

## Scheduling Inland Waterway Transport Vessels and Locks Using a Switching Max-Plus-Linear Systems Approach

Segovia, Pablo; Pesselse, Mike; Van Den Boom, Ton; Reppa, Vasso

**DOI**

[10.1109/OJITS.2022.3218334](https://doi.org/10.1109/OJITS.2022.3218334)

**Publication date**

2022

**Document Version**

Final published version

**Published in**

IEEE Open Journal of Intelligent Transportation Systems

**Citation (APA)**

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 Journal of Intelligent Transportation Systems*, 3, 748-762. <https://doi.org/10.1109/OJITS.2022.3218334>

**Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

# Scheduling Inland Waterway Transport Vessels and Locks Using a Switching Max-Plus-Linear Systems Approach

PABLO SEGOVIA<sup>1</sup>, MIKE PESSELSE<sup>2,3</sup>, TON VAN DEN BOOM<sup>2</sup>, AND VASSO REPPA<sup>1</sup>

<sup>1</sup>Department of Maritime and Transport Technology, Delft University of Technology, 2628 CD Delft, The Netherlands

<sup>2</sup>Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands

<sup>3</sup>Power2X, 1074 VJ Amsterdam, The Netherlands

CORRESPONDING AUTHOR: P. SEGOVIA (e-mail: p.segoviacastillo@tudelft.nl)

This work was supported in part by the Project "Novel Inland Waterway Transport Concepts for Moving Freight Effectively (NOVIMOVE)" and in part by the European Union's Horizon 2020 Research and Innovation Programme under Grant 858508.

**ABSTRACT** This paper considers the inland waterborne transport (IWT) problem, and presents a scheduling approach for inland vessels and locks to generate optimal vessel and lock timetables. The scheduling strategy is designed in the switching max-plus-linear (SMPL) systems framework, as these are characterized by a number of features that make them well suited to represent the IWT problem. In particular, the resulting model is linear in the max-plus algebra, and SMPL systems can switch between modes, an interesting feature due to the presence of vessel routing and ordering constraints in the model. Moreover, SMPL systems can be transformed into mixed-integer linear programming (MILP) problems, for which efficient solvers are available. Finally, a realistic case study is used to test the approach and assess its effectiveness.

**INDEX TERMS** Inland waterborne transport, intelligent transportation systems, vessel-to-infrastructure interaction, scheduling, max-plus algebra, switching max-plus-linear systems.

## I. INTRODUCTION

CONSUMER and producer distance has increased as a result of globalization. Large container vessels are nowadays sailing across the oceans, delivering goods and bulk from port to port across the globe. However, these vessels cannot reach inland destinations due to their size, and thus goods and bulk must be transferred to various inland transportation methods, e.g., inland vessels, trucks and trains, for hinterland transportation.

Institutions such as the European Commission [1], [2] and the Port of Rotterdam [3] are heavily promoting a modal shift towards inland waterborne transport (IWT) in order to make better use of this natural network of inland waterways. The main reasons for this shift are cost efficiency [4], environmental friendliness [5], [6], high degree of safety [7] and

reduction of road congestion [8]. However, implications of this trend can already be noticed, and especially on locks, which are the main IWT bottlenecks. An example of this is the freight moved on the Yangtze river (China), which has increased more than expected. In particular, the Three Gorges lock reached its maximum capacity of 100 million tons in 2011 [9], which is 19 years earlier than predicted. This has made the Three Gorges lock a bottleneck on the Yangtze river, limiting further IWT development. This situation is something that other inland waterways are slowly running into as well.

Increasing lock capacity can help mitigate this situation. However, this entails high operational costs, and expansion is not always possible due to environmental or spatial reasons. Another possible approach is that of scheduling lock operations in an optimal manner to maximize vessel passage efficiency, which can have a significant advantage in moving freight from seaports to the hinterland without the high

The review of this article was arranged by Associate Editor Edwin van Hassel.

investment costs of building additional infrastructure. This strategy, whereby vessel-to-infrastructure (V2I) interaction is introduced to coordinate the operations, transforms the original management problem into an optimization problem. Its resolution determines feasible decisions that maximize a certain profit function, which is usually linked to minimizing passage times. As vessels passing through locks occupy resources (namely space and time slots), the main challenge is the optimal allocation of these resources.

Locks can consist of one or multiple chambers, each characterized by a specific capacity and operation duration. Moreover, the latter depends on both the water levels and the number and type of vessels present during a particular operation. The lock scheduling problem then consists in minimizing waiting times of vessels passing through the lock while meeting capacity and duration constraints. This problem has received considerable attention from the research community, and different methodologies have been devised. An integrated approach to the generalized lock scheduling problem is discussed in [10]. Vessel passage through locks requires to solve two problems: determine vessel position inside the chamber and decide starting times of lock operations.

The vessel placement problem (VPP) is concerned with minimizing the number of lock operations needed to move all vessels to the other side of the lock. This is achieved by first deciding the vessel sequence, which is based on the service policy. Choice of service policy is examined in [11], [12], and the shortest-processing-time-first policy is shown to result in less overall network delay than first-come-first-serve. Once the vessel sequence has been decided, vessels must be arranged within lock chambers while observing capacity and safety constraints. This problem is recast to a two-dimensional bin packaging problem in [13], and is solved to minimize the total operation time. The same authors present a model for the chamber-arranging problem in [14], and compare the performances offered by an exact decomposition method and a multi-order best fit heuristic method.

Once a solution has been determined for the VPP, the second step consists in solving the lock scheduling problem (LSP), whereby the optimal operation plan—time instants at which lock movements must be performed—must be determined to minimize the sum of waiting times of individual vessels [15]. A so called late-acceptance algorithm for the LSP that simultaneously considers waiting times and water usage in lock operations is proposed in [16]. Research focusing on the vessel order sequence is tackled in [17] by making an analogy to the identical parallel machine scheduling problem with sequence-dependent setup times and release dates.

The previous references focus on the scheduling of individual locks. On the other hand, the problem of scheduling inland vessels through locks considering larger portions of a waterway network has only started receiving increased attention in the last few years. The single-chamber serial

lock scheduling scenario is examined in [18], while the multiple-chamber case is studied in [19], [20]. A solution to the problem combining a decision-making method to perform infrastructural modifications and dynamic lock scheduling is proposed in [21]. General serial-lock scheduling solutions using flexible job-shop scheduling and a two-dimensional bin-packing problem on the one hand, and a multi-commodity network perspective on the other hand, are presented in [22] and [23], respectively. Arrival sequence of vessels at locks, lockage operation and service time are jointly considered in [24] using a mixed-integer programming (MIP) model, which is solved with large neighborhood search-based heuristics.

This paper presents the design of a scheduling strategy, whereby multiple IWT vessels sail through a waterway network and pass multiple locks. The main difference with respect to other papers on serial lock scheduling is the fact that the discrete-event systems (DES) framework is employed here for the first time to model the IWT system, as the dynamics are governed by the evolution of events, and not time. Moreover, a switching max-plus-linear (SMPL) system modeling approach is adopted, as it allows to capture the characteristics of IWT operation of both locks and vessels, and also to introduce changes in the order of events. Then, the optimization of lock and vessel operation is carried out following a mixed-integer linear programming (MILP) approach.

Contributions of this work with respect to the current state of the art are described below:

- Only a limited number of DES have been described as SMPL systems, such as production scheduling systems [25], [26], [27], railway traffic systems [25], [28], [29] and legged locomotion [30]. Conversely, modeling the IWT as an SMPL system is a novel approach to the best knowledge of the authors, and allows to address two problems simultaneously: choosing the route a vessel takes and deciding the lock entering order.
- The vessel-lock interplay can be generalized to a production system in which products arrive from opposite directions and compete for the same resource. However, the issue of uneven water levels upstream and downstream of the lock introduces a particular feature: lock operations require water level inside the lock to match that of the passing vessel. This fact is taken into account in the derivation of the approach.
- The solution presented in this paper allows vessels to overtake each other during navigation to exploit differences in terms of size, maximum speed and processing times at locks. To the best knowledge of the authors, there is no previous research on SMPL systems where jobs are allowed to overtake each other.
- Although this paper addresses the particular application of IWT, the methodology is presented in such way that it can be applied to any system characterized by an analogous formulation.

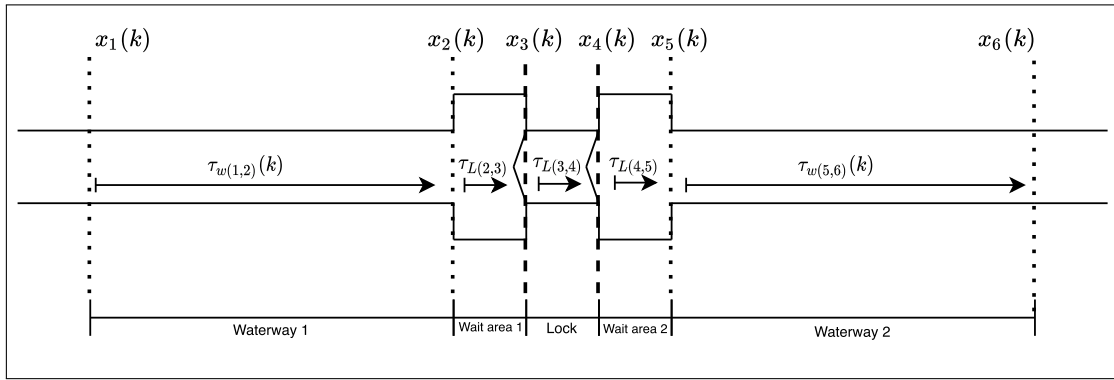


FIGURE 1. Schematic top view of two waterways connected by a lock.

The rest of the paper is organized as follows: the IWT scheduling problem for locks and vessels is described in Section II. The proposed scheduler architecture is detailed in Section III, respectively. Section IV provides the complete derivation of modeling and scheduling approaches. A realistic case study is presented in Section V, together with the simulation results and discussion. Conclusion are drawn and future research directions are outlined in Section VI.

## II. PROBLEM STATEMENT

The problem considered in this paper is that of designing a scheduler for vessels and locks that minimizes cumulative arrival times of vessels at destinations, and can be formulated as follows. A set of vessels  $\mathcal{V} = \{1, \dots, N_v\}$  sailing through an inland waterway network must pass a set of locks  $\mathcal{L} = \{1, \dots, N_l\}$  on their way from origin to destination. These locks divide the overall network into different waterways, and are equipped with waiting areas so that vessels can wait until permission to enter locks is granted. Moreover,  $\mathcal{V} = \mathcal{V}_d \cup \mathcal{V}_u$ , where  $\mathcal{V}_d$  and  $\mathcal{V}_u$  denote vessels traveling downstream (from inland towards the sea) and upstream (from the sea towards inland), respectively, and  $\mathcal{V}_u \cap \mathcal{V}_d = \emptyset$ .

The following inland waterway and vessel information are inputs to the scheduler:

- Inland waterway: network topology matrix, locations of locks and endpoints, and lock processing times.
- Vessels: total number of vessels sailing the network, sailing times on all waterways, departure locations and times, arrival locations and deadlines, and relative priority of vessels (it might be critical for certain vessels to arrive at the destination faster than others depending on, e.g., cargo type).

