



Mathematical programming models for lock scheduling with an emission objective[☆]



Ward Passchyn^{a,*}, Dirk Briskorn^b, Frits C. R. Spieksma^a

^a KU Leuven Faculty of Economics and Business, ORSTAT, Naamsestraat 69, 3000 Leuven, Belgium

^b Chair of Production and Logistics, Bergische Universität Wuppertal, Rainer-Gruenter-Straße 21, 42119 Wuppertal, Germany

ARTICLE INFO

Article history:

Received 14 December 2014

Accepted 6 September 2015

Available online 14 September 2015

Keywords:

Integer programming

Lock scheduling

Emission reduction

ABSTRACT

We investigate the scheduling of series of consecutive locks. This setting occurs naturally along canals and waterways. Our contribution is to (i) provide two distinct mathematical programming formulations and compare them empirically, (ii) investigate the trade-off between reducing flow time and reducing emissions, and (iii) compare the results of the integrated model to those of a heuristic scheduling the locks separately. Our findings confirm that integrated scheduling of consecutive locks can reduce flow time significantly, and reveal that both model formulations have their merits when compared to each other.

© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

1. Introduction

On many inland waterways, locks are required to ensure a suitable water level for navigation. Typically, and notably when the waterway traffic density is high, locks act as bottlenecks, introducing waiting time for ships that pass through these canals and waterways. We consider here the setting where a series of consecutive locks is arranged in a sequence along a canal. In this problem setting, ships travel in both directions and each lock acts as a single server which handles both the upstream and the downstream traffic. Lock operations must alternate between upwards (downstream to upstream) and downwards (upstream to downstream) movements. Results on scheduling a single lock in order to minimize ships' flow times exist in the literature (see Section 2). The first contribution of this work is to provide two general mixed integer linear programming (MIP) models that consider multiple locks in sequence. By scheduling the system of locks as a whole, we are able to obtain a global schedule that minimizes the ships' flow times over the entire length of the canal (where the flow time of a ship is the total time that the ship spends in the canal). Moreover, these models can deal with an alternative objective function reflecting minimization of fuel cost or pollutant emissions. In both cases, communicating the outcome of the models to the ships

allows them to lower their movement speed when locks are known to be unavailable. This avoids unnecessary idle time where ships would arrive at a lock before it is available. At the same time, this yields a reduction in fuel cost as well as pollutant emissions. The second contribution of this paper is a computational study evaluating the performance of the models. The parameters used in the instances are derived from a real-world dataset. It is reasonable to expect a trade-off between reducing emissions on the one hand, and minimizing the total flow time on the other hand. Our computational study confirms the existence of this trade-off and allows us to analyse this trade-off quantitatively. Finally, we evaluate the impact of an integrated model as compared to a heuristic that is based on scheduling each lock individually.

The remainder of the paper is structured as follows. Section 2 provides a brief overview of related literature and known results. In particular, we mention some literature related to fuel consumption and greenhouse gas emissions. A formal definition of the problem is given in Section 3. We aim to provide a general problem description which may be easily extended by incorporating case-specific constraints or alternative objective functions. For this general problem setting, we propose two integer programming models in Section 4. We additionally provide some valid inequalities and performance improvements for both models. Section 5 outlines the computational experiments. Finally, Section 6 concludes the paper.

2. Related literature

Applying optimization techniques in the context of scheduling locks is not new in literature. Moreover, its relevance is emphasized

[☆] A preliminary version of this work has been presented at the 14th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2014).

* Corresponding author. Tel.: +3216326909.

E-mail addresses: ward.passchyn@kuleuven.be (W. Passchyn), briskorn@uni-wuppertal.de (D. Briskorn), frits.spieksma@kuleuven.be (F.C.R. Spieksma).

in recent papers such as Caris, Janssens, and Macharis (2007), who point out that lock scheduling decisions strongly affect the simulated waiting time and suggest efficient decision rules for lock operating as future work. We first mention three specific waterways that have triggered scientific publications.

An early example is the case of the Welland Canal in North America, which allows ships to bypass the Niagara Falls. The St. Lawrence Seaway Authority, which maintains the canal, faced increasing congestion at the locks along the canal. In Petersen and Taylor (1988), an integer programming model for this situation is described. The authors also discuss a dynamic programming model for scheduling the operations of a single lock, and extend this model to a heuristic that yields an operating schedule for the series of locks along the Welland Canal. In the problem setting considered, the Welland Canal locks can contain at most one ship.

Due to the computational effort involved, a majority of works in the literature restricts the attention to simulation models and heuristic solutions. Examples can be found in a number of publications that consider the Upper Mississippi River (UMR). Smith, Sweeney, and Campbell (2009) use simulation models in order to aid policy decisions to reduce congestion on the UMR. Nauss (2008) describes an integer programming model for clearing a queue at a lock, for example due to technical failure, in the presence of sequence-dependent setup times. A similar MIP that strives to minimize vessel waiting times is integrated in a simulation by Smith et al. (2011) and compared to other priority policies. Different ship sequencing policies are also evaluated by Ting and Schonfeld (2001), who also take chamber assignment policies into account for locks where multiple chambers are available in parallel. A notable difference between the UMR and the problem setting considered in this work is that barges on the UMR are typically not handled in batches and may even require more than one lockage due to their length exceeding the lock capacity.

A final example is the Kiel Canal, which connects the Baltic Sea to the North Sea. Lübbecke, Lübbecke, and Möhring (2014) propose a model for traffic optimization, including the sequencing of ships and scheduling of locks. A related implementation of a lock scheduling heuristic is available online, see Luy (2012).

Research on problems involving a single lock can also be found. Hermans (2008) discusses a MIP model and a MIP-based heuristic for the single lock setting. Passchyn et al. (2015) propose a dynamic programming approach where the total flow time is minimized, while allowing batches of multiple ships to be processed in a single lockage. The authors cover a number of problem extensions to the initial problem setting and perform some computational tests to investigate the performance of heuristics. Verstichel, De Causmaecker, Spijksma, and Vanden Berghe (2014a, 2014b) consider the problem of packing ships in lock chambers and introduce an integer programming model that minimizes the total flow time for a single lock, which may consist of parallel chambers, while taking this packing aspect into account.

Results on obtaining optimum solutions to a system of multiple locks as a whole, are more scarce in the literature. The potential of a centralized approach to scheduling has recently attracted more attention in the field. The Dutch waterway management organization Rijkswaterstaat, for example, is shifting its focus from decentralized lock operations towards the fluent operation of certain ‘corridors’ as a whole, see Kunst (2013). A recent publication by Prandtstetter, Ritzinger, Schmidt, and Ruthmair (2015) discusses a setting very similar to the one considered in this work; they introduce a formal definition for the problem of scheduling locks arranged in a sequence and describe a variable neighbourhood search approach to obtain heuristic solutions. A notable difference with the problem setting discussed here is that in Prandtstetter et al. (2015), the speed of all ships is considered to be fixed.

Besides minimizing the total flow time, a different objective could be to minimize the fuel consumption for ships passing through the waterway system. While the fuel consumption may be an

important economical factor for ship operators, the related emission of greenhouse gases may also be an optimization criterion for governments or waterway organizations. For this, it is important to take the ship’s speed into account, as both the fuel consumption and emissions per kilometre travelled increase at higher ship speeds. In recent years, the operational speed of intercontinental container ships has been decreased to improve fuel efficiency, a practice referred to as ‘slow steaming’. A large body of literature exists on optimizing ship speed for maritime shipping, see e.g. Ronen (1982) for some early results on this topic, and Psaraftis and Kontovas (2013) for a recent survey.

In the context of ocean liner shipping, a reduction in ship speeds translates to a reduction in transport efficiency. On inland waterways however, ships are likely to incur waiting time near bottlenecks such as locks. This provides ships with the opportunity of decreasing their maximal speed on each of the sections along the canal while their total time spent inside the canal, i.e. the flow time, remains unchanged. A paper by Ting and Schonfeld (1999) considers a strategy of reducing ship speed to avoid idle time, reports significant economic benefits for a single lock. In the context of road transport, a model for the time-varying vehicle routing problem is proposed by Qian and Eglese (2014), where the goal is to minimize the greenhouse gas emissions.

To the best of our knowledge, our paper is the first one to provide MIPs for emission reduction in the context of multiple locks on inland waterways.

3. Problem definition

We give here a formal statement of the problem we are considering. For clarity of presentation, all known parameters are represented in uppercase, decision variables in lowercase, and sets in calligraphic script. Given is a series of L consecutive locks, for example, along a canal. We number them according to their position, with lock 1 referring to the lock on the downstream end of the waterway. We refer to the locks using the set $\mathcal{L} = \{1, \dots, L\}$. Over time, S ships arrive at either end of the canal. The set $\mathcal{S} = \{1, \dots, S\}$ allows us to uniquely identify these ships. We assume that the arrival time A_s is known for each $s \in \mathcal{S}$. Each of the ships travels to the end of the canal opposite to its arrival. Thus, a ship either arrives on the downstream side of lock 1 and must pass each of the locks $1, \dots, L$ in that order, or a ship arrives at lock L and must pass all locks in the order $L, \dots, 1$. We use the sets \mathcal{U} and \mathcal{D} to refer to the ships arriving on the upstream and downstream side of the canal respectively. Each lock $\ell \in \mathcal{L}$ has a strictly positive lockage time P_ℓ , which is the time needed for a lock to change its position and to allow ships to leave and enter as needed. We assume that the lockage time is independent of the number of ships in the lock and their direction of travel. Locks also have a positive capacity C_ℓ which gives, for each $\ell \in \mathcal{L}$, an upper bound on the number of ships that can simultaneously be present in lock ℓ . We do not assume an initial position for the locks, i.e. each lock may start at either its upstream or downstream position. The models presented in Section 4 can be easily extended to enforce a fixed initial position. Locks ℓ and $\ell + 1$, $\ell \in \mathcal{L} \setminus \{L\}$, are separated by a section of canal (reach) with length S_ℓ . Each ship $s \in \mathcal{S}$ has a fixed minimum and maximum travel speed, represented by V_s^{\min} and V_s^{\max} respectively. Ship s may thus travel at an arbitrary speed contained in the interval $[V_s^{\min}, V_s^{\max}]$. While the ships can travel at a different speed on different sections of the canal, we assume that each ship maintains a constant speed within each section. Each ship $s \in \mathcal{S}$ may also have an imposed deadline D_s , before which it must have left the last lock that it needs to pass. The problem parameters are summarized in Table 1.

