

AN ARCHITECTURE: SOLVING THE LOCK SCHEDULING PROBLEM AT ARTIFICIAL WATERWAYS WITH REINFORCEMENT LEARNING

MUMM, LORENZ⁽¹⁾, ROUSSELET, DAMIANO⁽¹⁾, FEUERSTACK, SEBASTIAN⁽¹⁾, HOEHN, DENNIS⁽¹⁾,
DAUBNER, JANNIS⁽²⁾, LEGEL, TOM⁽²⁾

⁽¹⁾ German Aerospace Center, Oldenburg, Germany
lorenz.mumm@dlr.de, damiano.rousselet@dlr.de, sebastian.feuerstack@dlr.de, dennis.hoeHN@dlr.de

⁽²⁾ Federal Waterways Engineering and Research Institute, Karlsruhe, Germany,
jannis.daubner@baw.de, tom.legel@baw.de

ABSTRACT

This study proposes an architecture to address a critical challenge in inland waterway transport systems: the Lock Scheduling Problem (LSP). Locks and ship lifts are essential components of waterborne infrastructure, yet they frequently act as bottlenecks in the inland waterway network. These bottlenecks can lead to traffic congestion, increased waiting times, and economic losses for industries relying on waterborne transport. In practice, the First-Come-First-Served (FCFS) principle is often used for scheduling vessels. However, this approach often yields suboptimal and inefficient lock utilization. Alternative methods based on mathematical optimization, such as Mixed-Integer-Programming (MIP), offer improved scheduling capabilities but suffer from long computation times and limited adaptability to dynamic and uncertain operational conditions.

To overcome these limitations, this study introduces an architecture that applies Reinforcement Learning (RL) to the LSP.

Keywords: Lock Scheduling, Reinforcement Learning, Artificial Intelligence, Scheduling, Optimization

1 INTRODUCTION

Inland Waterway Transport (IWT) is vital for transporting bulk goods, offering an efficient and more environmentally friendly option compared to road and rail transport. With ongoing advancements in automation, digitalization, and integration into intermodal logistics networks, IWT has the potential to become a key element of future transport systems.

One of the key operational challenges in IWT is the LSP. Since many vessels must pass through one or more locks along their route, the efficiency of lock management has a direct impact on transit times, vessel waiting times, and overall system performance. As a result, optimizing lock scheduling is essential for improving the reliability, attractiveness, and competitiveness of IWT services.

Infrastructure expansion projects are often associated with high costs and long implementation timelines. In contrast, digitalization and process optimization represent promising and cost-effective alternatives for enhancing operational efficiency. The growing availability of data, including Automatic Identification System (AIS) data and lock operation records, creates new opportunities for developing data-driven approaches to solve the LSP.

Currently, lock scheduling is typically performed using FCFS principles combined with manual planning. While alternative approaches based on mathematical optimization models have been proposed, these methods often face challenges related to computational efficiency or solution robustness, particularly under complex and dynamic real-world operating conditions (Chapter 2).

In this study, the architecture of an LSP solution approach based on RL is introduced, and important design decisions are discussed. A theoretical comparison with other mathematical optimization methods is performed, showing the advantages of RL-based scheduling methods for the LSP.

The structure of this paper is organized as follows. Initially, Chapter 2 introduces the LSP and reviews existing solution approaches. In Chapter 3, an overview of RL is provided. Afterward, Chapter 4 describes the methodology developed in this study to address the LSP using RL. Chapter 5 presents an evaluation of the proposed approach, comparing its performance with that of existing lock scheduling strategies and highlighting both its advantages and limitations. Finally, Chapter 6 provides an outlook on further research activities considering the proposed approach.

2 LOCK SCHEDULING PROBLEM

The LSP is one of the most complex and essential planning challenges in IWT. It requires the coordination of multiple locks, often containing chambers of varying sizes, distributed along Artificial Waterways (AWs),

such as canals. Effective scheduling is crucial for maintaining smooth vessel traffic, optimizing operational efficiency, and reducing delays.

