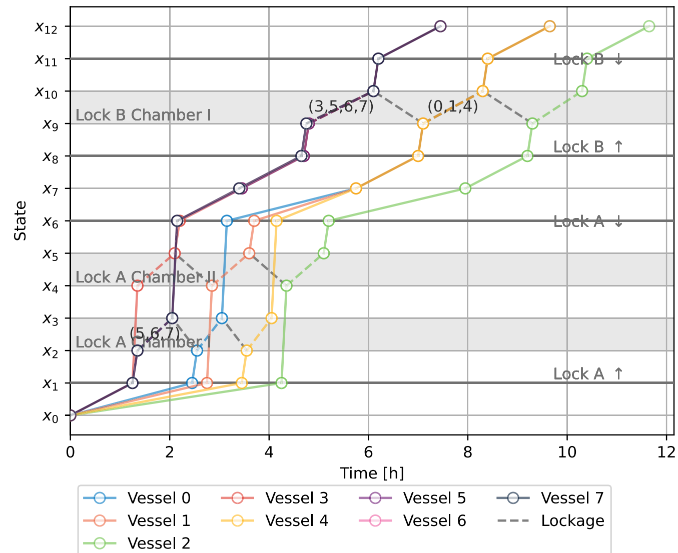

Fig. 11. Journey plots for first test.

where $x_{b(k)}(k)$ is the departure time of vessel k, $\forall k \in \mathcal{K}$, to which a small penalty is applied.

The second auxiliary objective is a waiting time objective, and ensures that a vessel that has to wait for a lockage does so by reducing speed on a non-waiting area arc, rather than by coming to a full stop in a waiting area of a lock. This, again, encourages slow-steaming behavior, and has no influence on vessel final arrival times. Fig. 8 provides a schematic representation of an arbitrary lock chamber arc (i, j), where $\delta(i)$ is the node of the arc that precedes (i, j), i.e., $(\delta(i), i)$. Let

$$\pi_{(i,j)}(k) \triangleq w_{(i,j)}(k)\big(x_i(k) - x_{\delta(i)}(k) - \tau_{(i,\delta(i))}(k)\big),\ \forall k \in \mathcal{K}, \forall (i,j) \in \mathcal{D}_{locks} \tag{78}$$

denote the waiting time spent on $(\delta(i), i)$ for vessel k minus the minimum travel time $\tau_{(\delta(i),i)}(k)$—as that time would be spent on that arc by the vessel even if it did not have to wait. A linear objective with the same underlying meaning can be built by defining the continuous variable $t_{\Pi,(i,j)}(k)$ as

$$t_{\Pi,(i,j)}(k) \triangleq \max\big(x_i(k) - x_{\delta(i)}(k) - \tau_{(i,\delta(i))}(k) + \beta(1 - w_{(i,j)}(k)), 0\big),\ \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{D}_{locks}. \tag{79}$$

After linearizing Eqn. (79), the vessel waiting time auxiliary objective can be computed as

$$J_\Pi = \sum_{(i,j)\in\mathcal{D}_{locks}} \sum_{k\in\mathcal{K}} t_{\Pi,(i,j)}(k) \cdot 10^{-3}. \tag{80}$$

The choice made in this paper with regard to the final cost function, which is the sum of (75), (76), (77) and (80), raises the question of whether lockage- and placement-related objectives should also be considered. On the one hand, empty lockages in between consecutive lockages in the same direction are difficult to linearly count in the objective function with the current model. If empty lockages were neglected, the number of lockages objective could be off by as much as a factor of two. On the other hand, the inclusion of placement-related objectives, e.g., penalize vessels not being placed as far forward within lock chambers as possible, would further increase the number of constraints, potentially leading to larger computation times. Instead, this could be handled by post-processing, wherein the optimal schedules could be used as a starting point by a lockmaster or a secondary heuristic algorithm, making small changes to improve solution practicality. It must also be taken into account that the four objectives selected represent time, whereas lockage- and placement-related objectives are likely to represent other magnitudes, hence making it harder to balance with the time-based objectives.