A solution is a schedule specifying the initial position of each lock, a series of times when movements start for each lock, an assignment of ships to lock movements such that each ship is assigned to exactly one appropriate movement (going in the right direction) of each lock,

Table 1
Summary of notation.

\mathcal{L}	$= \{1, \dots, L\}$	The set of all locks
S	$= \{1, \dots, S\}$	The set of all arriving ships
U		The set of all ships arriving on the upstream side
D		The set of all ships arriving on the downstream side
P_ℓ	$(\ell \in \mathcal{L})$	The processing time, i.e. lockage time, for lock ℓ
C_ℓ	$(\ell \in \mathcal{L})$	The lock capacity for lock ℓ
S_ℓ	$(\ell \in \mathcal{L} \setminus \{L\})$	The length of the canal section from lock ℓ to $\ell + 1$
A_s	$(s \in S)$	The arrival time of ship s
V_s^{\min}	$(s \in S)$	The minimum speed attainable by ship s
V_s^{\max}	$(s \in S)$	The maximum speed attainable by ship s
D_s	$(s \in S)$	The deadline for ship s

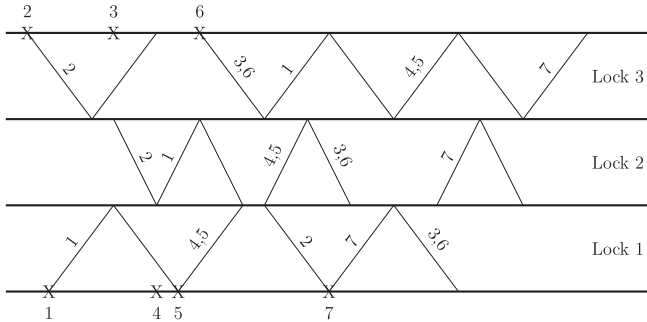


Fig. 1. Illustration of a problem instance ($L = 3, S = 7, C_\ell = 2$) and a feasible solution. Time passes from left to right. Each non-empty lock movement is annotated with the ships it contains. Distances between consecutive locks are not explicitly visualized.

and a speed for each ship on each section between two consecutive locks. For a solution to be feasible we impose four conditions. First, each lock’s movements alternate between the upwards and downwards direction and none of the lock’s movements overlap. Second, the number of ships assigned to a single lock movement must not exceed the lock’s capacity. Third, the speed chosen for each ship s and each section must be contained in the interval $[V_s^{\min}, V_s^{\max}]$. Fourth, each ship must be assigned movements that are consistent with respect to timing, i.e. s cannot enter the first lock before its arrival time A_s , it must not leave the last lock after its deadline D_s , and the time interval available between the exiting of a lock and the entering of the subsequent lock must allow the ship to traverse the corresponding section at the chosen speed.

We can graphically represent an example instance and a feasible solution as depicted in Fig. 1. Time is represented horizontally; the arrival of each ship, over time, is marked with an ‘X’. The positions are represented vertically. Each tilted line corresponds to a lock movement, i.e. the act of changing the water level and possibly transferring a ship to the other side of the lock. The assignment of ships to lock movements is also represented, the figure thus also represents a feasible solution. In the schedule depicted in Fig. 1, we assume that each lock has a capacity of at least 2.

The problems we consider then are to find an optimum feasible solution with respect to the objective functions discussed in Sections 3.1 and 3.2.

3.1. Minimizing total flow time

The first objective we consider is that of minimizing the total flow time, i.e. the elapsed time between the ship’s arrival at the first lock and departure from the last lock in its direction of travel, summed over all ships.

Given this objective function we can easily see that we can fix the speed of each ship s to its maximum value V_s^{\max} without sacrificing optimality.

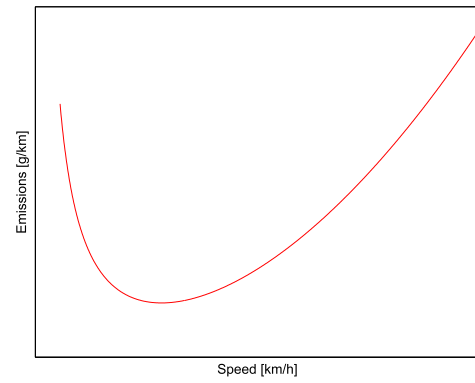


Fig. 2. Qualitative graph of an emission function exhibiting a minimum at a threshold speed.

3.2. Minimizing emissions

A second objective we consider is to minimize total emission by allowing variable ship speeds. We assume that an emission function $E(v)$ is given which expresses the emission of pollutants, in tonnes per kilometre, as a function of the ship speed v for $v > 0$. For simplicity, we assume that this function is identical for all ships. Note that this function E depends on many external factors and is generally non-linear. In idealized settings, the emission per kilometre is typically a strictly increasing (superlinear) function of ship speed due to the ship’s increasing resistance in the water. A cubic relationship, bounded by a constant minimum at low speeds, is often assumed between ship speed and the emissions per time unit, e.g. by Fagerholt, Laporte, and Norstad (2010). When this relationship is expressed in emissions per unit of distance travelled, this corresponds to a quadratic relationship, bounded by a hyperbola for low speed values. A qualitative illustration can be seen in Fig. 2.

Clearly, it is suboptimal to sail at a speed below the threshold value where the minimum is attained, as increasing both the ship speed and subsequent idle time yields a better solution. Regardless of the existence of such a threshold speed, imposing a minimum value for the speed is also justified in order to guarantee sufficient manoeuvrability.

Let $v_{s,p}$ denote the speed of ship s along segment p . The total emission of greenhouse gases E_{tot} can then be computed as follows:

$$E_{\text{tot}} = \sum_{s \in S} \sum_{\ell \in \mathcal{L} \setminus \{L\}} S_\ell E(v_{s,\ell}).$$

A similar approach can be applied in order to minimize fuel consumption, which may be more desirable from the ship operator’s point of view.

We note that for arbitrary functions $E(v)$, it cannot be guaranteed that travelling at a constant speed in an individual section minimizes total emission. However, a constant speed is easily seen to be optimal for convex emission functions, see also Hvattum, Norstad, Fagerholt, and Laporte (2013).

4. Mathematical programming models

We introduce here two distinct MIP formulations representing the problems introduced in Section 3. In this section, we present a basic formulation for both models. Valid inequalities and other model improvements are presented in Appendices A and B.

To state the travel time, which appears in the constraints of both models, as a linear expression of the variables, we introduce the variables $\bar{v}_{s,\ell}$, which equal the reciprocal of $v_{s,\ell}$, i.e. $\bar{v}_{s,\ell} = \frac{1}{v_{s,\ell}}$. The travel time for ship s along the section between locks ℓ and $\ell + 1$ then equals $\bar{v}_{s,\ell} S_\ell$. To characterize the emissions, we use the function $\bar{E}(\bar{v})$, which

expresses the emissions as a function of the reciprocal of ship speed. To obtain a linear model, we use a piecewise linear approximation of $\bar{E}(\bar{v})$. Note that, in general, using a piecewise linear approximation may introduce additional variables as the number of segments increases. If the piecewise function is convex and is to be minimized, however, it suffices to introduce a single additional variable and a set of additional constraints. In order to reduce the notational burden we do not give the linearization here and use $\bar{E}(\bar{v})$ when we refer to the approximation. For an example and additional details on how to compute $E(v)$ and $\bar{E}(\bar{v})$, we refer to Section 5.3.1.

Note that even with S_p and $v_{s,p}$ integral, the travel time for some sections may be fractional. From a practical point of view, however, it might not make sense to schedule lock movements with a higher precision than the unit in which the arrival times are expressed, e.g. minutes. The time-indexed model described below allows lockages to start only at integral moments in time, whereas the second model allows arbitrary starting times for the lock movements.

4.1. Time-indexed formulation

For the model formulation we consider a discretized planning horizon. We consider a set $\mathcal{T} = \{0, \dots, T\}$ of points in time. In the model we restrict ourselves to start times from \mathcal{T} for each lock movement.

We obtain the bound T on the latest possible lockage start time as the sum of the latest arrival time, maximum total travel time, twice the total lockage time, and the maximum waiting time. An upper bound for the latter is derived as the maximum waiting time that a ship can spend at an artificial lock having capacity C_{\min} and lockage duration P_{\max} and assuming that all other ships travelling in the same direction are waiting when s arrives.

Let W_u (W_d) denote the maximum waiting time for a ship arriving on the upstream (downstream) side, and let $C_{\min} = \min_{\ell \in \mathcal{L}} C_\ell$ and $P_{\max} = \max_{\ell \in \mathcal{L}} P_\ell$. Then

$$W_u = \left\lceil \frac{|\mathcal{U}|}{C_{\min}} \right\rceil 4P_{\max},$$

$$W_d = \left\lceil \frac{|\mathcal{D}|}{C_{\min}} \right\rceil 4P_{\max},$$

and with TR_{\max} equal to the maximum time needed to pass through the entire canal without waiting, we obtain

$$T = \max \left(\max_{s \in \mathcal{U}} A_s + TR_{\max} + 2 \times P_{\text{total}} + W_u, \right. \\ \left. \max_{s \in \mathcal{D}} A_s + TR_{\max} + 2 \times P_{\text{total}} + W_d \right).$$

For each ship s and each lock ℓ we can bound the set of points in time at which a lockage of lock ℓ containing ship s can start. It is not hard to compute an earliest and latest possible starting time for such lockages, obtaining a set $\mathcal{T}_{s,\ell}$ for each ship s and lock ℓ .

In addition to the variables $\bar{v}_{s,\ell}$ introduced in Section 3.2, we define the following binary decision variables: For each $s \in \mathcal{S}$, $\ell \in \mathcal{L}$, $t \in \mathcal{T}_s$, let

$$x_{s,\ell,t} = \begin{cases} 1 & \text{if, at time } t, \text{ lock } \ell \text{ starts a lockage with ship } s \\ & \text{inside the lock,} \\ 0 & \text{otherwise.} \end{cases}$$

Note that the completion time c_s can then be expressed as follows:

$$c_s = \begin{cases} \sum_{t \in \mathcal{T}_{s,\ell}} x_{s,\ell,t} (t + P_\ell) & \text{if } s \in \mathcal{D}, \\ \sum_{t \in \mathcal{T}_{s,1}} x_{s,1,t} (t + P_1) & \text{if } s \in \mathcal{U}. \end{cases}$$

Model 1 shows the mathematical formulation of the time-indexed model. For a schedule to be feasible, each ship should pass each of the

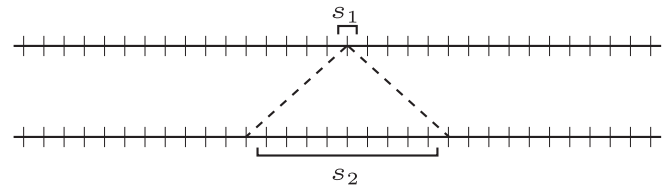


Fig. 3. Visualization of constraint (5), ships travelling in opposite direction.