According to Verstichel et al. (2014), the LSP can be divided into three closely related sub-problems. The first is the vessel placement problem, which focuses on grouping vessels for joint lockage and determining their arrangement within the chamber. This task resembles a two-dimensional bin-packing problem, where vessel dimensions and mooring requirements must be considered to ensure a safe and feasible configuration. The second sub-problem is the chamber assignment, which involves selecting an appropriate chamber for each lockage. This requires matching the characteristics of the vessels, such as length, width, and draft, with the available chamber dimensions. Since lock chambers often differ in size, proper allocation is crucial to prevent inefficiencies and delays. The third sub-problem is lockage scheduling, which determines the timing of each lockage. This includes defining start and end times while considering factors such as vessel arrival times, sequence-dependent setup operations, and standard operating procedures, including FCFS rules.

These three sub-problems are highly interconnected, making the overall LSP particularly difficult to solve.

As noted by Passchyn et al. (2019), even small instances of general LSPs are computationally intensive and typically belong to the class of NP-hard problems.

Solving the LSP often involves optimizing multiple objectives simultaneously (Verstichel et al. (2014)). Common goals include reducing vessel waiting times, minimizing overall transit durations, and lowering the number of lockages. A range of hard and soft constraints further complicates the problem. Hard constraints must always be satisfied. These include safety requirements such as safety distances, restrictions on overtaking or encountering, prohibitions on mixing vessel types (e.g., passenger and hazardous cargo vessels), and priority rules for certain vessels. Soft constraints, while not mandatory, often support fairness and operational satisfaction. They may aim, for example, to ensure that no vessels or vessel types are systematically disadvantaged in schedules, thereby increasing acceptance.

Better planning results in more predictable vessel movements, lower fuel consumption, and more efficient water use by reducing the total number of lockages required. These outcomes contribute to a more sustainable and reliable IWT system.

2.1 Related Work

In recent years, a substantial number of studies have addressed the LSP, contributing a variety of models and solution techniques.

Smith et al. (2011) presented a scheduling methodology targeting transportation bottlenecks, with a particular emphasis on river locks. Their study compared solutions obtained from a Mixed-Integer-Linear-Programming (MILP) model with those generated by a heuristic scheduling algorithm within an event-based simulation of five locks, each equipped with a single chamber, on the Upper Mississippi River. Results showed that the heuristic method effectively reduced vessel waiting times.

One of the foundational works in this area is by Verstichel et al. (2014), which introduces the Generalized Lock Scheduling Problem and proposes a solution using a MILP model. Their method was tested on scenarios involving both a single lock chamber and a lock with two large chambers. The results indicated that enforcing a FCFS policy significantly improved computational efficiency. For instance, in scenarios involving only 28 vessels, computation times exceeded 12 hours without the FCFS constraint, while applying FCFS reduced this time to approximately 45 minutes.

To further address long computation times, Verstichel et al. (2015) developed an solution method based on Combinatorial Benders' Decomposition. In this approach, the original problem was divided into a master problem, responsible for vessel scheduling and chamber assignment, and a sub-problem, focused on vessel placement. Computational experiments on a single lock with varying chamber configurations demonstrated that the decomposition approach significantly reduced computation times compared to monolithic MILP formulations.

A different modeling approach was proposed by Passchyn et al. (2016), who represented the lock scheduling problem with a single chamber as a directed acyclic graph. The optimal schedule was determined by identifying the shortest path from the source to the sink of the graph. This method offered a novel approach to effectively determining solutions through graph processing methods.

Guan et al. (2021) developed a MILP model that was solved using a large neighborhood search heuristic. Their model divided the waterway into discrete segments, with vessels traveling at constant, vessel-specific speeds. The objective was to minimize waiting times and fuel consumption. The model was evaluated on scenarios with 64 vessels navigating through three sequential locks, each separated by 12-kilometer segments. The results showed improvements in waiting times, travel times and a reduction in the number of lockages.