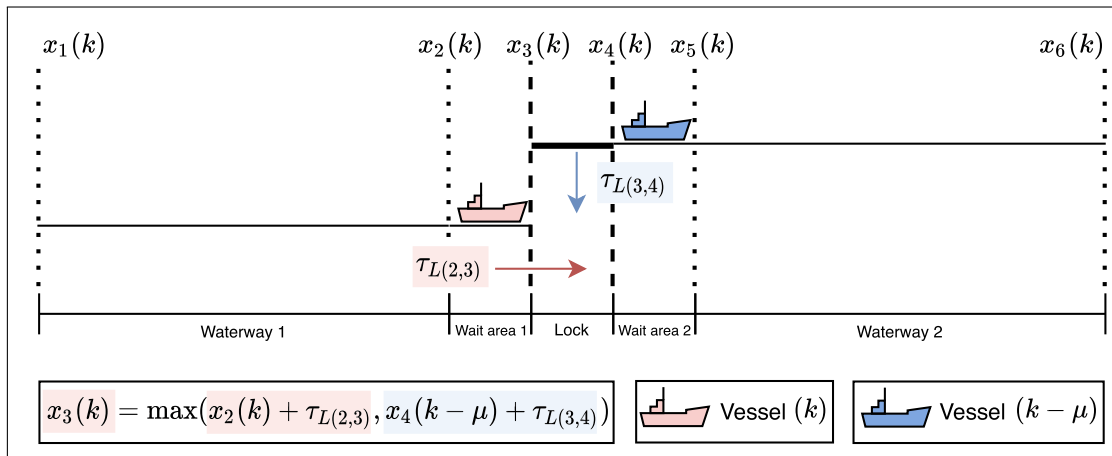
The assumptions of the problem are given below:

- All vessels navigate between the same two endpoints, from end to end.
- Bidirectional traffic between the endpoints is assumed.
- There exists more than one waterway connecting the endpoints.
- Vessels and locks are fully compliant with the solution of the scheduler.

- All locks consist of a single chamber, which can only fit a single vessel.
- Lock operations start as soon as a vessel enters the lock.
- Lock operation durations are only assumed to depend on the lock, and not on the vessel that is processed.
- Sailing times of individual vessels for individual waterways are assumed to be known beforehand.
- Vessels maintain a constant speed while sailing through a waterway.
- Vessels can overtake other vessels in all waterways and waiting areas.
- Departure times of vessels at the origin cannot be decided by the scheduler, but they are given.
- Upper bounds on the maximum capacity of the waiting areas are not considered.

System dynamics depend on the evolution of events. On the one hand, vessels sail through the waterways, and arrive at, move inside and exit a lock, after which they continue their journey along the adjacent waterway. On the other hand, locks can remain idle or process vessels, which requires to open, close, carry out leveling operations and open again to allow vessels resume navigation. Therefore, the DES framework can be conveniently employed to model optimization problems characterized by event-driven dynamics. In other words, state evolution depends entirely on the occurrence of asynchronous discrete events over time [31]. DES scheduling for IWT is then concerned with vessel routing and ordering decisions, i.e., determine the subset of waterways to be sailed and the relative passing order through each lock for each vessel, respectively. Occurrence of these events, and not time progression, causes the system to evolve to a different state.

As the operations of vessels and locks must be coordinated, event synchronization must be ensured. The example depicted in Figure 1 is introduced for clarity, and its associated state event variables and parameters are defined in Table 1. Consider that vessel  $k$  enters the system through waterway 1 and arrives at waiting area 1: in order for vessel  $k$  to move into the lock, it must sail through the waiting area first. This is only possible once the lock has processed the current vessel inside the lock (denoted with



**FIGURE 2.** Schematic side view of two waterways connected by a lock, with red vessel  $k$  and blue vessel  $k - \mu$  sailing through the network. Red vessel  $k$  has to wait for blue vessel  $k - \mu$  until it can enter the lock.

**TABLE 1.** Notation.

State event variable	Description
$x_1(k)$	Time instant when vessel $k$ enters waterway 1
$x_2(k)$	Time instant when vessel $k$ enters waiting area 1 and leaves waterway 1
$x_3(k)$	Time instant when vessel $k$ enters the lock and leaves waiting area 1
$x_4(k)$	Time instant when vessel $k$ enters waiting area 2 and leaves the lock
$x_5(k)$	Time instant when vessel $k$ enters waterway 2 and leaves waiting area 2
$x_6(k)$	Time instant when vessel $k$ leaves waterway 2
Parameter	Description
$\tau_{w(1,2)}(k)$	Sailing time of vessel $k$ from the start of waterway 1 to the end of waterway 1
$\tau_{L(2,3)}$	Operation time from the start of waiting area 1 to the end of waiting area 1
$\tau_{L(3,4)}$	Lock operation (i.e. drainage) time to level the water to the other side of the lock
$\tau_{L(4,5)}$	Operation time from the start of waiting area 2 to the end of waiting area 2
$\tau_{w(5,6)}(k)$	Sailing time of vessel $k$ from the start of waterway 2 to the end of waterway 2

$k - \mu$ ), and the water level inside the lock matches the water level of vessel  $k$ . This situation is illustrated in more detail in Figure 2.

*Remark 1:* As vessel overtaking is possible, vessel  $k - 1$  (which departed immediately before vessel  $k$ ) may not always be right in front of vessel  $k$ . Parameter  $\mu$  is introduced to account for the fact that relative positions of vessels may change during system evolution. Moreover,  $\mu$  does not define a single scalar, but a set of natural scalars, i.e.,  $\mu \in \{\mu_{\min}, \dots, \mu_{\max}\}$ . Therefore, the current vessel inside the lock might be any vessel within the range defined by  $\mu$ .

The time instant at which vessel  $k$  enters the lock, i.e.,  $x_3(k)$ , is then equal to the maximum time of the previous events:

- Vessel  $k$  arrives at the waiting area and enters the lock:  $x_2(k) + \tau_{L(2,3)}$ .
- Vessel  $k - \mu$  exits the lock, which is then drained to match water level of vessel  $k$ :  $x_4(k - \mu) + \tau_{L(3,4)}$ .

This simple case can be modeled using the maximization operator, which is inherently nonlinear in conventional algebra [25]. However, max-plus-linear (MPL) systems are a class of DES that can be described by a model that is linear in the max-plus algebra, with maximization and addition as basic operations. Moreover, the use of MPL systems as

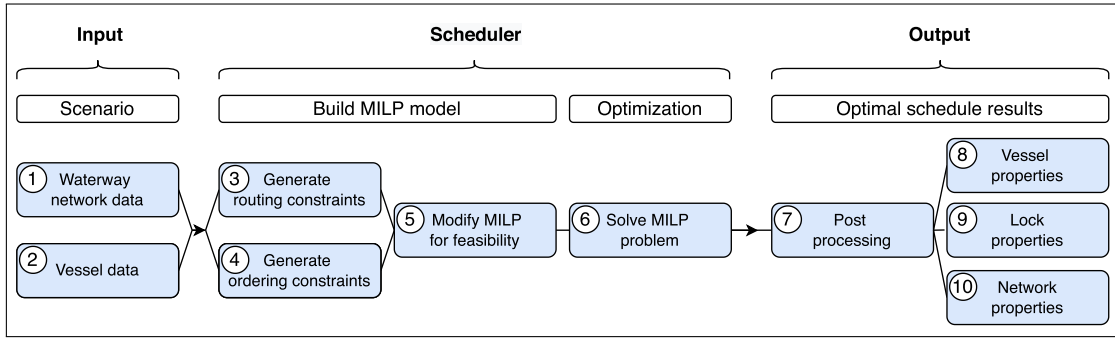
basic models for scheduling offers additional advantages. Several system-theoretical results are available in the literature [32], [33], and it is possible to find bottlenecks in the scheduling process as well as good initial scheduling values by using system tools, e.g., max-plus eigenvalues and eigenvectors [29].

Conventional MPL systems are characterized by a fixed model structure, and therefore system dynamics always remain the same and cannot be influenced. Conversely, the IWT system is not fixed, and the system dynamics can change due to, e.g., vessels taking different routes in the network and overtaking other vessels. There exists a class of MPL systems that is endowed with the capability to switch between different modes: switching max-plus-linear (SMPL) systems.

Therefore, the proposed solution to address operational inefficiencies in IWT consists in designing a scheduler based on an SMPL model of the IWT system. The overall scheduler architecture is discussed next, after which the modeling and scheduling approaches will be presented.

### III. SCHEDULER ARCHITECTURE

The overall architecture is depicted in Figure 3, and consists of three parts: input, scheduler and output.


**FIGURE 3.** Schematic overview of the scheduler architecture.

**TABLE 2.** Scheduler input.

① Waterway network data	② Vessel data
(i) Topology matrix, $\mathbf{T}$	(i) Number of vessels sailing, $N_v$
(ii) Lock locations	(ii) Sailing times on waterways, $\tau_{w(i,j)}(k)$
(iii) Lock processing times, $\tau_{L(i,j)}$	(iii) Departure location, $d(k)$
(iv) Departure locations	(iv) Departure time, $u_i(k)$
(v) Arrival locations	(v) Arrival deadlines, $a_i(k)$
	(vi) Vessel priority, $\sigma(k)$

### A. INPUT

Inland waterway and vessel data are provided as inputs to the scheduler. This information, which must be made available before the first vessel departs, is summarized in Table 2.

On the one hand, the inland waterway is characterized by:

- Topology matrix,  $\mathbf{T}$ , which describes the same information as the network graph. A waterway is represented by two nodes connected by one arc, while a lock consists of four nodes (to represent all positions of interest) connected by three arcs.
- Processing time of the lock that connects nodes  $i$  and  $j$ ,  $\tau_{L(i,j)}$ . This information is expressed in time units.
- Positions of locks and endpoints. It is recalled that all vessels are assumed to sail from one endpoint to the other, each vessel in one of the two possible directions.

On the other hand, vessel  $k$  is characterized by:

- Sailing time along the waterway delimited between nodes  $i$  and  $j$ , which is denoted with  $\tau_{w(i,j)}(k)$ . This information is assumed to be known beforehand, and is expressed in time units.
- Departure location,  $d(k)$ . Note that this implicitly defines the sailing direction and the arrival location.
- Departure time (expressed in time units),  $u_i(k)$ . It is recalled that this time cannot be controlled.
- Arrival deadlines (expressed in time units),  $a_i(k)$ . These are passed as hard constraints to the scheduler.
- Vessel priority,  $\sigma(k)$ . Setting a higher or lower numerical value of each entry of  $\sigma(k)$ , i.e.,  $\sigma_i(k)$ , leads to more or less effort from the scheduler in optimizing event  $i$  of vessel  $k$ , i.e.,  $x_i(k)$ .

### B. SCHEDULER

As mentioned in Section II, scheduling for IWT systems aims to determine the route to be followed by each vessel and the order in which each vessel passes through each lock en route from its origin to its destination. This requires to generate routing and ordering constraints, respectively. The resulting model is modified for feasibility, and additional constraints are added, e.g., parametrization of the routing control variables, and ensuring both that only one route is chosen per vessel and vessels can only sail in one direction. With all this, the model is solved by making use of available efficient solvers.