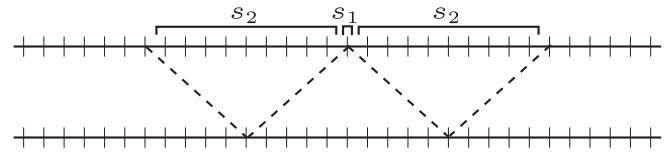


Fig. 4. Visualization of constraints (6) and (7), ships travelling in the same direction.

locks, as imposed by constraints (2). Notice that these constraints also implicitly ensure that the arrival time and deadline for each ship are respected, since only variables in $\mathcal{T}_{s,\ell}$ are considered. Further, all locks must be passed in the correct order, i.e. a ship must arrive at a lock before it can enter that lock. We achieve this by imposing constraints (3) and (4). In these inequalities, the left hand side reflects the time difference between the starting time of the lock movement that handles ship s at lock ℓ , and the ending time of the lockage that handles ship s at the preceding lock.

Additionally, a lockage can only start when the lock is in the appropriate position and not currently moving, i.e. lockages of the same lock should not overlap in time. We impose this by adding the constraints (5)–(7). Figs. 3 and 4 give a visual representation of these constraints. In the figures, for any pair of ships s_1, s_2 travelling in the appropriate direction, a movement for ship s_2 cannot start in the indicated interval if the x variable corresponding to s_1 equals one. Note that constraints (6) and (7), which concern ships travelling in the same direction, allow multiple ships to be handled at the same time.

Finally, the locks' capacity restrictions, the limits on attainable ship speed, and the domain of all variables are trivially modelled by constraints (8)–(11).

We point out that this model contains $O(SLT)$ binary variables, $O(SL)$ real variables, and $O(S^2LT)$ constraints. We refer to Appendix A for a number of model improvements that enhance the performance of the time-indexed model (1)–(11).

4.2. Lockage-based formulation

A notable disadvantage to the time-indexed model is that the number of (binary) variables grows as the time horizon increases. For a small discretization step or when arrival times are large, the value for the upper bound T and thus the number of variables and the computation time to find an optimal solution, may grow prohibitively large. We introduce an alternative formulation that does not use a time index for the variables, and instead numbers the possible lockages. It is clear that for each lock, the number of lockages in an optimal solution does not need to be greater than $2S$. Furthermore, we may also use the bound T introduced in the TI model to bound the number of lockages:

$$K = \min \left(2S, \left\lceil \frac{T + \max_{\ell \in \mathcal{L}} P_\ell}{\min_{\ell \in \mathcal{L}} P_\ell} \right\rceil \right).$$

We thus define the set $\mathcal{K} = \{1, \dots, K\}$ to identify the lockages of each lock. In fact, it is easy to see that we can replace \mathcal{K} with an individual expression \mathcal{K}_ℓ for each lock, but we ignore this for simplicity of notation.

$$\begin{aligned}
 & \text{Minimize } \sum_{s \in \mathcal{S}} (c_s - A_s) \quad \text{or} \quad \text{Minimize } \sum_{s \in \mathcal{S}} \sum_{\ell \in \mathcal{L} \setminus \{L\}} S_\ell \bar{E}(\bar{v}_{s,\ell}) \quad (1) \\
 & \text{Subject to:} \\
 & \sum_{t \in \mathcal{T}_{s,\ell}} x_{s,\ell,t} = 1 \quad \forall s \in \mathcal{S}, \ell \in \mathcal{L} \quad (2) \\
 & \sum_{t \in \mathcal{T}_{s,\ell}} (t x_{s,\ell,t}) - \sum_{t \in \mathcal{T}_{s,\ell+1}} (t x_{s,\ell+1,t}) \geq P_{\ell+1} + \bar{v}_{s,\ell} S_\ell \quad \forall s \in \mathcal{U}, \forall \ell \in \mathcal{L} \setminus \{L\} \quad (3) \\
 & \sum_{t \in \mathcal{T}_{s,\ell}} (t x_{s,\ell,t}) - \sum_{t \in \mathcal{T}_{s,\ell-1}} (t x_{s,\ell-1,t}) \geq P_{\ell-1} + \bar{v}_{s,\ell-1} S_{\ell-1} \quad \forall s \in \mathcal{D}, \forall \ell \in \mathcal{L} \setminus \{1\} \quad (4) \\
 & x_{s_1,\ell,t} + \sum_{\substack{\tau \in \mathcal{T}_{s_2,\ell} \cap \\ \{t-P_\ell+1, \dots, t+P_\ell-1\}}} x_{s_2,\ell,\tau} \leq 1 \quad \forall \ell \in \mathcal{L}, s_1 \in \mathcal{U}, s_2 \in \mathcal{D}, t \in \mathcal{T}_{s_1,\ell} \quad (5) \\
 & x_{s_1,\ell,t} + \sum_{\substack{\tau \in \mathcal{T}_{s_2,\ell} \cap \\ (\{t-2P_\ell+1, \dots, t-1\} \cup \\ \{t+1, \dots, t+2P_\ell-1\})}} x_{s_2,\ell,\tau} \leq 1 \quad \forall \ell \in \mathcal{L}, s_1, s_2 \in \mathcal{U}, t \in \mathcal{T}_{s_1,\ell} \quad (6) \\
 & x_{s_1,\ell,t} + \sum_{\substack{\tau \in \mathcal{T}_{s_2,\ell} \cap \\ (\{t-2P_\ell+1, \dots, t-1\} \cup \\ \{t+1, \dots, t+2P_\ell-1\})}} x_{s_2,\ell,\tau} \leq 1 \quad \forall \ell \in \mathcal{L}, s_1, s_2 \in \mathcal{D}, t \in \mathcal{T}_{s_1,\ell} \quad (7) \\
 & \sum_{s \in \mathcal{S}, t \in \mathcal{T}_{s,\ell}} x_{s,\ell,t} \leq C_\ell \quad \forall \ell \in \mathcal{L}, t \in \bigcup_{s \in \mathcal{S}} \mathcal{T}_{s,\ell} \quad (8) \\
 & \frac{1}{V_s^{\max}} \leq \bar{v}_{s,\ell} \leq \frac{1}{V_s^{\min}} \quad \forall s \in \mathcal{S}, \ell \in \mathcal{L} \setminus \{L\} \quad (9) \\
 & x_{s,\ell,t} \in \{0, 1\} \quad \forall s \in \mathcal{S}, \ell \in \mathcal{L}, t \in \mathcal{T}_{s,\ell} \quad (10) \\
 & \bar{v}_{s,\ell} \in \mathbb{R}^+ \quad \forall s \in \mathcal{S}, \ell \in \mathcal{L} \setminus \{L\} \quad (11)
 \end{aligned}$$

Model 1. Time-indexed model.

In addition to the variables $\bar{v}_{s,p}$, we introduce the following decision variables for each $s \in \mathcal{S}, \ell \in \mathcal{L}, k \in \mathcal{K}$:

$$z_{s,\ell,k} = \begin{cases} 1 & \text{if ship } s \text{ is handled by the } k\text{th lock movement} \\ & \text{of lock } \ell, \\ 0 & \text{otherwise.} \end{cases}$$

$t_{\ell,k}$ equals the starting time of the k th lockage of lock ℓ ,

c_s equals the completion time of ships.

Note that the number of variables does not increase with T . Model 2 presents the complete formulation.

Notice that the model introduces M_s^u, M_s^d , and M_{k_1,k_2} as big- M values. To define their value, we first define the parameter \bar{T} , which gives an upper bound on the latest starting time of any lock movement. Note that this value may exceed the T introduced for the time-indexed model. We have at least $\lceil \frac{S}{c_{\min}} \rceil$ non-empty lockages, thus:

$$\bar{T} = T + P_{\max} + \left(K - \left\lceil \frac{S}{c_{\min}} \right\rceil \right) P_{\max}.$$

We now discuss the individual constraints. Inequalities (13) and (14) ensure that the completion time of each ship, used in the objective function, is consistent with the timing of the last lockage of each ship and enforce the deadlines for each ship. The values M_s^u and M_s^d introduced here give an upper bound on the difference between the latest possible starting time for a lockage and the earliest completion time for a ship s travelling downstream and upstream respectively. With TR_{\min} equal to the minimum time required to cross the entire

canal, we can write:

$$\begin{aligned}
 M_s^u &= \bar{T} + P_1 - A_s - TR_{\min} \\
 M_s^d &= \bar{T} + P_L - A_s - TR_{\min}
 \end{aligned}$$

Constraints (15) guarantee that each ship passes through each of the locks and constraints (16) ensure that they do not overlap in time.

The next set of inequalities, (17)–(19), ensures that each lockage can only handle ships in a single direction, and that no two consecutive lockages carry ships in the same direction. Since the only way to characterize the direction of a lockage is to consider the direction of ships inside the lockage, the direction of lockages need not necessarily alternate when empty lockages are present. Using constraints (17)–(19), however, the model does guarantee that all non-empty lockages satisfy all requirements for a feasible solution.