More recent research has shifted toward environmentally focused objectives. Wu et al. (2024) introduced the Green Lock Scheduling Problem, which focuses on reducing fuel consumption and vessel waiting times. They formulated a MILP model and solved it using a large neighborhood search algorithm embedded in a

decomposition framework. For scenarios involving up to 20 vessels, the model achieved solution times of under one minute while maintaining adherence to the FCFS policy.

Despite the promising theoretical results of these studies, their practical applicability remains limited. Many proposed methods are constrained to simplified scenarios, such as those involving a single lock or a single chamber. In contrast, real-world inland waterways typically consist of sequences of locks, often with multiple chambers, which substantially increases the complexity of scheduling.

Another key limitation is the lack of support for stochastic system behavior. Most mathematical models assume deterministic inputs, such as precise lockage durations, Estimated Times of Arrival (ETA), and processing times. In operational circumstances, however, these parameters are subject to uncertainty due to factors such as weather conditions, human decision-making, and equipment malfunctions. These uncertainties often lead to deviations from planned schedules.

When deviations become significant, rescheduling is required. Although recent methods support the rapid generation of new schedules, as seen in Wu et al. (2024), which include adjustments to lockage times and vessel speeds, excessive rescheduling may reduce acceptance among captains and lock operators. Frequent changes to schedules can be perceived as impractical or disruptive to established workflows, reducing the operational feasibility of such systems.

3 REINFORCEMENT LEARNING

RL is one of the three primary paradigms of machine learning, alongside supervised and unsupervised learning. While supervised learning relies on labeled datasets and unsupervised learning aims to identify patterns in unlabeled data, RL differs fundamentally in that it does not rely on predefined training data. Instead, learning occurs through interaction with an environment, allowing an agent to autonomously generate experience and improve its decision-making ability through trial and error. This interactive nature makes RL particularly well-suited for tasks such as playing games, robotic control, and scheduling.

Especially for scheduling tasks, RL has shown significant results over the last few years. Gao et al. (2017) and Swapno et al. (2024) used RL algorithms to optimize traffic flow by controlling traffic lights, and Lang et al. (2021) modeled production scheduling problems as RL environments based on discrete-event simulation, similar to the approach proposed in this study.

The RL framework consists of two core components: The environment and the agent. The environment defines the problem to be solved and manages all interactions with the agent. At each decision step, the environment provides the agent with an observation, which is a structured, machine-readable representation of the current state. Based on this observation, the agent selects an action. The environment then executes the action, resulting in a new state. It returns a new observation along with a scalar reward that reflects the quality of the chosen action and following state. This sequence comprising observation, action, and reward constitutes one step. A series of such steps, ending in a terminal state, forms an episode. (Sutton et al. (2018))

The agent's goal is to learn a policy that maximizes the cumulative reward. To do so, it must balance two competing objectives: exploration, which involves trying new actions to gather information, and exploitation, which involves selecting known high-reward actions based on prior experience. The reward signals obtained across episodes serve as feedback, enabling the agent to update its policy and improve its decision-making performance over time. (Sutton et al. (2018))

RL has emerged as a powerful approach for addressing complex control problems in dynamic, stochastic, and partially observable environments. One of its key advantages over traditional optimization techniques lies in its separation of computational cost from the decision-making process. Although training and model development can be computationally intensive, a trained RL agent can make decisions in real time (Radac et al. (2025)). Through continued interaction with the environment, often provided via simulation, the agent can learn operational constraints, system dynamics, and random disturbances, ultimately developing a robust and generalizable control strategy.

This robustness is especially beneficial in high-dimensional scenarios where explicit modeling is complex or unreliable.

4 CONCEPT

As discussed in Chapter 2, determining performant and robust solutions for complex instances of the LSP remains a challenging task. This study presents and discusses an RL-based solution architecture.