The previous paragraph is only a brief summary—the design will be properly detailed in Section IV.

### C. OUTPUT

The scheduler returns a single vector containing the optimal solution, which must be post-processed to retrieve information of interest. The scheduler output can be split into vessel, lock and network properties.

*Remark 2:* It should be noted that the scheduler only returns a schedule if the problem is feasible for given waterway network and vessel data. If this is not the case, e.g., an arrival deadline is too strict, the scheduler will indicate that the problem is infeasible.

Information of interest for vessel  $k$  determined by the scheduler is indicated below:

- Arrival time at destination,  $x_a(k)$ .
- Arrival times at intermediate nodes,  $x_i(k)$ .
- Delays at lock  $l$ ,  $\Delta_l(k)$ . These are determined as the difference between the obtained optimal schedule (with the influence of other vessels) and the schedule that would result if the waterway was empty (save for vessel  $k$ ).
- Intermediate node departure times, which can be inferred from the optimal solution.

Information of interest for lock  $l$  determined by the scheduler is as follows:

- Optimal operation times, which indicate when the levelings must be realized.
- Number of total lock levelings performed by lock  $l$ ,  $\mathcal{L}_l$ .

**TABLE 3.** Scheduler output (\* denotes *inferred information from model output*).

⑧ Vessel schedule and properties	⑨ Lock schedule and properties	⑩ Network properties
(i) Optimal routes	(i) Optimal operation times for lock $l$	(i) Cumulative arrival time, $\mathcal{A}_{\text{net}}$
(ii) Destination arrival time, $x_a(k)$	(ii) Number of lock levelings, $\mathcal{L}_l$	(ii) Cumulative delay, $\mathcal{D}_{\text{net}}$
(iii) Intermediate node arrival times, $x_i(k)$	(iii) Number of empty lock levelings, $\mathcal{L}_{e,l}^*$	(iii) Make-span, $C_{\text{max}}$
(iv) Delays at lock $l$ for vessel $k$ , $\Delta_l(k)$	(iv) Occupancy of lock $l$ at time $t$ , $\mathcal{O}_l(t)^*$	(iv) Bottleneck, $\mathcal{B}_{\text{net}}$
(v) Intermediate node departure times*	(v) Queue order at lock $l$ at time $t$ , $\mathcal{Q}_l(t)^*$	

- Number of empty lock levelings,  $\mathcal{L}_{e,l}$ . This is defined as the number of lock operations when no vessel occupies the lock.
- Occupancy of lock  $l$  at time unit  $t$ ,  $\mathcal{O}_l(t)$ , which is a binary variable.
- Queue order at lock  $l$  at time unit  $t$ ,  $\mathcal{Q}_l(t)$ . This corresponds to the list of vessels that are waiting to be serviced by lock  $l$  at time  $t$ .

Several global waterway network properties can be computed from the scheduler solution, which allows to analyze IWT system performance:

- The cumulative arrival time  $\mathcal{A}_{\text{net}}$  is defined as the sum of all individual vessel destination arrival times  $x_a(k)$ :

$$\mathcal{A}_{\text{net}} = \sum_{k=1}^{N_v} x_a(k). \quad (1)$$

- The cumulative vessel delay  $\mathcal{D}_{\text{net}}$  is defined as the sum of all individual vessel delays at all locks:

$$\mathcal{D}_{\text{net}} = \sum_{l=1}^{N_l} \sum_{k=1}^{N_v} \Delta_l(k). \quad (2)$$

- The makespan  $C_{\text{max}}$  is defined as the largest time of completion of all jobs in the system:

$$C_{\text{max}} = \max \left( \{x_a(k), \dots, x_a(N_v)\} \right). \quad (3)$$

- The bottleneck  $\mathcal{B}_{\text{net}}$  is defined as the lock where most vessel delays occur. This is defined as the maximum of the sum of all vessel delays for each lock:

$$\mathcal{B}_{\text{net}} = \max \left( \left\{ \sum_{k=1}^{N_v} \Delta_1(k), \dots, \sum_{k=1}^{N_v} \Delta_{N_l}(k) \right\} \right). \quad (4)$$

Table 3 summarizes what the scheduler produces after the problem is solved.

#### IV. SMPL-BASED IWT SCHEDULING

This section presents the design of the scheduler, which constitutes the core of the architecture presented in Figure 3. The scheduler is designed as an SMPL system, as conventional MPL systems are not endowed with the switching capabilities required by IWT routing and ordering constraints. However, the resulting SMPL system must be transformed into an MILP problem to be used for scheduling.

The corresponding derivations are detailed in the following sections. Prior to that, some max-plus algebra definitions are provided.

#### A. SOME MAX-PLUS ALGEBRA DEFINITIONS

The basic max-plus operators oplus  $\oplus$  and otimes  $\otimes$ , also referred to as tropical addition and multiplication, respectively, are defined as

$$a \oplus b \stackrel{\text{def}}{=} \max(a, b) \quad \text{and} \quad a \otimes b \stackrel{\text{def}}{=} a + b. \quad (5)$$

Just as the zero-element 0 and unit-element 1 are the neutral elements in conventional algebra, the two neutral max-plus algebra elements are denoted with  $\epsilon$  and  $e$ , and are defined as

$$\epsilon \stackrel{\text{def}}{=} -\infty \quad \text{and} \quad e \stackrel{\text{def}}{=} 0. \quad (6)$$

Max-plus binary variables can take the max-plus zero  $\epsilon$  or the max-plus unit  $e$  values. Let  $v \in \{\epsilon, e\}$  be a max-plus binary variable, then the adjoint variable  $\bar{v}$  is defined as follows:

$$\bar{v} = \begin{cases} e, & \text{if } v = \epsilon, \\ \epsilon, & \text{if } v = e. \end{cases} \quad (7)$$

#### B. SMPL SYSTEMS MODELING

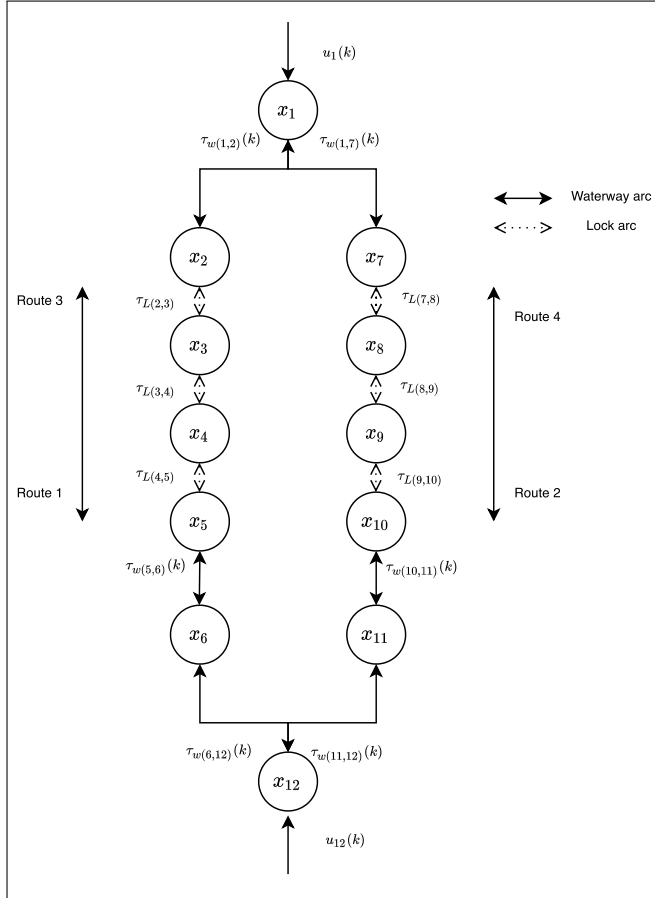
A suitable model must incorporate routing and ordering capabilities to reflect the real features of IWT. These concepts are defined as follows:

- *Routing* describes the route a vessel takes through the waterway network. If there is only one path between the two endpoints, the routing is fixed. However, this is not the case, as multiple paths were assumed to exist in Section II. Therefore, the routing is variable, and the scheduler determines the optimal route. Routing constraints must then be formulated to capture the dynamics of all routes.
- *Ordering* describes the order in which vessels pass through a lock. When several vessels arrive simultaneously at a lock, the scheduler determines the order of the queue in which vessels are allowed to enter the lock. This can be achieved by incorporating ordering constraints.

The schematic representation of the waterway network depicted in Figure 4 is used to illustrate the derivation of the SMPL model. This network comprises two corridors such as the one in Figure 1, i.e., each consisting of three waterways and a lock. Bidirectional traffic is allowed as indicated by the side arrows: vessels start at either the upstream node  $x_1$  or the downstream node  $x_{12}$ , and sail until reaching the opposite endpoint through either corridor, which leads to a variable routing situation. Description of variables and parameters introduced in Figure 4 is provided in Table 4.

**TABLE 4.** Variables and parameters introduced in Figure 4.

Variable or parameter	Description
$x_i(k)$	Variable; denotes time instant at which event $i$ happens for vessel $k$
$u_1(k), u_{12}(k)$	Parameter; denotes time instant at which vessel $k$ departs from $x_1(k)$ or $x_{12}(k)$ , respectively
$\tau_w(i,j)(k)$	Parameter; denotes constant sailing time of vessel $k$ on the waterway from node $i$ to node $j$
$\tau_L(i,j)$	Parameter; denotes time of lock operation from node $i$ to node $j$


**FIGURE 4.** Topology graph of a bidirectional IWT system with variable routing.

### 1) ROUTING CONSTRAINTS

Max-plus binary routing control variables must be introduced to activate only one of the routes for vessel  $k$ . The variables are denoted by  $s_{ij}(k)$ , and are defined as follows:

$$s_{ij}(k) = \begin{cases} e, & \text{if arc } (i,j) \text{ is used by vessel } k, \\ \epsilon, & \text{if arc } (i,j) \text{ is not used by vessel } k. \end{cases} \quad (8)$$

The previous definition requires to introduce a routing variable between each pair of nodes. However, the following relationship holds for route 1:  $s_{1,2}(k) = s_{2,3}(k) = s_{3,4}(k) = s_{4,5}(k) = s_{5,6}(k) = s_{6,12}(k)$ , i.e., if vessel  $k$  sails from node 1 to node 2, it must sail through the rest of nodes in that route. Then, a single routing variable can be used for the whole route 1, which is simply denoted with  $s_1(k)$ . Naturally, similar relationships can be derived for the rest of routes.