Obviously, the locks should be passed in the correct order. This is imposed by specifying that the waiting time each ship incurs at each position must be non-negative. Constraints (20) and (21) ensure this for the upstream side of lock L and the downstream side of lock 1, where the arrival times are known. Constraints (22) and (23) do the same for the remaining sides of locks, where ship arrivals are determined by the timing at a preceding lock. The big- M value M_{k_1,k_2} gives an upper bound on the difference in starting time between the lockages k_1 and k_2 :

$$M_{k_1,k_2} = \bar{T} - (k_1 - 1)P_{\min} - (K - k_2 - 1)P_{\min}$$

$$\text{Minimize } \sum_{s \in S} (c_s - A_s) \quad \text{or} \quad \text{Minimize } \sum_{s \in S} \sum_{\ell \in \mathcal{L} \setminus \{L\}} S_\ell \bar{E}(\bar{v}_{s,\ell}) \quad (12)$$

Subject to:

$$D_s \geq c_s \geq t_{1,k} + P_1 - M_s^u \left(1 - \sum_{\kappa=k}^K z_{s,1,\kappa}\right) \quad \forall s \in \mathcal{U}, k \in \mathcal{K} \quad (13)$$

$$D_s \geq c_s \geq t_{L,k} + P_L - M_s^d \left(1 - \sum_{\kappa=k}^K z_{s,L,\kappa}\right) \quad \forall s \in \mathcal{D}, k \in \mathcal{K} \quad (14)$$

$$\sum_{k \in \mathcal{K}} z_{s,\ell,k} = 1 \quad \forall s \in S, \ell \in \mathcal{L} \quad (15)$$

$$t_{\ell,k} \geq t_{\ell,k-1} + P_\ell \quad \forall \ell \in \mathcal{L}, k \in \mathcal{K} \setminus \{1\} \quad (16)$$

$$z_{s_1,\ell,k} + z_{s_2,\ell,k} \leq 1 \quad \forall s_1 \in \mathcal{U}, s_2 \in \mathcal{D}, \ell \in \mathcal{L}, k \in \mathcal{K} \quad (17)$$

$$z_{s_1,\ell,k-1} + z_{s_2,\ell,k} \leq 1 \quad \forall s_1, s_2 \in \mathcal{U}, \ell \in \mathcal{L}, k \in \mathcal{K} \setminus \{1\} \quad (18)$$

$$z_{s_1,\ell,k-1} + z_{s_2,\ell,k} \leq 1 \quad \forall s_1, s_2 \in \mathcal{D}, \ell \in \mathcal{L}, k \in \mathcal{K} \setminus \{1\} \quad (19)$$

$$t_{L,k} \geq A_s \sum_{\kappa=1}^k z_{s,L,\kappa} \quad \forall s \in \mathcal{U}, k \in \mathcal{K} \quad (20)$$

$$t_{1,k} \geq A_s \sum_{\kappa=1}^k z_{s,1,\kappa} \quad \forall s \in \mathcal{D}, k \in \mathcal{K} \quad (21)$$

$$t_{\ell,k_1} \geq t_{\ell+1,k_2} + P_{\ell+1} + \bar{v}_{s,\ell} S_\ell - M_{k_1,k_2} \left(2 - \sum_{\kappa_1=1}^{k_1} z_{s,\ell,\kappa_1} - \sum_{\kappa_2=k_2}^K z_{s,\ell+1,\kappa_2}\right) \quad \forall s \in \mathcal{U}, \ell \in \mathcal{L} \setminus \{L\}, k_1, k_2 \in \mathcal{K} \quad (22)$$

$$t_{\ell,k_1} \geq t_{\ell-1,k_2} + P_{\ell-1} + \bar{v}_{s,\ell-1} S_{\ell-1} - M_{k_1,k_2} \left(2 - \sum_{\kappa_1=1}^{k_1} z_{s,\ell,\kappa_1} - \sum_{\kappa_2=k_2}^K z_{s,\ell-1,\kappa_2}\right) \quad \forall s \in \mathcal{D}, \ell \in \mathcal{L} \setminus \{1\}, k_1, k_2 \in \mathcal{K} \quad (23)$$

$$\sum_{s \in S} z_{s,\ell,k} \leq C_\ell \quad \forall \ell \in \mathcal{L}, k \in \mathcal{K} \quad (24)$$

$$\frac{1}{V_s^{\max}} \leq \bar{v}_{s,\ell} \leq \frac{1}{V_s^{\min}} \quad \forall s \in S, \ell \in \mathcal{L} \setminus \{L\} \quad (25)$$

$$z_{s,\ell,k} \in \{0, 1\} \quad \forall s \in S, \ell \in \mathcal{L}, k \in \mathcal{K} \quad (26)$$

$$t_{\ell,k} \in \mathbb{R}_+ \quad \forall \ell \in \mathcal{L}, k \in \mathcal{K} \quad (27)$$

$$c_s \in \mathbb{R}_+ \quad \forall s \in S \quad (28)$$

Model 2. Lockage-based model.

Finally, the capacity constraint (24) and the domain restrictions for the variables are again straightforward to specify.

The lockage-based model involves $O(S^2L)$ binary variables, $O(SL)$ real variables, and $O(S^3L)$ constraints. We refer to Appendix B for a number of performance improvements for the lockage-based model (12)–(28).

4.3. First-come first-served constraints

Notice that, in general, both models allow ships to overtake each other. That is, for two ships $i, j \in S$ with $A_i < A_j$, the model allows ship j to enter any of the locks strictly before ship i enters that lock. In practice, a first-come first-served rule is often enforced for ships travelling in the same direction. Moreover, in case of no deadlines and identical speed intervals $[V_s^{\min}, V_s^{\max}]$, an exchange argument implies that a solution with minimum flow time can be transformed into a solution that still has minimum flow time and also satisfies first-come first-served at each lock. In fact, this argument also applies to a case where

given deadlines satisfy $D_s - A_s = C$ for some constant C . Since this is the case in all instances we consider in Section 5, we order \mathcal{U} and \mathcal{D} by arrival times and add the following constraints to the TI-model:

$$\sum_{\tau \leq t} x_{s_1,\ell,\tau} \geq \sum_{\tau \leq t} x_{s_2,\ell,\tau} \quad \forall s_1 < s_2 \in \mathcal{U}, \ell \in \mathcal{L}, t \in \mathcal{T} \quad (29)$$

$$\sum_{\tau \leq t} x_{s_1,\ell,\tau} \geq \sum_{\tau \leq t} x_{s_2,\ell,\tau} \quad \forall s_1 < s_2 \in \mathcal{D}, \ell \in \mathcal{L}, t \in \mathcal{T} \quad (30)$$

For the LB-model, we add the following:

$$\sum_{\kappa=1}^k z_{s_1,\ell,\kappa} \geq \sum_{\kappa=1}^k z_{s_2,\ell,\kappa} \quad \forall s_1 < s_2 \in \mathcal{U}, \ell \in \mathcal{L}, \kappa \in \mathcal{K} \quad (31)$$

$$\sum_{\kappa=1}^k z_{s_1,\ell,\kappa} \geq \sum_{\kappa=1}^k z_{s_2,\ell,\kappa} \quad \forall s_1 < s_2 \in \mathcal{D}, \ell \in \mathcal{L}, \kappa \in \mathcal{K} \quad (32)$$

We note that this FCFS assumption need not hold in general. For example, it is easy to see that a deadline can be chosen such that a

ship must move immediately upon its arrival, ignoring waiting ships that arrived at an earlier time.

It is also worth noting that some of the constraints introduced in [Models 1](#) and [2](#) can be tightened by implicitly taking this FCFS property into account. In order to preserve the flexibility of the original model when FCFS does not hold, however, we do not modify the original constraints.

4.4. Model extensions

We note that the models presented above do not include all practical considerations which may occur in a specific real-world setting. The generality of both models, however, allows such extensions to be included with relative ease, although the required computational effort is likely to increase significantly as a result. We briefly mention some of these additional issues that may arise:

- Ships may arrive at, or depart from, the system in between locks, as opposed to at the ends of the canal only.
- The lockage time may depend on the number of ships inside the lock. In particular, empty lockages could be performed very efficiently in comparison to a non-empty lockage.
- The water velocity may affect the lockage time depending on whether the water level is raised or lowered,
- The water velocity may also affect the emission characteristics of ships depending on their direction of travel.

5. Computational study

In this section, we evaluate the performance, with respect to minimizing total flow time, of the two models and their respective performance improvements described in [Appendices A](#) and [B](#). We then use the lockage-based model to minimize the emissions and evaluate the trade-off between emissions and flow time on instances based on real-world data. All instances and results are made available online at <https://perswww.kuleuven.be/~u0086328/lockmasterdata.html>.

5.1. Comparison of the MIP-models

We implement the different model formulations with the flow time objective in CPLEX (version 12.5.1). We evaluate the performance on different sets of instances. We start with the description of a reference scenario in [Section 5.1.1](#). In what follows, we then investigate the impact of different parameters on the performance of the models. More concrete, the number of ships is varied in [Section 5.1.2](#), the number of locks in [Section 5.1.3](#), the time granularity in [Section 5.1.4](#) and the lockage time in [Section 5.1.5](#). Each scenario is evaluated for 10 randomly generated instances. We enforce a limit on the computation time of 15 minutes per instance. No model tuning is performed, and the CPLEX solver options are left to their default values. For a summary of the results, we refer to [Section 5.1.7](#).

5.1.1. Reference scenario

Our reference scenario consists of instances with 3 identical locks and 15 ships. The capacity is equal to 2 ships. We choose a time unit of 5 minutes, i.e. all arrival times are rounded to the nearest multiple of 5 minutes. We choose $P_\ell = 2$ for all $\ell \in \mathcal{L}$. Arrival times are generated randomly from a Poisson arrival process with parameter $\lambda = 0.0536$ (see also [Section 5.3.1](#)). This corresponds to an expected time between arrivals of 18.66 minutes. For simplicity, the distance between the locks is considered to be zero. This allows us to ignore the minimum and maximum speed. Results concerning the distance and ship speed between locks are presented in [Section 5.3](#). Furthermore, since we are minimizing the flow time, we do not need to impose any deadlines. A summary of the input parameters for the reference scenario is given in [Table 2](#).

Table 2
Instance properties for the reference scenario.

Input parameters	
$S = 15$	$C_\ell = 2$
$L = 3$	$\lambda = 0.0536$
$S_\ell = 0$	$D_\ell = \infty$
$P_\ell = 2$	$[V_s^{\min}, V_s^{\max}] = [2, 12]$

Table 3
Model comparison for the reference scenario.

No. of instances solved			
TI	TI+	LB	LB+
10	10	0	10
Avg. relative optimality gap (percent)			
TI	TI+	LB	LB+
0	0	185	0
Avg. time to generate model (seconds)			
TI	TI+	LB	LB+
21.14	1.81	2.63	1.44
Avg. solution time (seconds)			
TI	TI+	LB	LB+
8.89	0.49	900.15	12.89

Table 4
Average number of variables and constraints for the reference instances.

No. of variables (binary, continuous)			
TI	TI+	LB	LB+
(3053, 1)	(784, 1)	(1350, 106)	(1395, 1549)
No. of constraints			
TI	TI+	LB	LB+
137,262	17,921	48,253	10,275

For this reference scenario, we compare the performance of the models presented in [Section 4](#) as well as the models with the performance improvements described in [Appendices A](#) and [B](#). The time-indexed and lockage based models, including the FCFS constraints discussed in [Section 4.3](#), are denoted by TI and LB respectively, while the models with the improvements described in [Appendices A](#) and [B](#) are labelled TI+ and LB+. [Table 3](#) presents the results.