4.1 Simulation

Simulations play a critical role in facilitating interaction with the problem space during the development of RL-based solution methods for the LSP. To support this objective, a flexible and extensible simulation framework will be designed to enable effective training of RL agents. Given the computational intensity of RL training, particular attention is paid to efficiency. An event-driven simulation approach is selected because it

typically provides better performance and accuracy than time-step-based methods, as demonstrated by Buss et al. (2010).

The simulation framework will be highly configurable, enabling the generation of realistic traffic scenarios that accurately reflect the specific characteristics of the AW under study. Scenarios are characterized based on AW configuration, including infrastructural features, vessel specifications, and prevailing weather conditions. Once a scenario is defined, the number of vessels and locks remains constant throughout the simulation run. A scenario concludes when all vessels have reached their respective destinations. If new vessels are introduced into the AW, a new scenario is initialized using parameters derived from the previous one. This structure ensures that each simulation instance reaches a terminal state, thereby preventing the need to handle non-terminating simulations.

The simulation framework enforces rule-compliant behavior among all entities, including vessels and locks, allowing the simulation to accurately reflect the real-world handling of suboptimal or infeasible scheduling situations.

4.1.1 Artificial Waterway Representation

Following the modeling strategy proposed by Guan et al. (2021), the canal is represented as a sequence of discrete segments. In contrast to their approach, each segment in this study is classified into a specific segment type that governs vessel behavior, especially vessel speed. This is displayed in Figure 2. Here, an exemplary composition of a canal is shown. It starts with multiple standard segments (blue boxes) that are connected to a lock chamber (red box) via upstream and downstream lock forebays (brown boxes). Further back, there are curve segments (green boxes) and a Port Facility (purple box), which finally leads over an unspecified segment (light blue box) to the end of the canal.

Another key distinction lies in the granularity of the segmentation. The segments employed in this work are significantly smaller, typically around 100 meters in length or less, except for special segments such as lock forebays, lock chambers, and port facilities. This fine-grained segmentation provides an approximation of vessel positions along the canal, enabling a detailed yet computationally manageable representation. To maintain a balance between modeling accuracy and computational efficiency, a meter-accurate resolution is deliberately avoided.

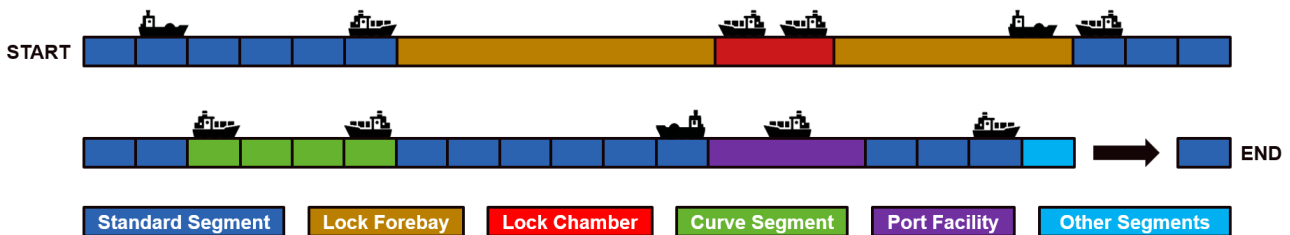


Figure 2. A schematic representation of the segment structure used in the approach.

Modeling vessel behavior in lock forebays presents additional challenges due to the complex and often variable nature of vessel movements in these areas. Therefore, the behavior of vessels within forebay segments is represented using state machines driven by processing times, which will be derived from statistical data. This approach enables the simulation to capture realistic operational variability without incurring the computational cost associated with high-resolution physical modeling.

4.1.2 Lock Operation

Each Lock Operation (LO) specifies key scheduling parameters, including the assigned lock and chamber, the scheduled start time for lockage, the required water level, and the set of vessels involved. To ensure operational robustness, the start time is defined within a 5-minute interval, allowing for minor real-world fluctuations. Each lock chamber processes its assigned LOs sequentially, adhering to the established schedule. Initially, the chamber selects the LO with the earliest scheduled start time. Once an LO has been selected, it is marked as active and becomes unchangeable.