The resulting MILP problem can be written as

$$\min_{x, w, z, s, \alpha, h, m} \quad J = J_{\mathcal{A}} + J_{\mathcal{H}} + J_u + J_\Pi \tag{81}$$

s.t Routing constraints (40)–(45), Ordering constraints (46)–(48), Synchronization constraints (51)–(54), Placement constraints (55)–(60),

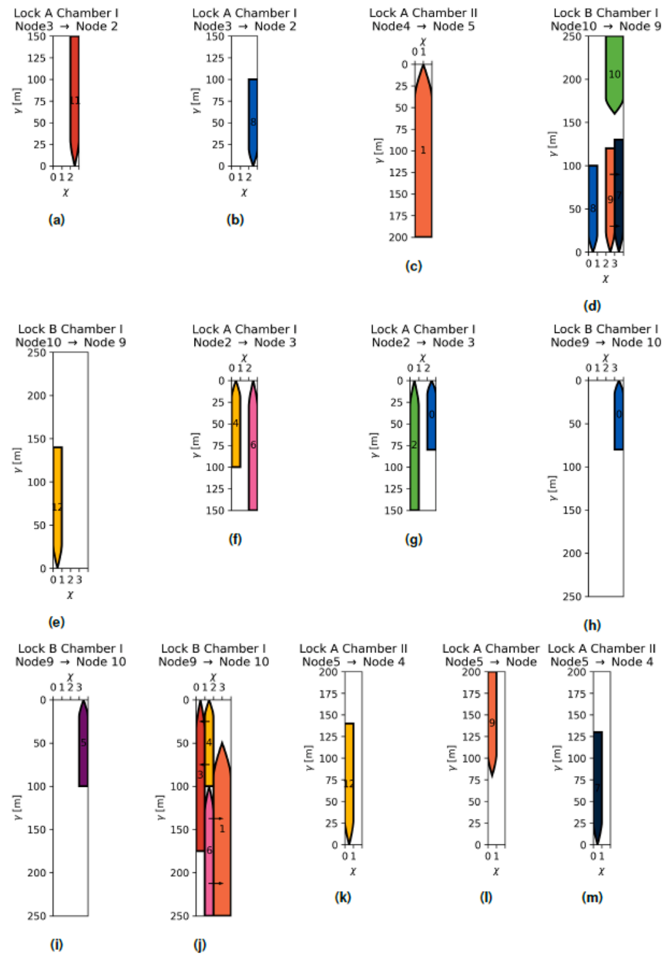

Fig. 12. Populated lockages for second test.

Table 2
Node parameters.

Node i	$p_{x,i}$ [km]	$p_{y,i}$ [km]
0	0	0
Lock A(1, 2, 3, 4, 5, 6)	12.5	0
7	25	0
Lock B(8, 9, 10, 11)	37.5	0
12	50	0

Mooring constraints (61)–(72), Non-SMPL constraints (73)–(74), where $\boldsymbol{x}$, $\boldsymbol{w}$, $\boldsymbol{z}$, $\boldsymbol{s}$, $\boldsymbol{\alpha}$, $\boldsymbol{h}$, $\boldsymbol{m}$ are arrays of the decision variables defined in Section 3 of appropriate dimensions.

6. Case study

The scheduling approach presented in this paper is tested considering a waterway with a multi-chamber lock and a single-chamber lock connected in series. The topology graph is depicted in Fig. 9, with the node and arc parameters summarized in Tables 2 and 3, respectively.

Section 6.1 presents a first test that is designed to be simple enough so that predictable results are obtained. This shall allow to verify the correctness of the implementation and assess its performance in a quantitative manner using KPIs. A second and more general test is then considered in Section 6.2, and the KPIs are compared to those obtained using a time-based simulation model that is representative of the current state of practice.

6.1. Test 1: Vessels carrying flammable and explosive goods

The test considers eight vessels of different sizes moving in the same direction and departing at the same time instant, as listed in Table 4. Vessel 0 is carrying flammable goods, so it should not moor to any other vessel. Vessel 2 is carrying explosive goods, so it should not be grouped into lockages with other vessels. The cumulative arrival time objective is used.

The nine populated lockages of the solution are shown in Fig. 10. It can be seen that vessel 0 is not moored to any other vessel in any of the lockages, and vessel 2 is always placed into separate lockages. The schedule in Fig. 11 shows the predictable pattern of prioritizing lockages with multiple vessels on lock A, followed by lockages for vessels that can join grouped lockages on lock B, and then having vessel 2 go last to prevent other vessels being held up. Lock B has a larger chamber than either of the chambers of lock A, so more vessels can fit into its lockages, causing vessels that passed through lock A separately to be grouped together on lock B. It can be concluded that the model is functioning correctly for multiple locks and vessels with flammable or explosive goods.