Perhaps surprisingly, the LB and LB+ models do not perform well, despite reducing the number of variables compared to the TI model. To illustrate this, we present the average number of variables and constraints in each of the models in [Table 4](#). Due to the filtering of variables in the TI+ model, this model has an even lower number of variables, although this number depends on the size of the time unit as well as the value \bar{T} , as discussed in [Section 4](#). While the LB model performs very badly, the stronger LB+ model does outperform the TI model without preprocessing. This is good to note since the TI preprocessing is only possible when minimizing the flow time, or alternatively when ship speed is fixed, whereas the improvements introduced in the LB+ model can also be applied in the emission context with variable speed.

For both models presented in [Section 4](#), it can be seen that the modifications suggested in [Appendices A](#) and [B](#) provide a strong reduction in computation time. In what follows, we investigate the sensitivity of the models' performance with respect to different input parameters. Given the drastic improvement of the TI+ and LB+ models over their basic variants, we limit ourselves to the improved models only.

5.1.2. Number of ships

We increase the size of the reference instances up to 25 and 35 ships. The other input parameters remain unchanged, e.g. we still

Table 5
Results for varying problem sizes: 25 ships (left) and 35 ships (right).

No. of instances solved		No. of instances solved	
TI+	LB+	TI+	LB+
10	1	10	0
Avg. relative optimality gap (percent)		Avg. relative optimality gap (percent)	
TI+	LB+	TI+	LB+
0	18	0	67
Avg. time to generate model (seconds)		Avg. time to generate model (seconds)	
TI+	LB+	TI+	LB+
7.62	5.33	22.58	7.02
Avg. solution time (seconds)		Avg. solution time (seconds)	
TI+	LB+	TI+	LB+
2.39	883.19	9.11	900.62

Table 6
Results for varying number of locks: single lock (left) and 5 locks (right).

No. of instances solved		No. of instances solved	
TI+	LB+	TI+	LB+
10	10	10	10
Avg. time to generate model (seconds)		Avg. time to generate model (seconds)	
TI+	LB+	TI+	LB+
1.24	0.97	2.40	1.22
Avg. solution time (seconds)		Avg. solution time (seconds)	
TI+	LB+	TI+	LB+
0.17	0.46	1.17	168.39

ignore distance and ship speed. Table 5 present the results for the TI+ and LB+ models. As a consequence of our method of generating Poisson arrivals, the time horizon is expected to grow proportionally with the number of ships. For the TI model, this implies that the number of variables (and constraints accordingly) will increase. Although the number of variables in the LB model does not depend on \bar{T} , several big- M values in the LB+ constraints will increase as a result. The results indicate that the TI+ model continues to outperform the LB+ model. Notice that the time required to generate the TI+ model grows much faster than that of the LB+ model, which reflects the rapidly increasing number of variables. This is also an early indication of the fact that the time-indexed models will fail to generate larger models due to insufficient memory, whereas the lockage-based models are expected not to suffer from this problem.

5.1.3. Number of locks

To investigate the effect of the number of locks on the solvability, we solve the reference scenario for a single lock as well as for a set of 5 locks. For the latter scenario, we maintain the assumption that all distances between locks are equal to zero. We note that the single-lock scenario should be considered as a simple reference benchmark only, since Passchyn et al. (2015) describe a polynomial time procedure for a variety of single-lock scheduling settings, which outperforms the MIP approach presented here. The results for both sets of instances can be found in Table 6.

For a single lock, it can be seen that the performance of both models is comparable. In the single lock setting, the set of big- M constraints (B.7) and (B.8) vanish from the model, strengthening the LP relaxation. When increasing the number of locks, however, the opposite occurs and additional big- M constraints are added. As a consequence, the LB+ becomes two orders of magnitude slower than the TI+ model.

5.1.4. Time unit

Here, we vary the size of the time unit, i.e. the discretization step size. We solve instances with a time unit equal to 1 minute and to 15 minutes. All arrival times are rescaled and rounded to the nearest unit. We select a lockage time P_ℓ equal to 12 time units and 1 time

Table 7
Results for varying time unit size: 1 minute (left) and 15 minutes (right).

No. of instances solved		No. of instances solved	
TI+	LB+	TI+	LB+
10	10	10	10
Avg. time to generate model (seconds)		Avg. time to generate model (seconds)	
TI+	LB+	TI+	LB+
6.19	1.08	1.18	1.09
Avg. solution time (seconds)		Avg. solution time (seconds)	
TI+	LB+	TI+	LB+
10.44	13.64	0.21	7.23

Table 8
Results for varying lockage time: 1 time unit (left) and 5 time units (right).

No. of instances solved		No. of instances solved	
TI+	LB+	TI+	LB+
10	10	10	10
Avg. time to generate model (seconds)		Avg. time to generate model (seconds)	
TI+	LB+	TI+	LB+
1.42	1.28	3.70	1.00
Avg. solution time (seconds)		Avg. solution time (seconds)	
TI+	LB+	TI+	LB+
0.17	27.38	5.15	3.44

unit in the respective test scenarios, for each lock. Table 7 shows the results.

Unsurprisingly, both models perform better compared to the reference scenario when a larger time unit is chosen. In the reference scenario, the average total computation time is equal to 2.30 and 14.33 seconds for the TI+ and LB+ models respectively (see Table 3), whereas for a time unit of 1 minute, this changes to 16.63 and 14.72 seconds respectively (Table 7). For smaller time units, the performance of the TI+ model thus quickly degrades, as could be expected, whereas the performance of the LB+ model does not seem to suffer significantly compared to the reference scenario.

5.1.5. Lockage time

We also investigate the effect of the lockage duration on the model performance. Table 8 shows the results for a lockage duration P_ℓ equal to 1 and 5 time units respectively. Note that the time unit, as well as the arrival times, remain unchanged. The table illustrates that the performance of the LB+ model improves whereas the TI+ model performs worse as the lockage time increases. This is consistent with the observation that a large lockage time tightens the LB+ model, through the value $M_{k,\ell}^{LB+}$ in constraint (B.11). In the TI+ model, however, increasing the lockage duration yields a larger value T , and thus increases the number of variables to be considered.

5.1.6. Arrival rate

We do not perform a separate test to check for the influence of the arrival rate parameter λ , since varying this parameter is equivalent to varying the time unit and simultaneously rescaling the value for the lockage duration so that the ratio of lockage duration to time unit remains unchanged. Results similar to the scenarios above can be expected, where a low arrival rate favours the LB+ model, and a relatively high arrival rate favours the TI+ model.

5.1.7. Summary of the MIP comparison

To summarize, we conclude that the TI+ model outperforms the other models for most of the tested instances. However, the TI+ model does suffer from two important drawbacks:

- As with any time-indexed model, the number of variables depends on the length of the time horizon. This can be seen in those instances where the time unit is chosen to be small, as well as for larger lockage durations. As the number of ships increases, the

time horizon grows (since the arrival rate λ is fixed), which causes the time needed to generate the TI+ model to increase quickly. Increasing the number of ships, however, is also detrimental to the performance of the LB+ model, prohibiting a comparison for instances containing a larger number of ships.

- The preprocessing performed in the TI+ model to reduce the number of variables assumes that the travel time for each ship on the different sections is known. Preprocessing is thus only possible when ship speed is fixed. Consequently, the TI+ model can only be applied when the objective is to minimize the flow time, in which case the speed for each ship s can be assumed to be equal to V_s^{\max} .

We stress that the second item is an important justification for the use of the LB+ model in Section 5.3, where our aim is to evaluate the speed-dependent emissions. While the TI model may also be used to minimize total emission, as discussed in Section 4.1, the corresponding computation time is significantly larger than that of the LB+ model. Furthermore, the relative difference in computation time is expected to increase even further as smaller time units are considered, as is the case in Section 5.3.

5.2. Comparison with a decentralized heuristic

While the MIP models allow us to find a provably optimal solution, the computational burden can be difficult to overcome for larger instances. To investigate whether the use of an integrated MIP model for multiple locks is justifiable, we compare the exact solution to a solution obtained from an intuitive heuristic. For this experiment we consider only the flow time objective since deadlines should be imposed to obtain sensible results for the emission objective, and it is unclear how to derive deadlines at each of the individual locks while still allowing the model sufficient freedom to minimize emissions.

The heuristic solution is based on a decentralized approach where optimal decisions are made for the individual locks. Each of the locks is iteratively scheduled to minimize flow time individually, using only the known ship arrival times. Initially, only the downstream arrivals are known for the first lock, as well as the upstream arrivals for the last lock. We note that obtaining an optimal solution to the single-lock problem, is possible in polynomial time, see Passchyn et al. (2015). From each individual solution, arrival times are then obtained for the neighbouring locks. This process is repeated until convergence or a fixed iteration limit, since convergence can in fact not be guaranteed. An outline of this procedure in pseudo-code is given in Algorithm 1, where SLS(U, D) represents the single lock solver procedure which returns a solution to the single lock problem given a set of upstream arrivals U and downstream arrivals D , and f_u (f_d) is a function that computes the upstream (downstream) arrival times at a lock given from a given schedule at the previous lock. We denote

Algorithm 1: Pseudo-code algorithm for the RISL heuristic.

Input: U, D, S_ℓ ($\ell \in \mathcal{L} \setminus \{1\}$), \bar{v}

Begin:

$i \leftarrow 1$

$U_1^i \leftarrow U, D_1^1 \leftarrow D$

repeat

$U_{i+1}^i \leftarrow U_i^i, D_{i+1}^1 \leftarrow D_i^1$

for $\ell \in \mathcal{L} \setminus \{1\}$ **do** $U_{i+1}^{\ell-1} = f_u(\text{SLS}(U_i^\ell, D_i^\ell)) + \bar{v}S_{\ell-1}$

for $\ell \in \mathcal{L} \setminus \{L\}$ **do** $D_{i+1}^{\ell+1} = f_d(\text{SLS}(U_i^\ell, D_i^\ell)) + \bar{v}S_\ell$

$i \leftarrow i + 1$

until ($U_i^\ell = U_{i-1}^\ell$ **and** $D_i^\ell = D_{i-1}^\ell$ **for all** $\ell \in \mathcal{L}$) **or** $i = 10L$

Table 9
Input for the RISL heuristic.

Input
$S = 15$
$L = 5$
$S_\ell = 0$
$P_\ell = 12$
$C_\ell = 2$
$\lambda = 0.107$
$D_s = \infty$

Table 10
Results for instances comparing the TI+ and RISL solutions.