In contrast, LOs that have not yet become active remain modifiable, enabling adaptive and flexible schedule management in response to dynamic system conditions. Upon selecting an LO, the current water level within the chamber is evaluated. If the existing level does not meet the LO's specified requirement, an empty lockage is immediately initiated to adjust the water level before vessel entry. This preemptive adjustment minimizes potential disruptions caused by stochastic influences, such as temporary malfunctions or delays in chamber operations.

Once the required water level is achieved, the lock chamber waits for the arrival and berthing of all vessels scheduled in the active LO. The lockage process then proceeds, encompassing the full sequence of operations, including the opening and closing of lock gates. After the lockage is complete, vessels exit the

chamber in sequence, marking the completion of the current LO and triggering the selection of the next one in the schedule.

In scenarios where vessels arrive at the lock forebay ahead of their scheduled time, they are required to berth within the forebay, which introduces additional time requirements. Similarly, unberthing operations necessitate further time allocations, both of which must be accounted for in the overall scheduling and simulation framework to ensure accurate modeling of real-world operational constraints.

4.2 Environment

To ensure standardized interaction between the simulation framework and the agent, the system adopts the Gymnasium environment interface as introduced in Towers et al. (2025). Consequently, the design of the reward function, observation space, and action space adheres to the specifications and structural requirements defined by this framework.

4.2.1 Action Space

Defining the action space is a central aspect of RL environment design. A key modeling decision of this problem concerns whether the lock schedule should be generated in a single step or constructed incrementally over time through a multi-step decision-making process. One possible approach involves producing a complete, global schedule in a single decision step, effectively reducing the episode length to a single step. While this method simplifies the decision process, it requires the agent to perform long-term planning based on a single observation. As a result, the agent's ability to adapt to schedule disturbances is limited, and the approach closely mirrors traditional optimization techniques, thereby underutilizing the adaptive capabilities of RL.

In contrast, the strategy adopted in this study employs a more dynamic and responsive approach, constructing the schedule incrementally. In this formulation, the RL agent generates new LOs or modifies existing ones whenever a lock chamber completes its current LO. This enables continuous adaptation to real-time system changes and provides each vessel with an updated Requested Time of Arrival (RTA) for its next lock. Such an approach improves coordinated vessel movement through the AW and leverages the strengths of RL in managing dynamic, sequential decision-making environments.

4.2.2 Observation Space

In RL, agents require access to relevant information about the environment to make informed and context-sensitive decisions. To enable this, the current state of the environment is encoded into an observation. The observation space in this study includes vessel-specific, lock chamber-specific, planning-related, and global environmental information.

Vessel-specific features include parameters such as current position, destination, speed, direction of travel, length, width, draft, vessel class, hazardous cargo classification (for example, dangerous goods), and any special operational rights. Temporal variables such as elapsed travel time, cumulative waiting time, ETA, and RTA are also included to support time-sensitive decision-making.

Lock chamber-related information includes each chamber's operational status (active or inactive), geographic location, physical dimensions (length and width), current water level, lockage durations, and the status of any ongoing LOs, including the estimated time remaining until completion. If a lock schedule is already available, it is also included in the observation.

Global environmental features further enrich the observation space. These include contextual variables such as the current time of day, season, hydrological data (such as the water level of the Rhine River), and meteorological conditions. This comprehensive structure enables the agent to understand the broader system context and adapt its scheduling strategy accordingly.

4.2.3 Reward Function

The LSP in this study is formulated as a multi-objective optimization problem. The primary goals are to minimize vessel transit times through the canal, reduce waiting times at lock forebays, and limit the number of lockages performed. These objectives are inherently interdependent and sometimes conflicting.