The four following cases can be distinguished, each with a different set of routing constraints:

- Vessel  $k$  sails *downstream* via left route 1,  $\forall k \in \mathcal{V}_d$ :

$$x_1(k) \geq u_1(k), \quad (9a)$$

$$x_2(k) \geq x_1(k) \otimes \tau_w(1,2)(k) \otimes s_1(k), \quad (9b)$$

$$x_3(k) \geq x_2(k) \otimes \tau_L(2,3) \otimes s_1(k), \quad (9c)$$

$$x_4(k) \geq x_3(k) \otimes \tau_L(3,4) \otimes s_1(k), \quad (9d)$$

$$x_5(k) \geq x_4(k) \otimes \tau_L(4,5) \otimes s_1(k), \quad (9e)$$

$$x_6(k) \geq x_5(k) \otimes \tau_w(5,6)(k) \otimes s_1(k), \quad (9f)$$

$$x_{12}(k) \geq x_6(k) \otimes \tau_w(6,12)(k) \otimes s_1(k). \quad (9g)$$

- Vessel  $k$  sails *downstream* via right route 2,  $\forall k \in \mathcal{V}_d$ :

$$x_1(k) \geq u_1(k), \quad (10a)$$

$$x_7(k) \geq x_1(k) \otimes \tau_w(1,7)(k) \otimes s_2(k), \quad (10b)$$

$$x_8(k) \geq x_7(k) \otimes \tau_L(7,8) \otimes s_2(k), \quad (10c)$$

$$x_9(k) \geq x_8(k) \otimes \tau_L(8,9) \otimes s_2(k), \quad (10d)$$

$$x_{10}(k) \geq x_9(k) \otimes \tau_L(9,10) \otimes s_2(k), \quad (10e)$$

$$x_{11}(k) \geq x_{10}(k) \otimes \tau_w(10,11)(k) \otimes s_2(k), \quad (10f)$$

$$x_{12}(k) \geq x_{11}(k) \otimes \tau_w(11,12)(k) \otimes s_2(k). \quad (10g)$$

- Vessel  $k$  sails *upstream* via left route 3,  $\forall k \in \mathcal{V}_u$ :

$$x_1(k) \geq x_2(k) \otimes \tau_w(1,2)(k) \otimes s_3(k), \quad (11a)$$

$$x_2(k) \geq x_3(k) \otimes \tau_L(2,3) \otimes s_3(k), \quad (11b)$$

$$x_3(k) \geq x_4(k) \otimes \tau_L(3,4) \otimes s_3(k), \quad (11c)$$

$$x_4(k) \geq x_5(k) \otimes \tau_L(4,5) \otimes s_3(k), \quad (11d)$$

$$x_5(k) \geq x_6(k) \otimes \tau_w(5,6)(k) \otimes s_3(k), \quad (11e)$$

$$x_6(k) \geq x_{12}(k) \otimes \tau_w(6,12)(k) \otimes s_3(k), \quad (11f)$$

$$x_{12}(k) \geq u_{12}(k). \quad (11g)$$

- Vessel  $k$  sails *upstream* via right route 4,  $\forall k \in \mathcal{V}_u$ :

$$x_1(k) \geq x_7(k) \otimes \tau_w(1,7)(k) \otimes s_4(k), \quad (12a)$$

$$x_7(k) \geq x_8(k) \otimes \tau_L(7,8) \otimes s_4(k), \quad (12b)$$

$$x_8(k) \geq x_9(k) \otimes \tau_L(8,9) \otimes s_4(k), \quad (12c)$$

$$x_9(k) \geq x_{10}(k) \otimes \tau_L(9,10) \otimes s_4(k), \quad (12d)$$

$$x_{10}(k) \geq x_{11}(k) \otimes \tau_w(10,11)(k) \otimes s_4(k), \quad (12e)$$

$$x_{11}(k) \geq x_{12}(k) \otimes \tau_w(11,12)(k) \otimes s_4(k), \quad (12f)$$

$$x_{12}(k) \geq u_{12}(k). \quad (12g)$$

Naturally, one and only one route must be selected for vessel  $k$ . This can be expressed as

$$\begin{aligned} & (s_1(k) \otimes \bar{s}_2(k) \otimes \bar{s}_3(k) \otimes \bar{s}_4(k)) \oplus \\ & (\bar{s}_1(k) \otimes s_2(k) \otimes \bar{s}_3(k) \otimes \bar{s}_4(k)) \oplus \\ & (\bar{s}_1(k) \otimes \bar{s}_2(k) \otimes s_3(k) \otimes \bar{s}_4(k)) \oplus \\ & (\bar{s}_1(k) \otimes \bar{s}_2(k) \otimes \bar{s}_3(k) \otimes s_4(k)) = 0. \end{aligned} \quad (13)$$



**TABLE 5.** Renamed max-plus binary ordering control variables and descriptions.

Variable	Description
$w_{DD,\mu}(k - \mu)$	Both vessel $k$ and $k - \mu$ sail downstream
$w_{UU,\mu}(k - \mu)$	Both vessel $k$ and $k - \mu$ sail upstream
$w_{DU,\mu}(k - \mu)$	Vessel $k$ sails downstream and vessel $k - \mu$ sails upstream
$w_{UD,\mu}(k - \mu)$	Vessel $k$ sails upstream and vessel $k - \mu$ sails downstream

## 2) ORDERING CONSTRAINTS

Multiple vessels are expected to pass through the same lock, which may lead to conflicts in the case of equal passing times. Therefore, ordering constraints must be introduced as follows.

Max-plus binary ordering control variables  $w_{i,\mu}(k - \mu)$  are defined as

$$w_{i,\mu}(k - \mu) = \begin{cases} e, & \text{if operation } i \text{ for vessel } k - \mu \\ & \text{precedes operation } i \text{ for vessel } k, \\ \epsilon, & \text{if operation } i \text{ for vessel } k \\ & \text{precedes operation } i \text{ for vessel } k - \mu. \end{cases} \quad (14)$$

The adjoint ordering variable  $\bar{w}_{i,\mu}(k - \mu)$  is defined as

$$\bar{w}_{i,\mu}(k - \mu) = \begin{cases} e, & \text{if } w_{i,\mu}(k - \mu) = \epsilon, \\ \epsilon, & \text{if } w_{i,\mu}(k - \mu) = e. \end{cases} \quad (15)$$

For every node in the network where the order of the vessels is fixed, two ordering constraints for state  $x_i$  must be introduced. By virtue of (14) and (15), only one of the two constraints (16) and (17) is simultaneously active. Note also that the inclusion of binary routing variables  $s_{i,j}(k - \mu) \otimes s_{i,j}(k)$  ensures constraint activation only when both vessels  $k$  and  $k - \mu$  take the same route.

- The time of event  $x_i$  for vessel  $k$  is greater or equal than the time of event  $x_i$  for vessel  $k - \mu$  plus some operation time  $\tau_{i,j}(k - \mu)$  of vessel  $k - \mu$ , only when both  $s_{i,j}(k - \mu)$  and  $s_{i,j}(k)$  equal  $e$ . Thus, when both vessels are sailing through the same route, event  $x_i$  happens first for vessel  $k - \mu$  and thus precedes vessel  $k$ . This can be expressed as

$$x_i(k) \geq x_i(k - \mu) \otimes \tau_{i,j}(k - \mu) \otimes w_{i,\mu}(k - \mu) \otimes s_{i,j}(k - \mu) \otimes s_{i,j}(k). \quad (16)$$

- The time of event  $x_i$  for vessel  $k - \mu$  is greater or equal than the time of event  $x_i$  for vessel  $k$  plus some operation time  $\tau_{i,j}(k)$  of vessel  $k$ , only when both  $s_{i,j}(k - \mu)$  and  $s_{i,j}(k)$  equal  $e$ . Thus, when both vessels are sailing through the same route, event  $x_i$  happens first for vessel  $k$  and thus precedes vessel  $k - \mu$ . This can be expressed as

$$x_i(k - \mu) \geq x_i(k) \otimes \tau_{i,j}(k) \otimes \bar{w}_{i,\mu}(k - \mu) \otimes s_{i,j}(k - \mu) \otimes s_{i,j}(k). \quad (17)$$

As overtaking on the waterways is allowed, ordering constraints are only necessary for the two locks. Equations (16) and (17) must then be extended to consider all possible cases, which amounts to four pairs of equations. A complete derivation is provided for the lock in route 1 (between nodes 3 and 4), but the same must be done for the lock in

route 2 (between nodes 8 and 9). For the sake of simplicity, ordering variables (14) are renamed as defined in Table 5 (renamed adjoint ordering variables follow the same pattern). Moreover, a safety timing  $\tau_{safety}$  is introduced to prevent a vessel from entering the lock immediately after the previous vessel has come out of it.

- Vessel  $k$  and  $k - \mu$  sail downstream with vessel  $k - \mu$  in front,  $\forall k, (k - \mu) \in \mathcal{V}_d$ :

$$x_3(k) \geq x_4(k - \mu) \otimes w_{DD,\mu}(k - \mu) \otimes \tau_{L(3,4)} \otimes s_{3,4}(k - \mu) \otimes s_{3,4}(k). \quad (18)$$

- Vessel  $k$  and  $k - \mu$  sail downstream with vessel  $k$  in front,  $\forall k, (k - \mu) \in \mathcal{V}_d$ :

$$x_3(k - \mu) \geq x_4(k) \otimes \bar{w}_{DD,\mu}(k - \mu) \otimes \tau_{L(3,4)} \otimes s_{3,4}(k - \mu) \otimes s_{3,4}(k). \quad (19)$$

- Vessel  $k$  and  $k - \mu$  sail upstream with vessel  $k - \mu$  in front,  $\forall k, (k - \mu) \in \mathcal{V}_u$ :

$$x_4(k) \geq x_3(k - \mu) \otimes w_{UU,\mu}(k - \mu) \otimes \tau_{L(3,4)} \otimes s_{4,3}(k - \mu) \otimes s_{4,3}(k). \quad (20)$$

- Vessel  $k$  and  $k - \mu$  sail upstream with vessel  $k$  in front,  $\forall k, (k - \mu) \in \mathcal{V}_u$ :

$$x_4(k - \mu) \geq x_3(k) \otimes \bar{w}_{UU,\mu}(k - \mu) \otimes \tau_{L(3,4)} \otimes s_{4,3}(k - \mu) \otimes s_{4,3}(k). \quad (21)$$

- Vessel  $k$  sails downstream and enters the lock, while vessel  $k - \mu$  sails upstream and exits the lock,  $\forall k \in \mathcal{V}_d, \forall (k - \mu) \in \mathcal{V}_u$ :

$$x_3(k) \geq x_3(k - \mu) \otimes w_{DU,\mu}(k - \mu) \otimes \tau_{safety} \otimes s_{4,3}(k - \mu) \otimes s_{3,4}(k). \quad (22)$$

- Vessel  $k$  sails downstream and exits the lock, while vessel  $k - \mu$  sails upstream and enters the lock,  $\forall k \in \mathcal{V}_d, \forall (k - \mu) \in \mathcal{V}_u$ :

$$x_4(k - \mu) \geq x_4(k) \otimes \bar{w}_{DU,\mu}(k - \mu) \otimes \tau_{safety} \otimes s_{4,3}(k - \mu) \otimes s_{3,4}(k). \quad (23)$$

- Vessel  $k$  sails upstream and enters the lock, while vessel  $k - \mu$  sails downstream and exits the lock,  $\forall k \in \mathcal{V}_u, \forall (k - \mu) \in \mathcal{V}_d$ :

$$x_4(k) \geq x_4(k - \mu) \otimes w_{UD,\mu}(k - \mu) \otimes \tau_{safety} \otimes s_{3,4}(k - \mu) \otimes s_{4,3}(k). \quad (24)$$

- Vessel  $k$  sails upstream and exits the lock, while vessel  $k - \mu$  sails downstream and enters the lock,  $\forall k \in \mathcal{V}_u$ ,  $\forall (k - \mu) \in \mathcal{V}_d$ :

$$x_3(k - \mu) \geq x_3(k) \otimes \bar{w}_{UD,\mu}(k - \mu) \otimes \tau_{safety} \otimes s_{3,4}(k - \mu) \otimes s_{4,3}(k). \quad (25)$$

### 3) COMPLETE SMPL MODEL

To derive a complete SMPL model, constraints on the arrival times must be added to the previous ordering and routing constraints, and can be expressed as

$$-x_{12}(k) \geq -a_{12}(k) \otimes s_1(k), \quad (26a)$$

$$-x_{12}(k) \geq -a_{12}(k) \otimes s_3(k), \quad (26b)$$

$$-x_1(k) \geq -a_1(k) \otimes s_2(k), \quad (26c)$$

$$-x_1(k) \geq -a_1(k) \otimes s_4(k), \quad (26d)$$

where  $a_i(k)$  denotes the arrival deadline of vessel  $k$  at node  $i$ , as indicated in Table 2.