The numerical values of the KPIs defined in Section 3 are summarized in Table 5. It is recalled that the two first KPIs coincide with the first two objectives in the cost function, given by Eqs. (75) and (76), respectively, $\mathcal{L}$ is the number of lockages and $\mathcal{T}$ is the average delay, which can be computed as

$$\tau = \sum_{\{k\in\mathcal{K}|\sigma_H(k)=0\}} \frac{\tau(k)}{|\{k\in\mathcal{K}|\sigma_H(k)=0\}|}, \tag{82}$$

where the denominator is the number of vessels with inactive arrival time offset objective, and $\tau(k)$ is computed as

$$\tau(k)[\%] = 100\left(\frac{\sum_{(i,j)\in D} w_{(i,j)}(k)(x_j(k)-x_i(k))}{\sum_{(i,j)\in D} w_{(i,j)}(k)\tau_{(i,j)}(k)} - 1\right). \tag{83}$$

The minimum value of τ is 0 %, which corresponds to the case where all vessels spend the minimum amount of time possible on all arcs of their route.

6.2. Test 2: Vessels with multiple travel directions and intermediate destinations

The test considers a more general setting, whereby thirteen vessels of different sizes departing at different time instants move in either direction. Moreover, vessels can now start and end their journeys at any arbitrary node. All vessels have the unweighted minimum travel time objective except for vessel 7, which has a planned arrival time. This information is summarized in Table 6.

The thirteen populated lockages of the solution are shown in Fig. 12. It can be seen that all vessels correctly fit into their lockages, properly mooring to other vessels when necessary. The schedule in Fig. 13 shows that all vessels have a consistent route regardless of their departure and destination nodes. The consideration of mixed objectives works as expected, as vessel 7 arrives at its destination exactly on time, with all other vessels still being scheduled to be as fast as possible.

A baseline approach that is representative of the current state of practice is used for comparison. This time-based simulation considers the same key operational constraints and updates the system components accordingly at every time interval dt. Moreover, the following considerations apply (Government, 2017):

- Vessels are assigned to lock chambers on a first-come, first-served (FCFS) basis.
- Smaller vessels may overtake larger vessels that arrived earlier if the larger vessels do not fit into a lockage, but the smaller vessels do.
- Only vessels in the waiting area are considered for placement in the chambers.

Table 3
Arc parameters.

Arc (i,j)	$d_{(i,j)}$ $[km]$	Specified $\tau_{(i,j)}(k)\,[h]$, $\forall k \in \mathcal{K}$	$N_{\chi,(i,j)}$	$N_{\gamma,(i,j)}\,[m]$	$\overline{\tau}_{(i,j)}\,[h]$
$(0,1),(1,0),(6,7),(7,6),\ (7,8),(8,7),(11,12),(12,11)$	Euclidean	-	-	-	-
Lock A Waiting Areas $(1,2),(2,1),(1,4),(4,1),\ (3,6),(6,3),(5,6),(6,5)$	-	0.1	-	-	-
Lock A Chamber I $(2,3),(3,2)$	-	0.5	3	150	0.1
Lock A Chamber II $(4,5),(5,4)$	-	0.75	2	200	0.1
Lock B Waiting Areas $(8,9),(9,8),(10,11),(11,10)$	-	0.1	-	-	-
Lock B Chamber I $(9,10),(10,9)$	-	1	4	250	0.1

Table 4
Test 1: vessel parameters.

Vessel k	$v_{\max,(i,j)}(k)\,[km/h]$, $\forall (i,j) \in \mathcal{D}$	$b(k)$	$d(k)$	$u(k)\,[h]$	W_χ	$W_\gamma\,[m]$	▽	▽▽
0	10	0	12	0	1	150	✓	
1	10	0	12	0	2	175		
2	10	0	12	0	1	75		✓
3	10	0	12	0	2	200		
4	10	0	12	0	2	100		
5	10	0	12	0	1	125		
6	10	0	12	0	1	100		
7	10	0	12	0	1	100		

Table 5
KPIs for first test.

KPI	Value
$J_{\mathcal{A}}$	70.4 h
$J_{\mathcal{H}}$	-
$\mathcal{L}$	15
$\mathcal{T}$	25.72 %

Table 6
Test 2: vessel parameters.