To manage these trade-offs, a control parameter is introduced that defines the ratio between acceptable increases in transit or waiting time and the benefit of avoiding additional lockages. This parameter allows the scheduling policy to be calibrated according to prevailing operational priorities. For instance, during periods of water scarcity, conserving resources becomes critical, and minimizing the number of lockages is prioritized. In contrast, during winter months, when hydrological constraints are less severe, reducing transit and waiting times typically becomes the dominant objective.

Beyond performance-related goals, the reward function also incorporates penalties for violating system constraints. Negative feedback is applied when the agent generates actions, such as LOs, that breach operational, safety, or regulatory rules. This mechanism ensures that the agent learns to generate feasible, compliant scheduling decisions.

An additional component of the reward function addresses the stability of the scheduling solution. Rescheduling actions are penalized, as frequent changes can result in operational inefficiencies and reduced acceptance by lock operators and vessel crews. A dedicated metric quantifies the degree of change between the current and previous schedule, with greater deviations resulting in higher penalties. This discourages unnecessary modifications while still allowing flexibility when adjustments are justified. The overall aim is to encourage the agent to produce robust, consistent, and practically implementable schedules.

4.3 Action Generation

The incremental scheduling strategy described in Section 4.2.1 raises a key methodological question: How should individual LOs be generated? Two principal approaches are considered.

The first is an operation-centric strategy. In this approach, the agent constructs each LO explicitly by selecting the lock, chamber, lockage start time, lockage direction (upstream or downstream), and the set of involved vessels. While this method offers fine-grained control over the scheduling process, it leads to a high-dimensional action space. As a result, learning becomes more complex, and the agent is more likely to produce infeasible or invalid actions. Examples include selecting vessels traveling in conflicting directions or failing to assign vessels to any LO, both of which can result in schedule inconsistencies.

To address these challenges, the present study adopts a vessel-centric generation strategy. Under this approach, the agent generates one LO per vessel at each decision step, specifying only the desired RTA and the chamber. The lock and direction of lockage are implicitly determined based on the vessel's attributes, which are provided in the observation. This formulation substantially reduces the dimensionality of the action space, thereby improving training efficiency.

However, this vessel-specific approach requires a postprocessing step to ensure that LOs with more than one vessel involved are processed correctly. Vessel-specific LOs that share the same parameters, excluding vessel identity, are merged into a joint LO.

4.4 Error Handling and Correction

A fundamental challenge in RL is managing errors during training, particularly when the agent explores actions that lead to constraint violations. In the context of the LSP, such violations are common during the early training phases, as the agent lacks sufficient experience to generate feasible schedules consistently.

One method for handling these errors is episode truncation, as described by Towers et al. (2025). Under this approach, the episode is terminated immediately upon detection of a constraint violation. Although the resulting incomplete episode is still used for training, constraint violations are penalized through negative feedback. This method can significantly slow learning progress. Frequent early terminations reduce the agent's exposure to advanced states and limit the diversity of its experience. This can delay the convergence toward viable scheduling policies.

An alternative and more interactive strategy adopted in this approach involves the use of shielding mechanisms introduced by Alshiekh et al. (2018). Shielding prevents the agent from executing unsafe actions by intervening after the agent has made a decision but before the action is applied to the environment. The process begins with the postprocessing step, in which vessel-specific LOs are merged into joint operations, as described in Section 5.4. Following, a schedule validation is performed. If any constraint violations are detected, a deterministic repair procedure is applied to transform the schedule into a valid one. Typical repairs include reassigning LOs to a different lock chamber, splitting overloaded LOs into smaller compliant ones, or shifting the LO start times.

Although the repair mechanism ensures feasibility, it may also increase transit times, waiting times, or the total number of lockages. These effects are translated into negative rewards (Section 4.2.3), providing the agent with additional feedback that discourages constraint-violating behavior over time. A key benefit of this approach is that it maintains episode continuity. By avoiding early termination, the agent can safely explore a broader range of scenarios.

4.5 Rescheduling

Adaptability is a fundamental concept in scheduling systems. It refers to the system's ability to respond appropriately to changing or unforeseen circumstances. This study considers two types of rescheduling mechanisms to support adaptability.