The binary routing and ordering variables are given by

- $s_l(k)$ ,  $l \in \{1, 2, 3, 4\}$  (routing), and
- $w_{\mu}(k - \mu)$ ,  $\mu \in \{\mu_{\min}, \dots, \mu_{\max}\}$  (ordering),

which can be stacked into a single vector  $\mathbf{v}(k)$  as

$$\mathbf{v}(k) = \begin{bmatrix} s_1(k) \\ s_2(k) \\ s_3(k) \\ s_4(k) \\ w_{\mu_{\min}}(k - \mu) \\ \vdots \\ w_{\mu_{\max}}(k - \mu) \end{bmatrix} \in (\mathbb{B}_{\epsilon})^{V_{\text{tot}} \times 1}, \quad (27)$$

and  $V_{\text{tot}}$  is the total number of scheduling variables.

The scheduling model can then be written in matrix form as

$$\mathbf{x}(k) = \bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} \mathbf{A}_{\mu}(\mathbf{v}(k), k) \otimes \mathbf{x}(k - \mu) \oplus \mathbf{B}(\mathbf{v}(k)) \otimes \mathbf{u}(k), \quad (28)$$

where  $\mathbf{A}_{\mu}$  and  $\mathbf{B}$  are the system matrices, and whose entries result from the appropriate stacking of individual equations.

### C. SCHEDULING WITH SMPL SYSTEMS: AN MILP REFORMULATION

The previous SMPL system (28) cannot be directly used for scheduling due to the fact that  $\epsilon \stackrel{\text{def}}{=} -\infty$ . Therefore, the optimal scheduling problem must be recast as an MILP problem of the form

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{v} \in \mathbb{Z}^l} & \quad \mathbf{c}_x^T \mathbf{x} + \mathbf{c}_v^T \mathbf{v} \\ \text{subject to} & \quad \mathbf{E}_x \mathbf{x} + \mathbf{E}_v \mathbf{v} \leq \mathbf{b}, \end{aligned} \quad (29)$$

with  $\mathbf{c}_x \in \mathbb{R}^n$ ,  $\mathbf{c}_v \in \mathbb{R}^l$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $\mathbf{E}_x \in \mathbb{R}^{m \times n}$  and  $\mathbf{E}_v \in \mathbb{R}^{m \times l}$ .

Definition of an appropriate performance function and reformulation of the previous routing and ordering constraints are addressed below, and will replace the general cost function and constraints in (29).

### 1) PERFORMANCE FUNCTION

The performance function for vessel  $k$  is denoted with  $J(k)$ , and is built as the sum of two terms as

$$J(k) = J_{\text{out}}(k) + J_{\text{in}}(k), \quad (30)$$

where  $J_{\text{out}}(k)$  and  $J_{\text{in}}(k)$  optimize the states  $\mathbf{x}(k)$  and the inputs  $\mathbf{v}(k)$ , respectively.

On the one hand, penalty on the states is defined as

$$J_{\text{out}}(k) = \sum_{i=1}^n \sigma_i(k) x_i(k), \quad (31)$$

where  $\sigma_i(k)$  is the weight on the state, i.e., time of event occurrence, for vessel  $k$  at node  $i$ , and  $n$  is the total number of nodes in the system. Selection of values for  $\sigma(k)$  should follow from the knowledge of the destination for vessel  $k$ , as only penalization on arrival at the destination is sought.

On the other hand, penalty on the inputs is defined as

$$J_{\text{in}}(k) = \sum_{i=1}^{V_{\text{tot}}} \lambda_i(k) v_i(k) \quad (32)$$

where  $\lambda_i(k)$  is the weight on the input for vessel  $k$  at node  $i$ ,  $V_{\text{tot}}$  is the total number of scheduling values for vessel  $k$  and  $\lambda(k) \stackrel{\text{def}}{=} [\lambda_1(k) \dots \lambda_{V_{\text{tot}}}(k)]$ . If vessel  $k$  can select all existing routes and overtaking with all other vessels is permitted, all entries of  $\lambda(k)$  can be set equal to 0.

$J(k)$  can be rewritten using standard MILP notation as

$$J(k) = \mathbf{c}_x^T \mathbf{x}(k) + \mathbf{c}_v^T \mathbf{v}(k), \quad (33)$$

with  $\mathbf{c}_x^T = \sigma(k)$  and  $\mathbf{c}_v^T = \lambda(k)$ .

### 2) CONSTRAINTS

The max-plus routing and ordering constraints presented in Sections IV-B1 and IV-B2, respectively, are reformulated as MILP constraints.

On the one hand, routing constraints are reformulated using the following transformation:

$$s_i^c = \begin{cases} 0, & \text{for } s_i = e, \\ 1, & \text{for } s_i = \epsilon, \end{cases} \quad (34)$$

such that  $s_i \approx \beta s_i^c$  and  $\beta \ll 0$ ,  $i = 1, 2, 3, 4$ .

The same four cases presented in Section IV-B1 are reformulated using (34):

- Vessel  $k$  sails *downstream* via left route 1,  $\forall k \in \mathcal{V}_d$ :

$$-x_1(k) + \beta s_1^c \leq -u_1(k), \quad (35a)$$

$$-x_2(k) + x_1(k) + \beta s_1^c(k) \leq -\tau_{w(1,2)}(k), \quad (35b)$$

$$-x_3(k) + x_2(k) + \beta s_1^c(k) \leq -\tau_{L(2,3)}, \quad (35c)$$

$$-x_4(k) + x_3(k) + \beta s_1^c(k) \leq -\tau_{L(3,4)}, \quad (35d)$$

$$-x_5(k) + x_4(k) + \beta s_1^c(k) \leq -\tau_{L(4,5)}, \quad (35e)$$

$$-x_6(k) + x_5(k) + \beta s_1^c(k) \leq -\tau_{w(5,6)}(k), \quad (35f)$$

$$-x_{12}(k) + x_6(k) + \beta s_1^c(k) \leq -\tau_{w(6,12)}(k). \quad (35g)$$

- Vessel  $k$  sails *downstream* via right route 2,  $\forall k \in \mathcal{V}_d$ :

$$-x_1(k) + \beta s_2^c \leq -u_1(k), \quad (36a)$$

$$-x_7(k) + x_1(k) + \beta s_2^c(k) \leq -\tau_{w(1,7)}(k), \quad (36b)$$

$$-x_8(k) + x_7(k) + \beta s_2^c(k) \leq -\tau_{L(7,8)}, \quad (36c)$$

$$-x_9(k) + x_8(k) + \beta s_2^c(k) \leq -\tau_{L(8,9)}, \quad (36d)$$

$$-x_{10}(k) + x_9(k) + \beta s_2^c(k) \leq -\tau_{L(10,11)}, \quad (36e)$$

$$-x_{11}(k) + x_{10}(k) + \beta s_2^c(k) \leq -\tau_{w(12,13)}(k), \quad (36f)$$

$$-x_{12}(k) + x_{11}(k) + \beta s_2^c(k) \leq -\tau_{w(11,12)}(k). \quad (36g)$$

- Vessel  $k$  sails upstream via left route 3,  $\forall k \in \mathcal{V}_u$ :

$$x_2(k) - x_1(k) + s_3^c(k)\beta \leq -\tau_{w(2,1)}(k), \quad (37a)$$

$$x_3(k) - x_2(k) + s_3^c(k)\beta \leq -\tau_{L(3,2)}, \quad (37b)$$

$$x_4(k) - x_3(k) + s_3^c(k)\beta \leq -\tau_{L(4,3)}, \quad (37c)$$

$$x_5(k) - x_4(k) + s_3^c(k)\beta \leq -\tau_{L(5,4)}, \quad (37d)$$

$$x_6(k) - x_5(k) + s_3^c(k)\beta \leq -\tau_{w(6,5)}(k), \quad (37e)$$

$$x_{12}(k) - x_6(k) + s_3^c(k)\beta \leq -\tau_{w(12,6)}(k), \quad (37f)$$

$$-x_{12}(k) + s_3^c(k)\beta \leq -u_{12}(k). \quad (37g)$$

- Vessel  $k$  sails upstream via right route 4,  $\forall k \in \mathcal{V}_u$ :

$$x_7(k) - x_1(k) + \beta s_4^c(k) \leq -\tau_{w(7,1)}(k), \quad (38a)$$

$$x_8(k) - x_7(k) + \beta s_4^c(k) \leq -\tau_{L(8,7)}, \quad (38b)$$

$$x_9(k) - x_8(k) + \beta s_4^c(k) \leq -\tau_{L(9,8)}, \quad (38c)$$

$$x_{10}(k) - x_9(k) + \beta s_4^c(k) \leq -\tau_{L(10,9)}, \quad (38d)$$

$$x_{11}(k) - x_{10}(k) + \beta s_4^c(k) \leq -\tau_{w(11,10)}(k), \quad (38e)$$

$$x_{12}(k) - x_{11}(k) + \beta s_4^c(k) \leq -\tau_{w(12,11)}(k), \quad (38f)$$

$$-x_{12}(k) + s_4^c(k)\beta \leq -u_{12}(k). \quad (38g)$$

Moreover, (13) must also be modified using (34):

$$s_1^c(k) + s_2^c(k) + s_3^c(k) + s_4^c(k) = 1. \quad (39)$$

On the other hand, ordering constraints are also reformulated using the following transformation:

$$w_{i,\mu}^c = \begin{cases} 0, & \text{for } w_{i,\mu} = e, \\ 1, & \text{for } w_{i,\mu} = \epsilon, \end{cases} \quad (40)$$

such that  $w_{i,\mu} = \beta w_{i,\mu}^c$  and  $\beta \ll 0$ ,  $i = 1, 2, 3, 4$ . Moreover, the adjoint can be approximated as:  $\bar{w}_{i,\mu} = \beta(1 - w_{i,\mu}^c)$ .