Vessel k	$v_{\max,(i,j)}(k)\,[km/h]$, $\forall (i,j) \in \mathcal{D}$	$b(k)$	$d(k)$	$u(k)\,[h]$	W_χ	$W_\gamma\,[m]$	$\hat{x}_{d(k)}\,[h]$
0	8	0	12	0	1	80	-
1	12	0	12	0.5	2	200	-
2	10	0	7	1	1	150	-
3	10	7	12	2	1	175	-
4	11	0	12	0.4	1	100	-
5	15	7	12	1	1	100	-
6	12	0	12	0	1	150	-
7	15	12	0	0	1	130	10
8	11	12	0	0.5	1	100	-
9	12	12	0	1	1	120	-
10	9	12	7	0.7	2	90	-
11	14	7	0	3	1	150	-
12	13	12	0	0	1	140	-

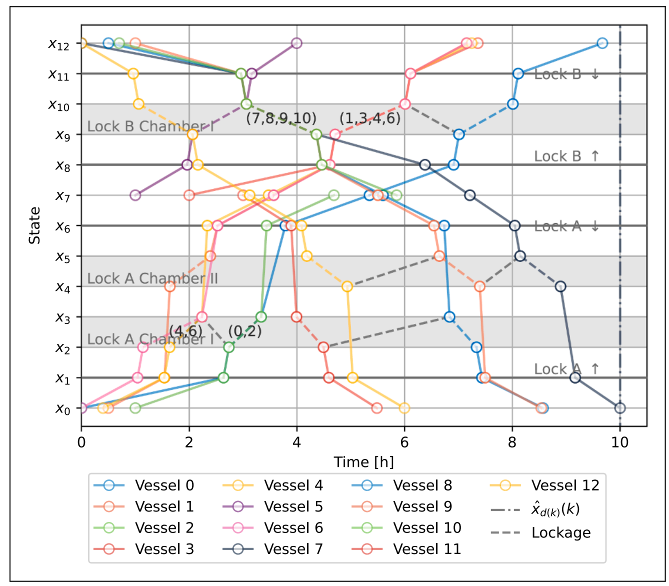

Fig. 13. Journey plots for second test.

Table 7
KPIs for second test.

Proposed			Baseline			Δ (Proposed - Baseline)		
J	$\mathcal{L}$	$\mathcal{T}$ [%]	J	$\mathcal{L}$	$\mathcal{T}$ [%]	J	$\mathcal{L}$	$\mathcal{T}$ [%]
77.50	16	15.64	79.91	20	16.37	-2.41	-4	-0.73

- Vessels in the waiting area of a lock are assigned to the chamber with the lowest processing time among the available chambers.
- Vessels travel at their maximum velocity on all arcs.
- Vessels are placed inside lock chambers according to a bin-based heuristic that obeys mooring constraints.
- All vessels have the minimum arrival time objective.
- A lockage with a set start time that has not started yet can be delayed to include a new vessel if a vessel that will fit in the lockage arrives at the waiting area.

The baseline approach is run with a time interval $dt = 10^{-4}$ h (such a small timestep is used to reduce the effect of discretization errors in the simulation). The numerical values of the KPIs for both the proposed and the baseline approaches are summarized in Table 7, together with the quantitative improvement (lower KPI values are preferable). The proposed solution yields better results, as vessels can be given priority at one lock so they can be grouped on a subsequent lockage to save time. In contrast, the baseline approach does not have the information to plan ahead. This feature suggests that the proposed approach will outperform the current state of practice for more complex graph topologies with multiple locks, though this should be confirmed by further testing.

7. Conclusions and future research

This paper contributes to the literature on lock scheduling with a solution based on switching max-plus algebra, which can be used to schedule vessel navigation through IWT networks (which may be characterized by any possible network topology) that require to pass multi-chamber locks. An arc-based route construction mechanism for vessel route choice based on the shortest path problem is introduced in SMPL algebra. Synchronization of vessels on lock chambers is enabled by incorporating two-dimensional ship placement capabilities that obey mooring constraints while also observing special rules for vessels carrying flammable or explosive goods. The resulting SMPL system is converted to an MILP problem, and the main operational objectives are defined. The performance of the proposed approach is compared to a baseline solution using relevant KPIs, allowing to demonstrate the superior performance of the former.

This paper has demonstrated a proof of concept, showing the potential of the approach for real IWT management. However, several assumptions that have been made in this paper hinder the applicability of the proposed approach in real environments. These assumptions are discussed hereunder, together with future research directions.