The first type, discussed in Sections 4.2.1 and 4.3, involves incremental adjustments to the schedule made by the agent during operation. This approach, referred to as soft rescheduling, enables the agent to maintain stable and consistent schedules while allowing for local adjustments. It is suitable for minor disruptions or gradual changes in operational conditions.

However, there are situations where incremental adjustments are no longer sufficient. When the gap between the current schedule and operational reality becomes too large, a more comprehensive response is required. Such cases may arise due to major disruptions, such as lock chamber malfunctions, traffic incidents, or low water conditions. In these situations, a hard rescheduling procedure is initiated. This involves

discarding the existing schedule entirely and generating a completely new one that reflects the updated state of the system.

To determine whether hard rescheduling is necessary, the system uses a schedule deviation metric. If the deviation exceeds a predefined threshold, a full rescheduling process is triggered. This threshold is a tunable hyperparameter that can be adjusted to balance robustness and responsiveness based on current operational demands and environmental conditions.

5 DISCUSSION

A central element of the proposed approach is the adaptive regulation of vessel behaviors. This form of localized optimization has been successfully applied by Guan et al. (2021) and Wu et al. (2024), demonstrating reductions in fuel consumption. Beyond energy efficiency, adaptive speed control contributes to shorter waiting times in lock forebays and, depending on current traffic conditions, may also reduce overall transit times through AWs.

An additional advantage lies in the range of optimization. The proposed RL approach and conventional optimization techniques both allow for global traffic optimization. For instance, the agent can try to improve traffic flows to alleviate emerging bottlenecks, such as congestion at lock forebays.

Another distinguishing feature of the RL-based method is its ability to generate dynamic and adaptive schedules. Unlike FCFS, which does not involve explicit planning and traditional methods that produce fixed, complete schedules, the RL approach generates partial plans incrementally during operations. This allows early-stage schedule disruptions to be mitigated more effectively and reduces their downstream impact. In contrast, classical optimization methods often require complete rescheduling in response to disturbances, resulting in significant deviations from the assigned initial RTAs.

The computational speed of solution generation is also a critical factor in real-world applications. In operational contexts such as the German IWT system, vessels may register at locks only shortly before arrival, leaving minimal time for optimization. As shown by (Verstichel et al. (2014); Wu et al. (2024)), traditional optimization methods can require substantial computation time. In contrast, a trained RL agent can produce decisions rapidly, depending on the size of the neural network, making it particularly suitable for time-critical scenarios (Radac et al. (2025)).

A further benefit of the proposed method is its reliance on a simulation-based environment rather than a strictly mathematical model, such as MIP or MILP. Although the initial development of a simulation environment entails greater effort, it offers more flexibility in integrating complex, real-world constraints. While mathematical models often abstract such complexity, simulations can incorporate realistic vessel behaviors, such as speed variations, increased space requirements in curved channels, or reduced vessel speed, to account for vessel-vessel encounters. These capabilities are essential for modeling encountering and overtaking restrictions, which require high spatial resolution.

Additionally, stochastic disturbances can be represented more accurately, thereby improving the realism of the model (Smith et al. (2011)).

However, despite its advantages, the RL-based approach also presents challenges. The training process is computationally intensive and time-consuming. However, this disadvantage will not be relevant during productive operations (see Chapter 3). Furthermore, structural changes such as altered lock operating times, fleet composition, or evolving weather changes necessitate retraining the agent. Finally, even with protective mechanisms such as shielding, the risk of suboptimal or erroneous decisions cannot be eliminated.

6 Outlook

In this study, an RL approach is introduced for the LSP. The underlying design decisions were outlined, and the method was discussed in comparison to alternative scheduling strategies, such as FCFS and optimization-based approaches using MIP or MILP.

Ongoing research will focus on testing the implementation of the proposed concept in different artificial and real-world scenarios, for example, derived from the Wesel-Datteln-Kanal, one of Germany's most frequently used AW.