In the same spirit as Section IV-B2, a complete derivation is provided for the lock in route 1 (between nodes 3 and 4), but the same must be done for the lock in route 2 (between nodes 8 and 9). Note that (18)–(25) are grouped below into the four possible cases for convenience:

- Both vessels  $k$  and  $k - \mu$  travel downstream,  $\forall k, (k - \mu) \in \mathcal{V}_d$ :

$$-x_3(k) + x_4(k - \mu) + \beta w_{3,DD,1}^c + \beta s_{3,4}(k) + \beta s_{3,4}(k - \mu) \leq -\tau_{L(3,4)}, \quad (41a)$$

$$-x_3(k - \mu) + x_4(k) - \beta w_{3,DD,1}^c + \beta s_{3,4}(k) + \beta s_{3,4}(k - \mu) \leq -\beta - \tau_{L(3,4)}. \quad (41b)$$

- Both vessels  $k$  and  $k - \mu$  travel upstream,  $\forall k, (k - \mu) \in \mathcal{V}_u$ :

$$-x_4(k) + x_3(k - \mu) + \beta w_{3,UU,1}^c$$

$$+ \beta s_{4,3}(k) + \beta s_{4,3}(k - \mu) \leq -\tau_{L(3,4)}, \quad (42a)$$

$$-x_4(k - \mu) + x_3(k) - \beta w_{3,UU,1}^c + \beta s_{4,3}(k) + \beta s_{4,3}(k - \mu) \leq -\beta - \tau_{L(3,4)}. \quad (42b)$$

- Vessel  $k$  travels downstream and vessel  $k - \mu$  travels upstream,  $\forall k \in \mathcal{V}_d, \forall (k - \mu) \in \mathcal{V}_u$ :

$$-x_3(k) + x_3(k - \mu) + \beta w_{3,DU,1}^c + \beta s_{3,4}(k) + \beta s_{4,3}(k - \mu) \leq -\tau_{safety}, \quad (43a)$$

$$-x_4(k - \mu) + x_4(k) - \beta w_{3,DU,1}^c + \beta s_{3,4}(k) + \beta s_{4,3}(k - \mu) \leq -\beta - \tau_{safety}. \quad (43b)$$

- Vessel  $k$  travels upstream and vessel  $k - \mu$  travels downstream,  $\forall k \in \mathcal{V}_u, \forall (k - \mu) \in \mathcal{V}_d$ :

$$-x_4(k) + x_4(k - \mu) + \beta w_{3,UD,1}^c + \beta s_{4,3}(k) + \beta s_{3,4}(k - \mu) \leq -\tau_{safety}, \quad (44a)$$

$$-x_3(k - \mu) + x_3(k) - \beta w_{3,UD,1}^c + \beta s_{4,3}(k) + \beta s_{3,4}(k - \mu) \leq -\beta - \tau_{safety}. \quad (44b)$$

## V. CASE STUDY

The scheduling approach presented in Section IV is tested on a realistic case study. The overall system is described first, and details regarding the considered fleet of vessels are provided. The scheduler is then tested using different vessel data, which allows to highlight specific features.

### A. SYSTEM DESCRIPTION

Figure 4 was used to derive an SMPL model of a bidirectional IWT system. This inland waterway network is extended for completeness with an additional lock in series as shown in Figure 5, and is used as the case study in this paper. The network consists of three locks: lock 1 (between nodes 3 and 4), lock 2 (between nodes 7 and 8) and lock 3 (between nodes 12 and 13). The starting sailing points are  $x_1(k)$  and  $x_{16}(k)$  for the downstream- and upstream-sailing vessels, respectively, and sail through the network to either  $x_{16}(k)$  or  $x_1(k)$ , respectively. As the endpoints are connected by two routes and vessels can sail in both directions, the equations derived in Section IV only need to be slightly modified to account for the additional lock.

The inputs to the scheduler are summarized in Table 6. Note that a discrete time setting is considered, and thus time parameters are expressed in discrete time units. Six vessels sail through the network: vessels 1, 2, and 5 sail downstream, while vessels 3, 4 and 6 sail upstream. For each of the six vessels, all parameters defined in Table 1 are provided:

- Vessels: waterway sailing times  $\tau_{w(i,j)}(k)$ , departure location  $d(k)$ , departure time  $u_i(k)$ , arrival deadline  $a_i(k)$  and vessel priority  $\sigma(k)$ . It can also be seen that vessels depart one time unit after the preceding vessel, with vessel 1 departing at time unit 1.
- Network: lock, departure and arrival locations, processing times  $\tau_{L(i,j)}$  (which are assumed to be equal for all three locks) and topology matrix  $\mathbf{T}$  (with binary entries:

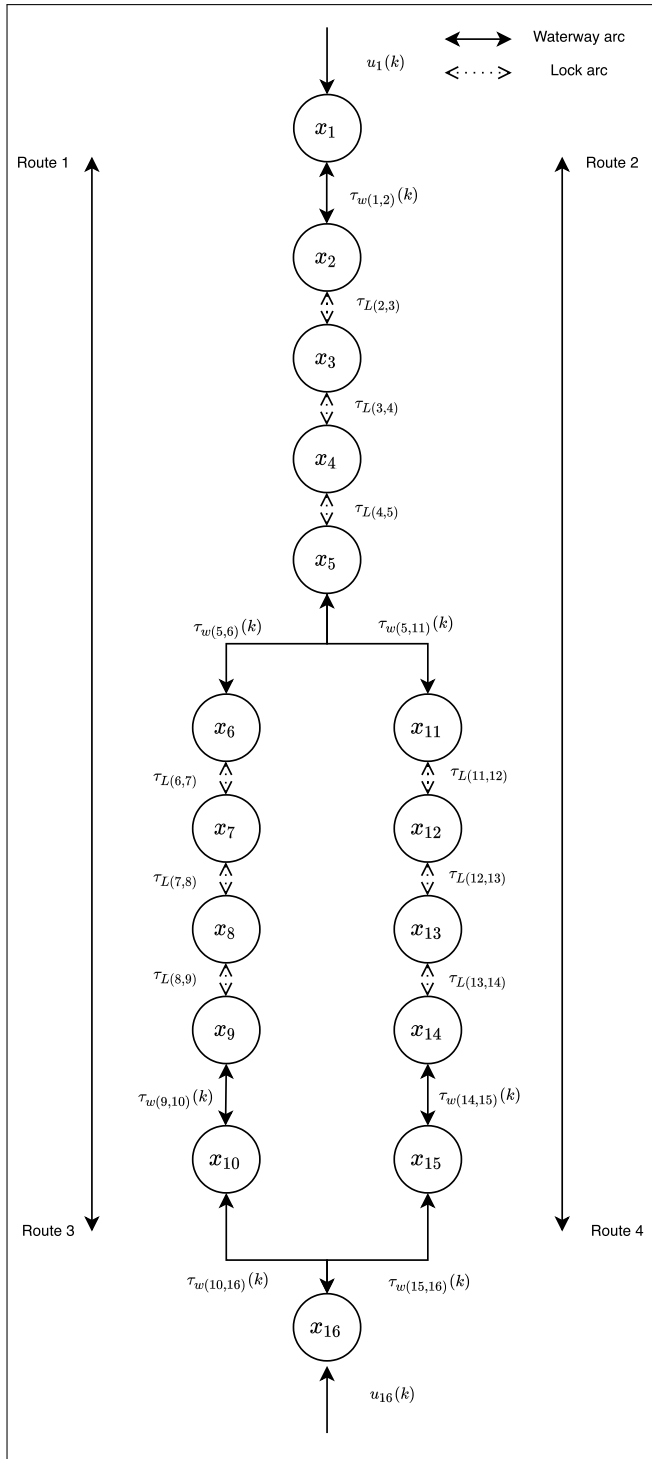


FIGURE 5. Topology graph of the case study.

1 if the two nodes are connected, and 0 otherwise). The latter is given at the bottom of the next page.

**B. RESULTS AND DISCUSSION**

The scheduler, which was built in the MATLAB [34] programming environment, is executed once, thus assuming that

TABLE 6. Definition of input data (expressed in discrete time units).

Operations	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
$\tau_{w(1,2)}(k)$	25	25	25	25	25	25
$\tau_L(2,3)$	2	2	2	2	2	2
$\tau_L(3,4)$	5	5	5	5	5	5
$\tau_L(4,5)$	2	2	2	2	2	2
$\tau_{w(5,6)}(k)$	25	25	25	25	25	25
$\tau_L(6,7)$	2	2	2	2	2	2
$\tau_L(7,8)$	5	5	5	5	5	5
$\tau_L(8,9)$	2	2	2	2	2	2
$\tau_{w(9,10)}(k)$	25	25	25	25	25	25
$\tau_{w(10,16)}(k)$	25	25	25	25	25	25
$\tau_{w(5,11)}(k)$	25	25	25	25	25	25
$\tau_L(11,12)$	2	2	2	2	2	2
$\tau_L(12,13)$	5	5	5	5	5	5
$\tau_L(13,14)$	2	2	2	2	2	2
$\tau_{w(14,15)}(k)$	25	25	25	25	25	25
$\tau_{w(15,16)}(k)$	25	25	25	25	25	25
$d(k)$	1	1	16	16	1	16
$u_1(k)$	1	2	-	-	5	-
$u_{16}(k)$	-	-	3	4	-	6
$a_1(k)$	-	-	140	150	-	170
$a_{16}(k)$	120	130	-	-	160	-
$\sigma(k)$	$0^{16 \times 1}$	$0^{16 \times 1}$	$0^{16 \times 1}$	$0^{16 \times 1}$	$0^{16 \times 1}$	$0^{16 \times 1}$

TABLE 7. Computational performance of scenarios with six vessels.

Constraints	Continuous variables	Binary variables	Gap	Solver time (s)
648	96	204	< 0.01%	< 5

TABLE 8. Optimal vessel schedules (expressed in discrete time units).

States	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Route	Left	Right	Right	Left	Left	Right
$x_1(k)$	1	2	121	131	5	141
$x_2(k)$	26	27	96	106	30	116
$x_3(k)$	28	38	94	104	48	114
$x_4(k)$	33	43	89	99	53	109
$x_5(k)$	35	45	87	88	55	97
$x_6(k)$	60	-	-	63	80	-
$x_7(k)$	62	-	-	61	82	-
$x_8(k)$	67	-	-	56	87	-
$x_9(k)$	69	-	-	54	89	-
$x_{10}(k)$	94	-	-	29	114	-
$x_{11}(k)$	-	70	62	-	-	72
$x_{12}(k)$	-	72	60	-	-	70
$x_{13}(k)$	-	77	55	-	-	65
$x_{14}(k)$	-	79	53	-	-	56
$x_{15}(k)$	-	104	28	-	-	31
$x_{16}(k)$	119	129	3	4	139	6

all required information is available, and the optimal solution is determined using GUROBI [35] as solver. Computational performance is summarized in Table 7.

Optimal schedules and subsequent state evolution of the six vessels are summarized in Table 8 and can be visualized in Figure 6, wherein vertical lines are used to bypass the states on the unselected route. On the other hand, lock passage order is as follows:

- Lock 1: vessel 1 → vessel 2 → vessel 5 → vessel 3 → vessel 4 → vessel 6.
- Lock 2: vessel 4 → vessel 1 → vessel 5.
- Lock 3: vessel 3 → vessel 6 → vessel 2.