Total flow time (minutes)		
TI+	RISL	Relative gain (percent)
1106	1179	6.19
1155	1155	0
1030	1030	0
1066	1066	0
1022	1044	2.11
1080	1080	0
1225	1249	1.92
1056	1057	0.09
1073	1101	2.54
1022	1115	8.34
Average computation time (seconds)		
TI+	RISL	
30.59	0.40	

the heuristic with RISL, short for 'repeated iterations of single lock scheduling'.

The input used for the instances is summarized in Table 9. We again compare the results for 10 instances; the results are listed in Table 10. The heuristic achieves optimal performance in 4 out of the 10 instances. The average performance gain by centralizing the scheduling is 2.12 percent, although the variation over the instances is quite large, up to a maximum of 8.34 percent. Naturally, this comes at the expense of a significantly increased computation time. With respect to minimizing the total flow time, we conclude that there is value in obtaining a centralized schedule. Furthermore, the emission objective inherently requires an integrated approach.

5.3. Evaluating emission reductions

We now evaluate the trade-off between emission reduction and flow time reduction based on an instance created from real-world data. We focus on a section of the upper Scheldt river near Ghent, Belgium.

5.3.1. Instance data

We consider three locks, each consisting of a single chamber with capacity $C_\ell = 2$ and lockage time $P_\ell = 12$, separated by a distance of 10.9 kilometres and 12 kilometres respectively. Arrival times are generated randomly for 25 ships, with a time unit of one minute. Based on a dataset with historical arrival data provided by 'Waterwegen en Zeekanaal' (the organization responsible for managing the locks), an arrival rate of $\lambda = 0.0536$ can be estimated between 6 a.m. and 8 p.m. We use this arrival rate, noting that we do not fix the time horizon of the instances since this is determined from the Poisson arrival process where both the number of arrivals and the Poisson parameter λ are fixed. We report the average results obtained over 10 instances.

In order to derive the piecewise linear function approximating $\bar{E}(\bar{v})$ we employ a tool originating from the ARTEMIS project, see Boulter (2007). It allows to evaluate the CO₂ emissions as a function of multiple inputs, including the ship speed and waterway characteristics. The tool's default ship data is used for ships of the type

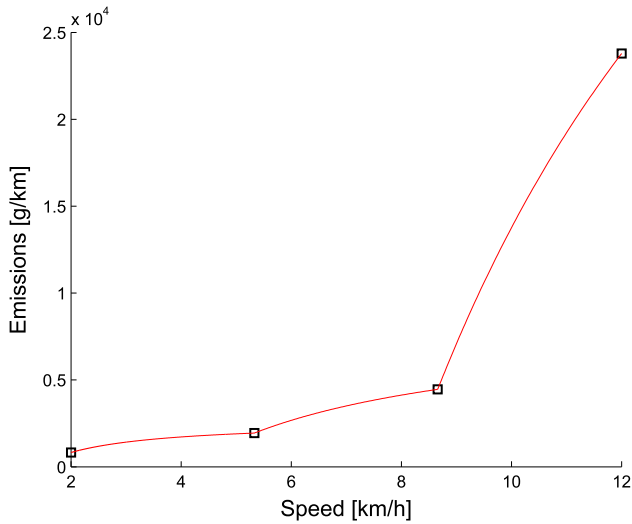


Fig. 5. Resulting approximation of $E(v)$.

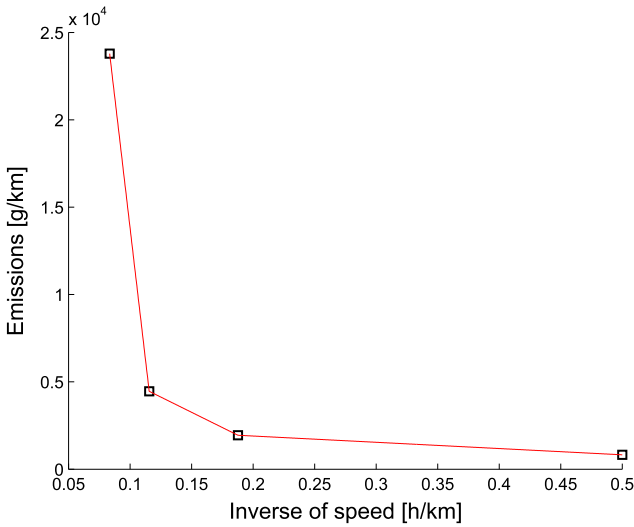


Fig. 6. Piecewise approximation of $\tilde{E}(\tilde{v})$.

‘Rhein–Herne–Kanal’ (RHK), named after the Rhine–Herne Canal for which ships of this size were originally designed. The RHK ships have a size which is representative for freight ships that pass through the upper Scheldt river. The waterway characteristics entered in the ARTEMIS tool are a water depth of 3 metres and an average waterway width of 50 metres.

Using the tool, we evaluate the CO₂ emissions at a speed of 2, 5.33, 8.66, and 12 kilometres/hour, depicted as dots in Figs. 5 and 6. We transform the results into a piecewise linear approximation of $\tilde{E}(\tilde{v})$ with three linear segments. It is important to note that if a piecewise linear function is translated into a function of emission depending on speed, the result is not convex. This is illustrated by the graph in Fig. 5. The effect can be made arbitrarily small, however, by choosing the piecewise linear segments sufficiently short. The resulting functions $\tilde{E}(\tilde{v})$ and $E(v)$ are shown in Figs. 5 and 6 respectively.

Further, we impose a minimum speed of 2 kilometres/hour and the canal’s maximum speed of 12 kilometres/hour for all ships.

5.3.2. Trade-off between emissions and flow time

The trade-off between flow time and emissions is evaluated as follows: first, the LB+ model is used to minimize the total flow time without imposing any deadlines, where ship speed is fixed to the maximum allowed speed of 12 kilometres/hour. Notice that the flow

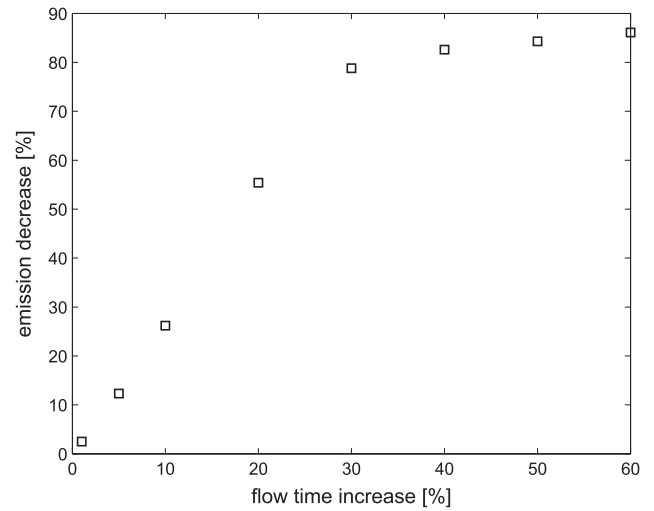


Fig. 7. Trade-off between flow time and emissions, averaged.

time of each ship is at least equal to 150.5 minutes: the time needed to traverse the entire canal at maximum speed plus the total time spent inside locks. To speed up the computation, we set a deadline $D_s = A_s + 240$ minutes for each of the ships. All instances are solved to optimality.

Solving each of these instances yields a set of completion times, which will be used as the reference deadlines in what follows. Next, we treat the ship speed as variable and gradually relax the deadlines so that the flow time for each individual ship is allowed to increase by a given percentage relative to the reference scenario. In each of these modified instances, the ship speed is variable and a deadline is specified for each ship. Table 11 lists the average over the instances of the total emission value, as the flow time is allowed to increase. Observe that a reduction in total emission is possible even without relaxing the reference deadlines, as the variable ship speed allows the slowing down, thus reducing emissions, when waiting time is unavoidable at the locks. This scenario, where the reference deadlines are imposed while ship speed is variable, will be referred to as the reference scenario. We find that the reference scenario allows an average emission reduction of 3.44 percent over the initial setting with fixed speed, while the total flow time remains unchanged. In other words, consistently travelling at the maximum allowable speed results in a total emission increase of 3.58 percent over the reference scenario, with no benefit to the total flow time.

Table 11 also lists the average computation time as flow times are increased. We should note that the time needed to determine the initial deadlines by minimizing the flow time with ship speed fixed to V_s^{\max} , is not included in the table. In fact, this first step provided the largest contribution to computation time, taking 895 seconds on average, although the required time varied greatly by instance, with a minimum of 71 seconds and a maximum of 4374 seconds. Once the reference deadline times are determined, subsequent tests complete much faster since the deadlines strongly constrain the feasible solution space.

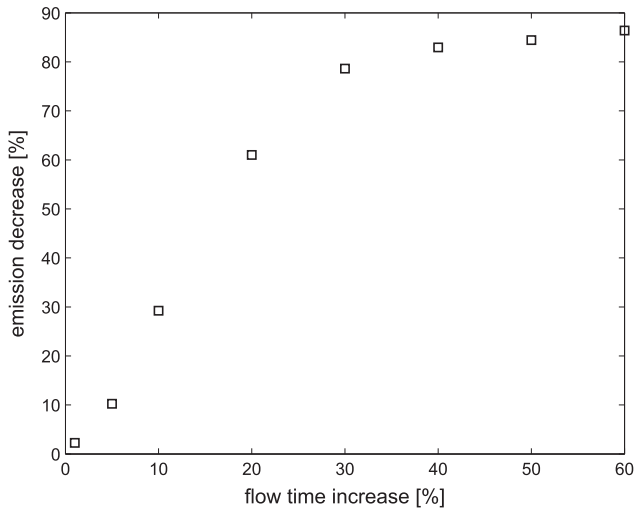
Fig. 7 shows the trade-off between emissions and flow times, averaged over the 10 instances considered. The emission reduction is given as a percentage relative to the solution which minimizes the total flow time where the ship speed is variable, i.e. the reference scenario. It can be seen that the trade-off is approximately linear up to a point where the curve flattens. That is, in our instances, allowing a 1 percent increase in flow time roughly translates into a 3 percent reduction in emission.

It should be mentioned that the trade-off is generally slightly more erratic for a single instance. When the increase in flow time

Table 11

Averaged emission results. The first row refers to the setting where ship speed is fixed at V_s^{\max} . In all other rows, the ship speed is variable and the deadlines are adjusted to constrain the increase in flow time.