ACKNOWLEDGEMENTS

This study was conducted as part of a research collaboration between the German Aerospace Center (DLR) and the Federal Waterways Engineering and Research Institute (BAW) within the SOKI project (Lock Optimization with Artificial Intelligence).

REFERENCES

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., & Topcu, U. (2018). Safe reinforcement learning via shielding. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*.

- Buss, A., & Al Rowaei, A. (2010). A comparison of the accuracy of discrete event and discrete time. *In Proceedings of the 2010 Winter Simulation Conference* (pp. 1468-1477).
- Gao, J., Shen, Y., Liu, J., Ito, M., & Shiratori, N. (2017). Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target Network. *ArXiv*, abs/1705.02755.
- Guan, H., Xu, Y., Li, L., & Huang, X. (2021). Optimizing Lock Operations and Ship Arrivals through Multiple Locks on Inland Waterways. *Mathematical Problems in Engineering*, 2021(1), 6220559.
- Lang, S., Kuetgens, M., Reichardt, P., & Reggelin, T. (2021). Modeling Production Scheduling Problems as Reinforcement Learning Environments based on Discrete-Event Simulation and OpenAI Gym. *IFAC-PapersOnLine*, 54(1), 793–798. doi:10.1016/j.ifacol.2021.08.093
- Passchyn, W., Coene, S., Briskorn, D., Hurink, J. L., Spieksma, F. C. R., & Vanden Berghe, G. (2016). The lockmaster's problem. *European Journal of Operational Research*, 251(2), 432–441. <https://doi.org/https://doi.org/10.1016/j.ejor.2015.12.007>
- Passchyn, W., & Spieksma, F. (2019). Scheduling parallel batching machines in a sequence. *J. of Scheduling*, 22(3), 335–357.
- Radac, M.-B., & Chirla, D.-P. (2025). Near real-time online reinforcement learning with synchronous or asynchronous updates. *Scientific Reports*, 15(1), 17158. doi:10.1038/s41598-025-00492-7
- Smith, L. D., Nauss, R. M., Mattfeld, D. C., Li, J., Ehmke, J. F., & Reindl, M. (2011). Scheduling operations at system choke points with sequence-dependent delays and processing times. *Transportation Research Part E: Logistics and Transportation Review*, 47(5), 669-680. <https://doi.org/10.1016/j.tre.2011.02.005>
- Sutton, R. S., Barto, A. G. (2018). Reinforcement Learning: An Introduction. *The MIT Press*.
- Swapno, S. M. M. R., Nobel, S. M. N., Meena, P., Meena, V. P., Azar, A. T., Haider, Z., & Tounsi, M. (2024). A reinforcement learning approach for reducing traffic congestion using deep Q learning. *Scientific Reports*, 14(1), 30452. doi:10.1038/s41598-024-75638-0
- Towers, M., Kwiatkowski, A., Terry, J. K., Balis, J. U., Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Tan, H. J. S., & Younis, O. G. (2025). Gymnasium: A Standard Interface for Reinforcement Learning Environments (v1.2.0). *Zenodo*. <https://doi.org/10.5281/zenodo.15753105>
- Verstichel, J., De Causmaecker, P., Spieksma, F., & Vanden Berghe, G. (2014). The generalized lock scheduling problem: An exact approach. *Transportation Research Part E: Logistics and Transportation Review*, 65, 16–34. <https://doi.org/10.1016/j.tre.2013.12.010>
- Verstichel, J., Kinable, J., De Causmaecker, P., & Vanden Berghe, G. (2015). A Combinatorial Benders' decomposition for the lock scheduling problem. *Computers & Operations Research*, 54, 117–128. <https://doi.org/10.1016/j.cor.2014.09.007>
- Wu, Z., Ji, B., & Yu, S. S. (2024). Modeling and Solution Algorithm for Green Lock Scheduling Problem on Inland Waterways. *Mathematics*, 12(8), 1192. <https://doi.org/10.3390/math12081192>