The proposed scheduling solution is essentially an offline (open-loop) optimization approach. This means that the schedule is produced under the assumption that all future information is known before the optimization process starts. Vessel and lock failures, vessels entering the system in the middle of the schedule, or vessels simply not complying with the solution can cause disruptions and delays, hence requiring to determine a new schedule with information that only becomes available as time progresses. Therefore, in order to better capture real-world conditions, an online (closed-loop) optimization strategy that is able to exploit system evolution will be designed.

Online optimization implementations for real environments require, as opposed to simulation environments, real-time feasibility. This concept represents the possibility to determine the solution to the optimization problem within two consecutive discrete-time instants (in a discrete-event systems sense). However, results have shown that scenarios with more than 10–15 vessels lead to unacceptable high computation times. National or regional scheduling systems may have to schedule hundreds of vessels simultaneously. To reduce computation burden, research efforts will be devoted to the development of distributed optimization approaches on the basis of the solution presented in this paper. The review of methods provided in Yang et al. (2019) constitutes a promising starting point.

Vessel data such as departure times, scheduled arrival times at the destination, and maximum velocity are assumed to be completely available to the scheduler. However, this is seldom the case in practice due to, e.g., privacy concerns and communication constraints. Future research should relax this assumption, contributing with a solution that can handle uncertainty. Possibilities that could be explored include the use of probabilistic models and the development of stochastic optimization algorithms. To this end, relevant methodologies reviewed in Powell (2019) will be considered.

CRediT authorship contribution statement

Pablo Segovia: Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Raphael Ummels:** Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Ton van den Boom:** Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Vasso Reppa:** Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Given her role as Deputy Editor, Vasso Reppa (co-author of this paper) had no involvement in the peer review of this article and had no access to information regarding its peer review. Full responsibility for the editorial process for this article was delegated to another journal editor. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Researchlab Autonomous Shipping (RAS) of Delft University of Technology.

References

Agency, E.E., 2021. Rail and waterborne best for low-carbon motorised transport. https://www.eea.europa.eu/publications/rail-and-waterborne-transport. https://doi.org/10.2800/85117

Bartels, L.B., Dambruin, S.A., van der Nagel, H.M., Terwindt, S.W.J., 2021. The relation between the computer game Tetris and max-plus scheduling (of industrial and logistic systems). Bsc in mechanical engineering (end project). Delft University of Technology. Delft, The Netherlands.

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

van den Boom, T., van den Muijsenberg, M., de Schutter, B., 2020. Model predictive scheduling of semi-cyclic discrete-event systems using switching max-plus linear models and dynamic graphs. Discrete Event Dyn. Syst.: Theory Appl. 30 4, 635–669. https://doi.org/10.1007/s10626-020-00318-w

Cohen, G., Gaubert, S., Quadrat, J.P., (1999). Max-plus algebra and system theory: where we are and where to go now. Annu. Rev. Control 23, 207–219. https://doi.org/10.1016/S1367-5788(99)90091-3

Dong, S., Yang, G., Liao, S., Li, L., (2025). Optimizing ship traffic scheduling in LNG ports: an enhanced model incorporating navigation characteristics and night sailing risks. Ocean Eng. 330, 121117. https://doi.org/10.1016/j.oceaneng.2025.121117

Commission, E., 2001. European transport policy for 2010: time to decide. European Commission Brussels COM(2001) 370.

Eurostat, 2022. Inland waterways freight transport - quarterly and annual data. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Inland_waterways_freight_transport_-_quarterly_and_annual_data. Accessed on: 13-12-2022.

Garrisi, G., Cervelló-Pastor, C., 2020. Train-Scheduling optimization model for railway networks with multiplatform stations. Sustainability 12 (1). https://doi.org/10.3390/su12010257

Government, D., 2017. Binnenvaartpolitiereglement. https://wetten.overheid.nl/BWBR0003628/2017-01-01. Accessed: 2024-12-16.

Heidergott, B., Olsder, G.J., Van Der Woude, J., 2014. Max Plus at work: modeling and analysis of synchronized systems: a course on Max-Plus algebra and its applications. Vol. 48. Princeton University Press.