All vessels have equal sailing times  $\tau_{w(i,j)}(k)$  and lock operation times  $\tau_{L(i,j)}$ , and no vessel has priority over other vessels. Therefore, vessels which are scheduled on the same

TABLE 9. Vessel delays per lock (expressed in discrete time units).

Object	Direction	Vessel 1	Vessel 2	Vessel 3	Vessel 4	Vessel 5	Vessel 6
Lock 1	Downstream	0	9	-	-	16	-
	Upstream	-	-	0	9	-	7
Lock 2	Downstream	0	-	-	-	0	-
	Upstream	-	-	-	0	-	-
Lock 3	Downstream	-	0	-	-	-	-
	Upstream	-	-	0	-	-	10

TABLE 10. Optimal lock schedules (expressed in discrete time units).

Objects	Direction	Timings				
Lock 1	Upstream	28	38	48	94	104
	Downstream	33	43	89	99	109
Lock 2	Upstream	-	62	82	-	-
	Downstream	56	67	-	-	-
Lock 3	Upstream	-	60	72	-	-
	Downstream	55	65	-	-	-

route and sail in the same direction are expected to arrive in the same order of departure, a fact that is confirmed by the results. Finally, delays at locks (in comparison to the situation in which each vessel sails alone) are given in Table 9.

In what concerns lock properties, the optimal schedules and subsequent state evolution of the locks are summarized in Table 10 and can be visualized in Figure 7. Their properties are as follows:

- Lock levelings:  $\mathcal{L}_1 = 10$ ,  $\mathcal{L}_2 = 4$ , and  $\mathcal{L}_3 = 4$ .
- Empty lock levelings:  $\mathcal{L}_{e,1} = 4$ ,  $\mathcal{L}_{e,2} = 1$ , and  $\mathcal{L}_{e,3} = 1$ .
- Lock occupancy can be inferred from Figure 7: locks are empty during the time intervals defined by horizontal lines. For instance,  $\mathcal{O}_{1,1}(50) = \text{vessel 1}$ .
- Queue order can also be inferred from Figure 7. For instance,  $\mathcal{Q}_{1,1}(35) = 2^{\text{nd}}$  vessel 5  $\rightarrow$  1<sup>st</sup> vessel 2.

TABLE 11. Global network properties.

Variable	Value [time units]
$\mathcal{A}_{\text{net}}$	780
$\mathcal{D}_{\text{net}}$	53
$\mathcal{C}_{\text{net}}$	141
$\mathcal{B}_{\text{net}}$	Lock 1

Finally, the global network properties are computed according to definitions (1)–(4) and summarized in Table 11. Lock 1 is identified as the IWT bottleneck, which comes as no surprise as it is present on all routes, and thus chances of vessels meeting there are higher.

### 1) SCENARIO WITH MULTIPLE VESSEL TYPES

This scenario uses the input data provided in Table 6, but three different types of vessels are considered instead. Vessels 1 and 2 are slow vessels and have a sailing time  $\tau_{w(i,j)}(k) = 25$ . Vessels 3 and 4 are normal vessels and have a sailing time  $\tau_{w(i,j)}(k) = 20$ . Vessel 5 and 6 are fast vessels and have a sailing time of  $\tau_{w(i,j)}(k) = 15$ .

The optimal schedule and state evolution of the six vessels can be visualized in Figure 8. It can be seen how vessels 2 and 5 must wait at lock 1, even if they arrive earlier than vessel 1 due to superior speed. This ordering ensures that vessel 1 meets its deadline on arrival. If a different scheduling policy, e.g., first-come first-serve, was applied, vessel 1 would miss its deadline. Moreover, as the number of vessels is the same as in the first scenario, Table 7 also represents the computational performance of this scenario.

### 2) SCENARIO WITH DIFFERENT VESSEL PRIORITIES

This scenario considers the same input data as in Section V-B1, except for the fact that vessel 5 now transports a special type of cargo that renders early arrival at destination convenient (although not mandatory). This can be

$$T = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \end{matrix}$$

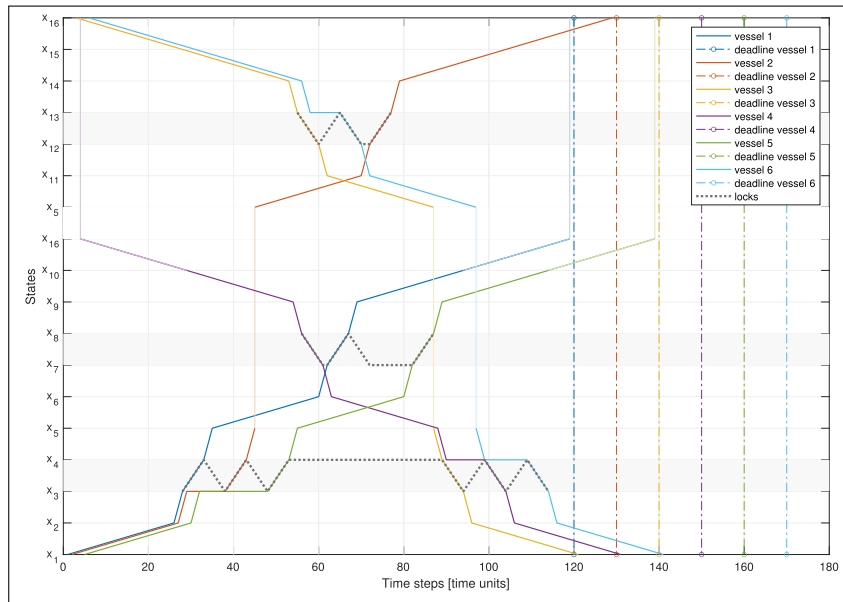


FIGURE 6. Visualization of the optimal vessel schedule (gray shaded areas represent the locks).

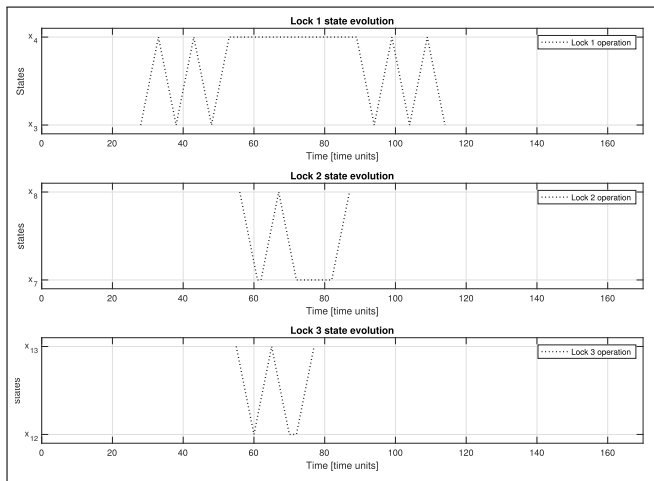


FIGURE 7. Visualization of the optimal lock schedules.

implemented by modifying the vessel priorities  $\sigma(k)$  as follows. Let  $\sigma_a(k)$  denote the entry of  $\sigma(k)$  that corresponds to the arrival node. Then, these values can be modified as follows:  $\sigma_a(5) > \max\{\sigma_a(1), \sigma_a(2), \sigma_a(3), \sigma_a(4), \sigma_a(6)\}$ .

The optimal schedule and state evolution of the six vessels can be visualized in Figure 9. As expected, vessel 5 is allowed to go through lock 1 before vessel 2. Interestingly enough, vessel 1 is now scheduled on the right route 2 instead of the left route 1, as was the case in the previous example. This allows to make room for vessel 5 on route 1. Once again, as this scenario also considers six vessels, Table 7 summarizes its computational performance.

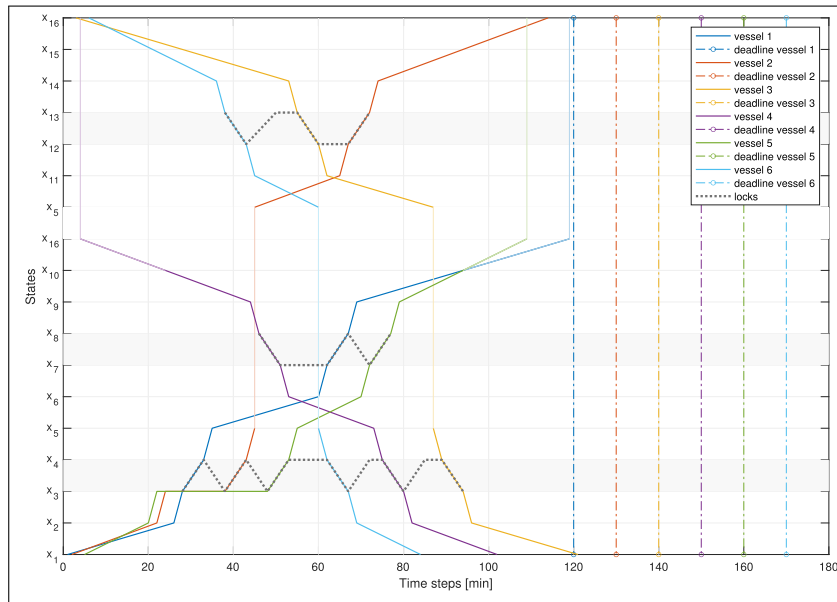
## VI. CONCLUSION AND FUTURE RESEARCH

This paper presented the design of a scheduler for inland waterway transport vessels and locks, as locks constitute

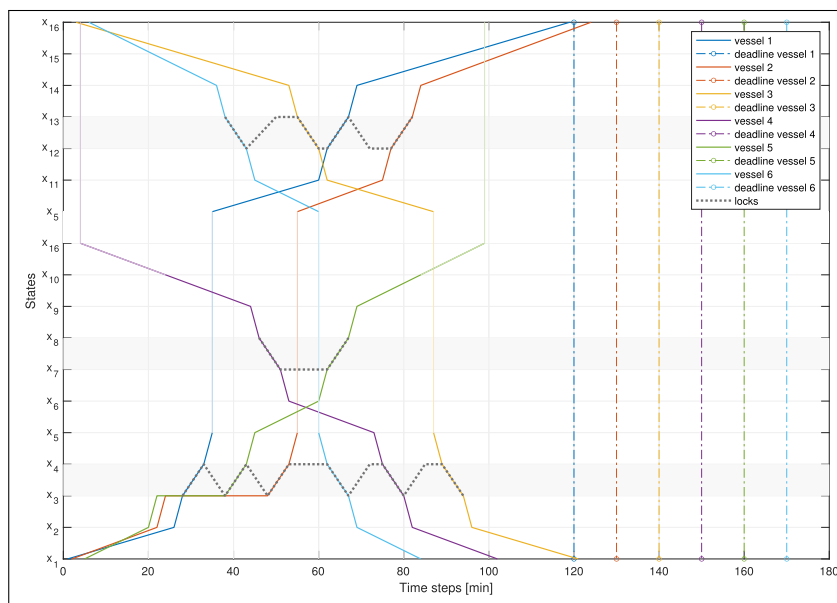
the main bottlenecks for adequate inland waterborne transport (IWT) flows. The scheduler was designed to ensure that vessels reach the destination as early as possible before the arrival deadlines. Although this is a valid performance criterion, it ignores the fact that this situation may incur additional berthing costs at ports and terminals. In this regard, future research will investigate the design of a performance criterion such that minimal gaps between arrival times and deadlines are obtained. This might also lead to lower vessel speeds with respect to the current solution, thus reducing fuel consumption and sailing more efficiently.