Flow time increase (percent)	CO ₂ (tonnes)	CO ₂ reduction (percent)	Comp. time (seconds)
0 (12 kilometres/hour)	13.616	−3.58	5.13
0 (ref)	13.148	0	4.84
1	12.816	2.5	4.85
5	11.534	12.3	5.01
10	9.702	26.2	5.31
20	5.861	55.4	18.53
30	2.788	78.8	37.58
40	2.289	82.6	97.58
50	2.064	84.3	151.13
60	1.834	86.1	243.22

**Fig. 8.** Trade-off between flow time and emissions, single instance.

is insufficient to allow any changes to the assignment of ships to lockages, an emission reduction is possible only by slightly delaying the starting time of the lockages. However, when the increase in flow time becomes sufficiently large, the assignment of a ship may change to a later lockage, introducing a discontinuity in the trade-off curve. Fig. 8 shows the trade-off profile evaluated for a single instance. While this effect is barely visible in this case, it is expected to be more prominent when distances S_ℓ are smaller, or for a larger lockage duration P_ℓ .

6. Conclusion and future work

In this work, we formulated the general problem of scheduling a system of locks arranged in a sequence. We described two mixed integer programming models which allow to solve the problem to optimality. Both models can be easily extended to include additional constraints or alternative objective functions such as minimizing the total greenhouse gas emissions, as opposed to the total flow time.

Computational testing confirms that a time-indexed model (TI) performs well when the unit of time is chosen sufficiently large, thus limiting the number of variables. When the objective is to minimize the total flow time, the TI model permits a preprocessing step (TI+) where the number of variables is reduced, greatly improving the performance. Perhaps more surprisingly, a second model (LB) where the number of variables does not depend on the time unit does not seem to outperform the TI+ model, even with the addition of valid inequalities (LB+) to tighten the LP-relaxation. In the context of minimizing emissions, however, the LB+ model will easily outperform the original TI model.

The limited performance for larger instances suggests a decomposition approach such as branch and price, which may overcome the weak LP relaxation of the LB problem while avoiding to include many unnecessary variables and suffering from the same drawback as the TI model.

Comparing the exact solutions to a heuristic indicates that, while the heuristic may frequently find an optimal solution, significant improvements can be achieved for other instances.

A different set of computational experiments evaluated the trade-off between emission reduction and increase in flow time for a system of three locks on the upper Scheldt river. Compared to the fixed speed scenario, an emission reduction of 3.44 percent could be achieved with no impact on the total flow time. Relaxing the flow time for individual ships allows a further reduction at an approximately linear rate.

One possible direction for future work concerns model extensions which could be more relevant in practical contexts. Examples include modelling the priority of ships, or modifying the lockage duration depending on the ships that are in the lock. Another direction would be to reformulate the models to allow decomposition, or to initialize the exact approach with a heuristic solution.

Acknowledgements

This research has been partially funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. Further, we would like to express our thanks to 'Waterwegen en Zeekanaal NV' for providing ship arrival data at the three locks on the upper Scheldt river. We are also grateful to the referees for their valuable comments and suggestions.

Appendix A. Model improvements for the time-indexed model

We present here a number of enhancements to the time-indexed model. A first performance improvement can be gained by reducing the number of variables if possible, detailed in Section A.1. Next, we discuss a number of valid inequalities which could be added to obtain a stronger LP-relaxation in Section A.2.

A.1. Reducing the number of variables

As argued before we can assume the ship's speeds to be fixed to the maximum speed when we consider the objective to minimize total flow time. It can be easily argued that there exists an optimal solution where each lockage either starts at the arrival time of some ship that is in the lockage, or immediately follows the lockage that precedes it. We can make use of this fact in order to obtain $T'_{s,\ell}$ by eliminating all elements from $T_{s,\ell}$ where no ship can arrive and where no lockage can end. Employing $T'_{s,\ell}$ instead of $T_{s,\ell}$ avoids including many unnecessary variables and yields a drastic performance

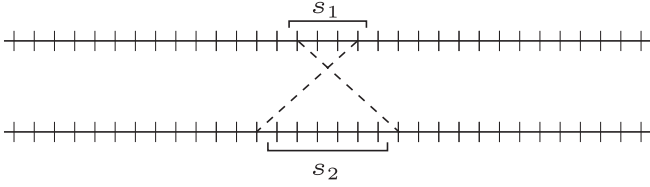


Fig. A.1. Visualization of constraints (A.1) and (A.2) for $p = 3$, ships travelling in opposite direction.

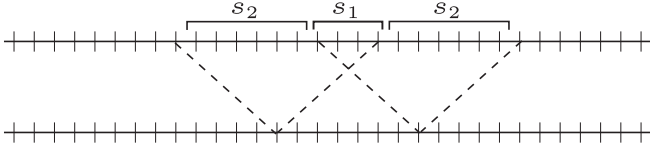


Fig. A.2. Visualization of constraints (A.3) and (A.4) for $p = 3$, ships travelling in the same direction.

improvement. We refer to Section 5 for a comparison of the TI model with and without the application of this procedure.

A.2. Valid inequalities

In the time-indexed model, the constraints that avoid overlap of the lockages contain a single variable $x_{s_1,\ell,t}$ for ship s_1 , and a sum over a time interval for ship s_2 . A generalization of these constraints can be obtained by summing over a wider interval for s_1 , and an appropriately modified interval for ship s_2 .

$$\sum_{\tau \in \mathcal{T}_{s_1,\ell} \cap \{t, \dots, t+p\}} x_{s_1,\ell,\tau} + \sum_{\tau \in \mathcal{T}_{s_2,\ell} \cap \{t-p+1, \dots, t+2p-1\}} x_{s_2,\ell,\tau} \leq 1 \quad \forall \ell \in \mathcal{L}, s_1 \in \mathcal{D}, s_2 \in \mathcal{U}, t \in \mathcal{T}_{s_1,\ell}, p \in \{0, \dots, 2p_\ell - 2\} \quad (A.1)$$

$$\sum_{\tau \in \mathcal{T}_{s_1,\ell} \cap \{t, \dots, t+p\}} x_{s_1,\ell,\tau} + \sum_{\tau \in \mathcal{T}_{s_2,\ell} \cap \{t-p+1, \dots, t+2p-1\}} x_{s_2,\ell,\tau} \leq 1 \quad \forall \ell \in \mathcal{L}, s_1 \in \mathcal{U}, s_2 \in \mathcal{D}, t \in \mathcal{T}_{s_1,\ell}, p \in \{0, \dots, 2p_\ell - 2\} \quad (A.2)$$

It is not hard to see that we do not exclude any feasible solution while cutting solutions to the LP-relaxation. We can use a similar approach when s_1 and s_2 are travelling in the same direction:

$$\sum_{\tau \in \mathcal{T}_{s_1,\ell} \cap \{t, \dots, t+p\}} x_{s_1,\ell,\tau} + \sum_{\tau \in \mathcal{T}_{s_2,\ell} \cap (\{t-2p+1, \dots, t-1\} \cup \{t+p+1, \dots, t+2p-1\})} x_{s_2,\ell,\tau} \leq 1 \quad \forall \ell \in \mathcal{L}, s_1 \in \mathcal{U}, s_2 \in \mathcal{U}, t \in \mathcal{T}_{s_1,\ell}, p \in \{0, \dots, 2p_\ell - 2\} \quad (A.3)$$

$$\sum_{\tau \in \mathcal{T}_{s_1,\ell} \cap \{t, \dots, t+p\}} x_{s_1,\ell,\tau} + \sum_{\tau \in \mathcal{T}_{s_2,\ell} \cap (\{t-2p+1, \dots, t-1\} \cup \{t+p+1, \dots, t+2p-1\})} x_{s_2,\ell,\tau} \leq 1 \quad \forall \ell \in \mathcal{L}, s_1 \in \mathcal{D}, s_2 \in \mathcal{D}, t \in \mathcal{T}_{s_1,\ell}, p \in \{0, \dots, 2p_\ell - 2\}. \quad (A.4)$$

Graphically, we can represent these constraints as shown in Figs. A.1 and A.2.

Note that for $p = 0$, constraints (A.2)–(A.4) imply constraints (5)–(6) of the original IP model. Although constraint (A.2) suffices to prevent overlap of alternating lock movements, we also repeat this constraint for ships moving from downstream to upstream to cut away additional LP-solutions. A tighter LP relaxation can thus be obtained by replacing constraints (5)–(7) by constraints (A.1)–(A.4).

Experiments with this stronger formulation show that, even though the LP-relaxation yields a stronger bound, the solution time does not decrease on average, which can be explained by the computational burden of generating and processing these $O(p_{\max} S^2 LT)$ additional constraints. However, these stronger constraints may still be

of interest when the focus is on finding a good LP solution, or if a separation procedure could be applied in order to avoid generating many of the non-binding constraints.

Appendix B. Model improvements for the lockage-based model

We introduce here a reformulation of the LB-model that allows to fix some of the variables, as well as different valid inequalities that strengthen the LP-bound.

Because lockages must alternate direction, we can impose that even-numbered lockages contain only ships in \mathcal{U} and all odd-numbered lockages contain only ships in \mathcal{D} , or vice versa. We can do this by enforcing all ships in \mathcal{U} and \mathcal{D} to be restricted to even and odd numbered lockages respectively, while allowing for an additional empty lockage that ends at time $t = 0$. This additional lockage is required to allow the first lockage starting not earlier than $t = 0$ to be an upstream as well as a downstream lockage. We impose that:

$$z_{s,\ell,k} = 0 \quad \forall s \in \mathcal{U}, \ell \in \mathcal{L}, k \in \mathcal{K} \cup \{K+1\} : k \text{ is even} \quad (B.1)$$

$$z_{s,\ell,k} = 0 \quad \forall s \in \mathcal{D}, \ell \in \mathcal{L}, k \in \mathcal{K} \cup \{K+1\} : k \text{ is odd}, \quad (B.2)$$

which fixes approximately half of the original $z_{s,\ell,k}$ variables to zero, effectively removing them from the model, at the cost of increasing K by one and some adjustments to inequalities (20) and (21) discussed below. Furthermore, we may remove constraints (17)–(19) from the formulation, since they are no longer relevant.

Some care must be taken to allow the first lockages at locks 1 and L to start at time $-P_1$ and $-P_L$. Constraints (20) and (21) imply non-negative starting times for all lockages. We can substitute the following for these two constraints:

$$t_{1,k} \geq \sum_{\kappa=1}^k z_{s,1,\kappa} (A_s + P_1) - P_1 \quad \forall s \in \mathcal{D}, k \in \mathcal{K} \quad (B.3)$$

$$t_{L,k} \geq \sum_{\kappa=1}^k z_{s,L,\kappa} (A_s + P_L) - P_L \quad \forall s \in \mathcal{U}, k \in \mathcal{K}. \quad (B.4)$$

In addition, the big- M value M_{k_1,k_2} should be increased by P_{\max} .