Hermans, J., 2014. Optimization of inland shipping - a polynomial time algorithm for the single ship single lock optimization problem. J. Schedul. 17, 305–319. https://doi.org/10.1007/s10951-013-0364-7

Ji, B., Sun, H., Yuan, X., Yuan, Y., Wang, X., 2020a. Coordinated optimized scheduling of locks and transshipment in inland waterway transportation using binary NSGA-II. Int. Trans. Oper. Res. 27, 1501–1525. https://doi.org/10.1111/itor.12720

Ji, B., Yuan, X., Yuan, Y., 2017. Orthogonal design-Based NSGA-III for the optimal lockage co-Scheduling problem. IEEE Trans. Intell. Transp. Syst. 18 (8), 2085–2095. https://doi.org/10.1109/TITS.2016.2630718

Ji, B., Yuan, X., Yuan, Y., 2019. A binary borg-Based heuristic method for solving a multi-Objective lock and transshipment co-Scheduling problem. IEEE Trans. Intell. Transp. Syst. 20 (3), 947–958. https://doi.org/10.1109/TITS.2018.2841022

Ji, B., Yuan, X., Yuan, Y., Lei, X., Iu, H.H.C., 2020b. An adaptive large neighborhood search for solving generalized lock scheduling problem: comparative study with exact methods. IEEE Trans. Intell. Transp. Syst. 21, 3344–3356. https://doi.org/10.1109/TITS.2019.2926405

Ji, B., Zhang, D., Yu, S.S., Fang, X., 2021a. An exact approach to the generalized serial-lock scheduling problem from a flexible job-shop scheduling perspective. Comput. Oper. Res. 127, 105164. https://doi.org/10.1016/j.cor.2020.105164

Ji, B., Zhang, D., Yu, S.S., Zhang, B., 2021b. Optimally solving the generalized serial-lock scheduling problem from a graph-theory-based multi-commodity network perspective. Eur. J. Oper. Res. 288, 47–62. https://doi.org/10.1016/j.ejor.2020.05.035

Kersbergen, B., 2015. Modeling and Control of Switching Max-Plus-Linear Systems: Rescheduling of railway traffic and changing gaits in legged locomotion. Ph.D. thesis. TU Delft. Delft, The Netherlands.

Kersbergen, B., Rudan, J., van den Boom, T., De Schutter, B., 2014. Towards railway traffic management using switching max-plus-linear systems: structure analysis and rescheduling. Discrete Event Dyn. Syst.: Theory Appl. 26 (2), 183–223. https://doi.org/10.1007/s10626-014-0205-7

Kougiatsos, N., Scheffers, E.L., van Benten, M.C., Schott, D.L., de Vos, P., Negenborn, R.R., Reppa, V., 2025. An intelligent agent-based resilient framework for marine vessel mission adaptations. IEEE Open J. Intell. Transp. Syst. 6, 184–203. https://doi.org/10.1109/OJITS.2025.3539419

Lan, J., Zheng, M., Chu, X., Liu, C., Ding, S., 2024. A ship high-precision positioning method in the lock chamber based on liDAR. Ocean Eng. 306, 118033. https://doi.org/10.1016/j.oceaneng.2024.118033

Liu, C., Qi, J., Chu, X., Zheng, M., He, W., 2021. Cooperative ship formation system and control methods in the ship lock waterway. Ocean Eng. 226, 108826. https://doi.org/10.1016/j.oceaneng.2021.108826

Liu, S., Zhang, Y., Guo, W., Wang, W., Zheng, Q., Yu, H., 2025. Ship appointment scheduling for lockage operations of waterway transport with non-punctual arrivals. Ocean Eng. 315, 119844.https://doi.org/10.1016/j.oceaneng.2024.119844

Passchyn, W., Briskorn, D., Spieksma, F.C.R., 2016a. Mathematical programming models for lock scheduling with an emission objective. Eur. J. Oper. Res. 246, 802–814. https://doi.org/10.1016/j.ejor.2015.09.012

Passchyn, W., Coene, S., Briskorn, D., Hurink, J.L., Spieksma, F. C.R., Vanden Berghe, G., 2016b. The lockmaster's problem. Eur. J. Oper. Res. 251 (2), 432–441. https://doi.org/10.1016/j.ejor.2015.12.007

Pour, F.K., Segovia, P., Duviella, E., Puig, V., 2022. A two-layer control architecture for operational management and hydroelectricity production maximization in inland waterways using model predictive control. Control Eng. Pract. 124. https://doi.org/10.1016/j.conengprac.2022.105172

Powell, W.B., 2019. A unified framework for stochastic optimization. Eur. J. Oper. Res. 275 (3), 795–821.