The scheduler was designed as a switching max-plus-linear (SMPL) system. On the one hand, event synchronization is modeled via maximization operations, which are linear in max-plus algebra. On the other hand, SMPL systems can deal with varying system dynamics in a straightforward manner, which is of interest in the case of IWT systems as it allows to consider routing and ordering constraints. It is worth noting that the routing and ordering constraints are network-dependent, and different network topologies lead to different MILP models. In this regard, future research will investigate how to automate the generation of models given the topology matrix of the network.

Modeling nonlinear discrete-event systems (DES) as an MPL or SMPL allows for the use of system-theoretical methods to analyze the network [32], [33]. The use of these methods could help address practical issues such as determining which waterway is most impacted when a vessel gets stuck, the maximum possible traffic flow increase with no vessels waiting at locks, and the system bottlenecks and possible mitigating actions. Furthermore, uncertainty analysis can be carried out by using the stochastic version of the SMPL model [26].



**FIGURE 8.** Visualization of optimal vessel schedules for scenario with multiple vessel types (gray shaded areas represent the locks).



**FIGURE 9.** Visualization of optimal vessel schedules for scenario with different vessel priorities (gray shaded areas represent the locks).

The scheduling problem was solved in an offline manner, thus assuming that all the information was available to the scheduler prior to the simulation. In addition to this requirement, the increase in computational burden for increased problem sizes may render offline scheduling unsuitable. An interesting extension to the current scheduling strategy is to perform online scheduling, as it allows for rescheduling in response to new vessel data and measurements (e.g., updated vessel speeds and arrival times), disturbances (e.g., unexpected vessel departures) and disruptions (e.g., blocked waterways and locks). This could be done using model predictive scheduling, which uses

a receding horizon principle that has been studied in detail [33].

The scheduler was tested on a realistic case study, including several variations (different speeds and priorities of vessels) to highlight particular features. While the scheduler succeeded in determining optimal operational timetables for vessels and locks, only single-chamber locks with capacity for one vessel were considered. Future research will consider multiple-chamber locks, which simply appears to require more routing control variables, and multiple vessels within a chamber, which is also based on synchronization but leads to a combinatorial problem. Furthermore, waiting areas at

locks with bounded capacities will also be considered, as the consideration of a maximum capacity renders the problem more realistic from the application perspective. In addition to these features, future work will also consider a wider array of different case studies, and real data will be used for further validation of its applicability.

## REFERENCES

- [1] "Promotion of inland waterway transport." European Commission. 2015. [Online]. Available: [https://ec.europa.eu/transport/modes/inland/promotion\\_en](https://ec.europa.eu/transport/modes/inland/promotion_en)
- [2] "Towards a Strategic Research Agenda: Inland Waterways Transport and Ports in Horizon Europe." Inland Navigation Europe. 2019. [Online]. Available: [https://www.waterborne.eu/images/pdf/190121\\_sra\\_iwp\\_def.pdf](https://www.waterborne.eu/images/pdf/190121_sra_iwp_def.pdf)
- [3] *Port of Rotterdam Vision 2030*, Port Rotterdam, Europort, The Netherlands, 2011.
- [4] S. Lechtenböhrer, D. Schostok, G. Kobiela, K. Knoop, A. Pastowski, and S. Heck. "Deep Decarbonisation Pathways for Transport and Logistics Related to the Port of Rotterdam." 2018. [Online]. Available: <https://www.portofrotterdam.com/sites/default/files/2021-06/wuppertal%20institut%202018%20decarbonization%20of%20transport%20and%20logistics%20synthesis%20report.pdf>
- [5] G. Pauli, "Sustainable transport: A case study of Rhine navigation," *Nat. Resour. Forum*, vol. 34, no. 4, pp. 236–254, Nov. 2010.
- [6] P. Segovia, L. Rajaoarisoa, F. Nejari, E. Duviella, and V. Puig, "Model predictive control and moving horizon estimation for water level regulation in inland waterways," *J. Process Control*, vol. 76, pp. 1–14, Apr. 2019.
- [7] A. S. Grzelakowski, "Assessment of the current state of the EU inland shipping development and its perspectives from the policy and transport market point of view," *Sci. J. Maritime Univ. Szczecin*, vol. 58, no. 130, pp. 74–83, 2019.
- [8] "Roadmap to a single european transport area—Towards a competitive and resource efficient transport system." European Commission. 2011. [Online]. Available: [https://ec.europa.eu/transport/themes/strategies/2011\\_white\\_paper\\_en](https://ec.europa.eu/transport/themes/strategies/2011_white_paper_en)
- [9] W. Li, J. Dai, Y. Xiao, S. Yang, and C. Song, "Estimating waterway freight demand at three gorges ship lock on Yangtze river by backpropagation neural network modeling," *Maritime Econ. Logist.*, vol. 23, no. 3, pp. 495–521, 2021.
- [10] J. Verstichel, P. De Causmaecker, F. Spieksma, and G. V. Berghe, "The generalized lock scheduling problem: An exact approach," *Transp. Res. E, Logist. Transp. Rev.*, vol. 65, no. 1, pp. 16–34, May 2014.
- [11] C. J. Ting and P. Schonfeld, "Control alternatives at a waterway lock," *J. Waterway, Port, Coastal, Ocean Eng.*, vol. 127, no. 2, p. 89, Apr. 2001.
- [12] C. J. Ting and P. Schonfeld, "Efficiency versus fairness in priority control: Waterway lock case," *J. Waterway, Port, Coastal, Ocean Eng.*, vol. 127, no. 2, p. 82, Apr. 2001.
- [13] J. Verstichel, "The lock scheduling problem," Ph.D. dissertation, Faculty Eng. Sci., Katholieke Universiteit Leuven, Leuven, Belgium, 2013.
- [14] J. Verstichel, P. De Causmaecker, F. C. Spieksma, and G. Vanden Berghe, "Exact and heuristic methods for placing ships in locks," *Eur. J. Oper. Res.*, vol. 235, no. 2, pp. 387–398, 2014.
- [15] W. Passchyn, S. Coene, D. Briskorn, J. L. Hurink, F. C. Spieksma, and G. V. Berghe, "The lockmaster's problem," *Eur. J. Oper. Res.*, vol. 251, no. 2, pp. 432–441, Jun. 2016.
- [16] J. Verstichel and G. V. Berghe, "A late acceptance algorithm for the lock scheduling problem," in *Logistik Management*, S. Voß, J. Pahl, and S. Schwarze, Eds. Heidelberg, Germany: Physica-Verlag HD, 2009, pp. 457–478.
- [17] J. Verstichel, P. de Causmaecker, and G. V. Berghe, "Scheduling algorithms for the lock scheduling problem," in *Procedia Social Behav. Sci.*, vol. 20, pp. 806–815, Dec. 2011.
- [18] W. Passchyn, D. Briskorn, and F. C. Spieksma, "Mathematical programming models for lock scheduling with an emission objective," *Eur. J. Oper. Res.*, vol. 248, no. 3, pp. 802–814, Feb. 2016.
- [19] B. Ji, X. Yuan, and Y. Yuan, "A hybrid intelligent approach for co-scheduling of cascaded locks with multiple chambers," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1236–1248, Apr. 2019.
- [20] B. Ji, D. Zhang, S. Y. Samson, and C. Kang, "Mathematical programming models for scheduling multiple cascaded waterway locks," *Comput. Ind. Eng.*, vol. 156, Jun. 2021, Art. no. 107289.
- [21] W. Hammadi, S. M. Senouci, P. Brunet, and M. Ramirez-Martinez, "Two-level optimization to reduce waiting time at locks in inland waterway transportation," *ACM Trans. Intell. Syst. Technol.*, to be published.
- [22] B. Ji, D. Zhang, S. S. Yu, and X. Fang, "An exact approach to the generalized serial-lock scheduling problem from a flexible job-shop scheduling perspective," *Comput. Oper. Res.*, vol. 127, Mar. 2021, Art. no. 105164.
- [23] B. Ji, D. Zhang, S. S. Yu, and B. Zhang, "Optimally solving the generalized serial-lock scheduling problem from a graph-theory-based multi-commodity network perspective," *Eur. J. Oper. Res.*, vol. 288, no. 1, pp. 47–62, 2021.
- [24] H. Guan, Y. Xu, L. Li, and X. Huang, "Optimizing lock operations and ship arrivals through multiple locks on inland waterways," *Math. Problems Eng.*, vol. 2021, pp. 1–13, Sep. 2021.
- [25] T. van den Boom and B. de Schutter, "Modelling and control of discrete event systems using switching max-plus-linear systems," *Control Eng. Pract.*, vol. 14, no. 10, pp. 1199–1211, 2006.
- [26] T. van den Boom and B. de Schutter, "Model predictive control for randomly switching max-plus-linear systems using a scenario-based algorithm," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, 2010, pp. 2298–2303.
- [27] T. van den Boom and B. de Schutter, "Modeling and control of switching max-plus-linear systems with random and deterministic switching," *Discr. Event Dyn. Syst. Theory Appl.*, vol. 22, no. 3, pp. 293–332, Sep. 2012.
- [28] T. van den Boom and B. de Schutter, "Dynamic railway network management using switching max-plus-linear models," *IFAC Proc. Vol.*, vol. 39, no. 12, pp. 343–348, 2006.
- [29] B. Kersbergen, J. Rudan, T. van den Boom, and B. de Schutter, "Towards railway traffic management using switching max-plus-linear systems: Structure analysis and rescheduling," *Discr. Event Dyn. Syst. Theory Appl.*, vol. 26, no. 2, pp. 183–223, 2014.
- [30] G. A. D. Lopes, B. Kersbergen, T. van den Boom, B. de Schutter, and R. Babuška, "Modeling and control of legged locomotion via switching max-plus models," *IEEE Trans. Robot.*, vol. 30, no. 3, pp. 652–665, Jun. 2014.
- [31] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. New York, NY, USA: Springer, 2008.
- [32] B. Heidergott, G. J. Olsder, and J. van der Woude, *Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*. Princeton, NJ, USA: Princeton Univ. Press, 2005.
- [33] T. van den Boom, M. van den Muijsenberg, and B. de Schutter, "Model predictive scheduling of semi-cyclic discrete-event systems using switching max-plus linear models and dynamic graphs," *Discr. Event Dyn. Syst. Theory Appl.*, vol. 30, no. 4, pp. 635–669, 2020.
- [34] *MATLAB, Version 9.9 (R2020ba)*. Natick, MA, USA: MathWorks Inc., 2020.
- [35] (Gurobi Optim., Houston, TX, USA). *Gurobi Optimizer Reference Manual*. (2022). [Online]. Available: <https://www.gurobi.com>