In the remainder, we will discuss a number of valid inequalities to enhance this reformulated LB-model. To express these valid inequalities, we introduce two new sets of continuous variables:

$a_{s,\ell} \in \mathbb{R}$ equals the earliest time at which ships are available to enter lock ℓ ,

$$b_{s,\ell,\kappa} \in \mathbb{R} = \begin{cases} 0 & \text{if } a_s \leq t_{\ell,\kappa}, \\ a_{s,\ell} - t_{\ell,\kappa} & \text{if } a_s > t_{\ell,\kappa} \end{cases}$$

We have the following constraints linking $a_{s,\ell}$ and $b_{s,\ell,\kappa}$ to the original variables:

$$a_{s,1} = A_s \quad \forall s \in \mathcal{D}, \quad (B.5)$$

$$a_{s,L} = A_s \quad \forall s \in \mathcal{U}, \quad (B.6)$$

$$a_{s,\ell} \geq t_{\ell-1,k} + P_{\ell-1} + S_{\ell-1} \bar{v}_{s,\ell-1} - (\bar{T} - E_{s,\ell}) \sum_{\kappa=1}^{k-1} z_{s,\ell-1,\kappa} \quad \forall s \in \mathcal{D}, \ell \in \mathcal{L} \setminus \{1\}, k \in \mathcal{K}, \quad (B.7)$$

$$a_{s,\ell} \geq t_{\ell+1,k} + P_{\ell+1} + S_{\ell} \bar{v}_{s,\ell} - (\bar{T} - E_{s,\ell}) \sum_{\kappa=1}^{k-1} z_{s,\ell+1,\kappa} \quad \forall s \in \mathcal{U}, \ell \in \mathcal{L} \setminus \{L\}, k \in \mathcal{K}, \quad (B.8)$$

$$b_{s,\ell,k} \geq a_{s,\ell} - t_{\ell,k} \quad \forall s \in \mathcal{S}, \ell \in \mathcal{L}, k \in \mathcal{K} \quad (B.9)$$

$$b_{s,\ell,k} \geq 0 \quad \forall s \in \mathcal{S}, \ell \in \mathcal{L}, k \in \mathcal{K} \quad (B.10)$$

where $E_{s,\ell}$ equals the earliest time at which ships s can arrive at lock ℓ .

Using the $a_{s,\ell}$ variables, constraints (22) and (23) in the original LB-model can be replaced by the following:

$$t_{\ell,k} \geq a_{s,\ell} - M_{k,\ell}^{\text{LB}+} \sum_{\kappa=k+1}^K z_{s,\ell,\kappa} \quad \forall s \in \mathcal{S}, \ell \in \mathcal{L}, k \in \mathcal{K} \quad (\text{B.11})$$

with $M_{k,\ell}^{\text{LB}+} = \bar{T} + P_\ell - (k-1)P_\ell$. Notice that the value $M_{k,\ell}^{\text{LB}+}$ is smaller than the value of M_{k_1,k_2} in the original LB-model.

We now express a number of valid inequalities which further strengthen the formulation of the LB-model.

While the model allows for any number of empty lockages provided that they do not overlap, it is trivial to argue that two consecutive empty movements can be removed from any schedule without affecting the solution value. We can thus prune away many equivalent solutions where two empty lockages are present before all ships have been handled. Note that because the number of non-empty lockages in the optimal solution is not known, we allow for two or more lockages to be empty if all subsequent lockages are also empty. We can impose this by introducing the following constraints:

$$\sum_{s \in \mathcal{S}} (z_{s,\ell,k} + z_{s,\ell,k-1}) \geq \frac{1}{S} \sum_{\substack{s \in \mathcal{S} \\ \kappa > k}} z_{s,\ell,\kappa} \quad \forall \ell \in \mathcal{L}, k \in \mathcal{K} \setminus \{1\} \quad (\text{B.12})$$

A second set of inequalities expresses the intuitively obvious statement that once a ship has arrived at one of the locks, it will enter the first available non-full lockage in the appropriate direction. Note that this does not imply that a ship must move at a high speed so that it arrives on time to enter the first following non-full lockage, but rather that if ship s , travelling e.g. in the upstream direction, has passed lock ℓ and there is a non-full lockage k for lock $\ell+1$ that does not contain s , then $\bar{v}_{s,\ell}$ must be such that s has not yet arrived at lock $\ell+1$ at time $t_{\ell+1,k}$.

The mathematical formulation is as follows:

$$1 - \frac{\sum_{\sigma \in \mathcal{S}} z_{\sigma,\ell,k}}{C_\ell} - b_{s,\ell,k} - \sum_{\kappa=1}^{k-1} z_{s,\ell,\kappa} \leq 0 \quad \forall \ell \in \mathcal{L}, s \in \mathcal{D}, k \in \mathcal{K}$$

$$1 - \frac{\sum_{\sigma \in \mathcal{S}} z_{\sigma,\ell,k}}{C_\ell} - b_{s,\ell,k} - \sum_{\kappa=1}^{k-1} z_{s,\ell,\kappa} \leq 0 \quad \forall \ell \in \mathcal{L}, s \in \mathcal{U}, k \in \mathcal{K}$$

Finally, we introduce additional constraints on the $a_{s,\ell}$ and c_s variables:

$$a_{s,\ell} \geq a_{s,\ell+1} + P_{\ell+1} + S_\ell \bar{v}_{s,\ell} \quad \forall s \in \mathcal{U}, \ell \in \mathcal{L} \setminus \{L\}$$

$$a_{s,\ell} \geq a_{s,\ell-1} + P_{\ell-1} + S_{\ell-1} \bar{v}_{s,\ell-1} \quad \forall s \in \mathcal{D}, \ell \in \mathcal{L} \setminus \{1\}$$

$$c_s \geq a_{s,1} + P_1 \quad \forall s \in \mathcal{U}$$

$$c_s \geq a_{s,L} + P_L \quad \forall s \in \mathcal{D}$$

Adding these valid inequalities to the reformulated LB model improves the performance of the model. We refer to Section 5 for computational experiments to compare the LB and LB+ models.

References

- Boulter, P. G. (2007). ARTEMIS: Assessment and reliability of transport emission models and inventory systems. *Technical report*. TRL limited.
- Caris, A., Janssens, G., & Macharis, C. (2007). A simulation approach to the analysis of intermodal freight transport networks. In *ESM'2007 proceedings*. EUROSIS.
- Fagerholt, K., Laporte, G., & Norstad, I. (2010). Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society*, 61(3), 523–529. doi:10.1057/jors.2009.77.
- Hermans, J. (2008). *Optimalisatie van binnenschepvaart*. KU Leuven (Master's thesis) (in Dutch).
- Hvattum, L. M., Norstad, I., Fagerholt, K., & Laporte, G. (2013). Analysis of an exact algorithm for the vessel speed optimization problem. *Networks*, 62(2), 132–135. doi:10.1002/net.21503.
- Kunst, M. (2013). *Organisation of vessel traffic management centres of the future*. In *Smart rivers conference 2013 abstract booklet*, Liege, Belgium.
- Lübbecke, E., Lübbecke, M. E., & Möhring, R. H. (2014). Ship traffic optimization for the Kiel canal. *Technical Report 4681*. (Optimization Online).
- Luy, M. (2012). Ship lock scheduling. <http://sourceforge.net/projects/lockscheduling/>.
- Nauss, R. M. (2008). Optimal sequencing in the presence of setup times for tow/barge traffic through a river lock. *European Journal of Operational Research*, 187, 1268–1281. doi:10.1016/j.ejor.2006.06.071.
- Passchyn, W., Coene, S., Briskorn, D., Hurink, J.L., Spijksma, F.C.R., & Vanden Berghe, G. (2015). The lockmaster's problem. <https://perswww.kuleuven.be/~u0086328/techReportPreprint.pdf>.
- Petersen, E. R., & Taylor, A. J. (1988). An optimal scheduling system for the Welland Canal. *Transportation Science*, 22, 173–185. doi:10.1287/trsc.22.3.173.
- Prandtstetter, M., Ritzinger, U., Schmidt, P., & Ruthmair, M. (2015). A variable neighborhood search approach for the interdependent lock scheduling problem. In G. Ochoa, & F. Chicano (Eds.), *Evolutionary computation in combinatorial optimization: Vol. 9026*. In *Lecture notes in computer science* (pp. 36–47). doi:10.1007/978-3-319-16468-7_4.
- Psarafitis, H. N., & Kontovas, C. A. (2013). Speed models for energy-efficient maritime transportation: A taxonomy and survey. *Transportation Research Part C: Emerging Technologies*, 26, 331–351. doi:10.1016/j.trc.2012.09.012.
- Qian, J., & Eglese, R. (2014). Finding least fuel emission paths in a network with time-varying speeds. *Networks*, 63(1), 96–106. doi:10.1002/net.21524.
- Ronen, D. (1982). The effect of oil price on the optimal speed of ships. *Journal of the Operational Research Society*, 33(11), 1035–1040. doi:10.2307/2581518.
- 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*, 47, 669–680. doi:10.1016/j.tre.2011.02.005.
- Smith, L. D., Sweeney, D. C., & Campbell, J. F. (2009). Simulation of alternative approaches to relieving congestion at locks in a river transportation system. *Journal of the Operational Research Society*, 60, 519–533.
- Ting, C., & Schonfeld, P. (1999). Effects of speed control on tow travel costs. *Journal of waterway, port, coastal, and ocean engineering*, 125(4), 203–206. doi:10.1061/(ASCE)0733-950X(1999)125:4(203).
- Ting, C., & Schonfeld, P. (2001). Control alternatives at a waterway lock. *Journal of waterway, port, coastal, and ocean engineering*, 127(2), 89–96. doi:10.1061/(ASCE)0733-950X(2001)127:2(89).
- Verstichel, J., De Causmaecker, P., Spijksma, F. C. R., & Vanden Berghe, G. (2014a). Exact and heuristic methods for placing ships in locks. *European Journal of Operational Research*, 235(2), 387–398. doi:10.1016/j.ejor.2013.06.045.
- Verstichel, J., De Causmaecker, P., Spijksma, F. C. R., & Vanden Berghe, G. (2014b). The generalized lock scheduling problem: An exact approach. *Transportation Research E, Logistics and Transportation Review*, 65, 16–34. doi:10.1016/j.tre.2013.12.010.