Prandtstetter, M., Ritzinger, U., Schmidt, P., Ruthmair, M., 2015. A Variable Neighborhood Search Approach for the Interdependent Lock Scheduling Problem. Springer, Switzerland. pp. 36–60. https://doi.org/10.1007/978-3-319-16468-7_4

Segovia, P., Pesselse, M., van den Boom, T., Reppa, V., 2022. Scheduling inland waterway transport vessels and locks using a switching max-plus-linear systems approach. IEEE Open J. Intell. Transp. Syst. 3, 748–762. https://doi.org/10.1109/OJITS.2022.3218334

Segovia, P., Puig, V., Negenborn, R.R., Reppa, V., 2025. Dynamic coordination of multiple movable bridges and vessels for time-efficient inland waterway navigation. IEEE Trans. Intell. Transp. Syst. 26 (4), 5481–5493. https://doi.org/10.1109/TITS.2025.3528184

Soveges, E., Stefanov, M., Puricella, P., Wieser, C., Ivanova, I., Must, E., Grassi, E., Hai, D., Murphy, T., 2015. Inland Waterway Transport in Europe: No significant improvements in modal share and navigability conditions since 2001. European Court of Auditors. Luxembourg. https://doi.org/10.2865/158305

Trespalacios, F., Grossmann, I.E., 2015. Improved big-M reformulation for generalized disjunctive programs. Comput. Chem. Eng. 76, 98–103.

Verstichel, J., De Causmaecker, P., Spieksma, F., Vanden Berghe, G., 2014a. The generalized lock scheduling problem: an exact approach. Transp. Res. Part E: Logistic. Transp. Rev. 65, 16–34. https://doi.org/10.1016/j.tre.2013.12.010

Verstichel, J., De Causmaecker, P., Spieksma, F.C.R., Vanden Berghe, G., 2014b. Exact and heuristic methods for placing ships in locks. Eur. J. Oper. Res. 235, 387–398. https://doi.org/10.1016/j.ejor.2013.06.045

Verstichel, J., De Causmaecker, P., Vanden Berghe, G., 2011. Scheduling algorithms for the lock scheduling problem. Procedia - Soc. Behav. Sci. 20, 806–815. https://doi.org/10.1016/j.sbspro.2011.08.089

Verstichel, J., Kinable, J., De Causmaecker, P., Vanden Berghe, G., 2015. A combinatorial Benders' decomposition for the lock scheduling problem. Comput. Oper. Res. 54, 117–128. https://doi.org/10.1016/j.cor.2014.09.007

Vielma, J.P., 2015. Mixed integer linear programming formulation techniques. SIAM Rev. 57 (1), 3–57. https://doi.org/10.1137/130915303

Wang, X., Zhao, Y., Sun, P., Wang, X., 2013. An analysis on convergence of data-driven approach to ship-lock scheduling. Math. Comput. Simul. 88, 31–38. https://doi.org/10.1016/j.matcom.2013.03.005

Yang, T., Yi, X., Wu, J., Yuan, Y., Wu, D., Meng, Z., Hong, Y., Wang, H., Lin, Z., Johansson, K.H., 2019. A survey of distributed optimization. Annu. Rev. Control 47, 278–305.

Yuan, X., Ji, B., Wu, X., Zhang, X., 2016. Co-scheduling of lock and water-land transshipment for ships passing the dam. Appl. Soft Comput. 45, 150–162. https://doi.org/10.1016/j.asoc.2016.04.019

Zhang, Y., Liu, S., Zheng, Q., Tian, H., Guo, W., 2024. Ship scheduling problem in an anchorage-to-quay channel with water discharge restrictions. Ocean Eng. 309, 118432. https://doi.org/10.1016/j.oceaneng.2024.118432

Zhang, Y., Zheng, Q., He, L., Tian, H., 2023. Ship traffic optimization method for solving the approach channel and lock co-scheduling problem of the three gorges dam on the yangzi river. Ocean Eng. 276, 114196. https://doi.org/10.1016/j.oceaneng.2023.114196

Zhao, X., Lin, Q., Yu, H., 2020. A co-scheduling problem of ship lift and ship lock at the three gorges dam. IEEE Access 8, 132893–132910. https://doi.org/10.1109/ACCESS.2020.3009775