Transportation Research Part E 65 (2014) 16–34

Contents lists available at ScienceDirect

Transportation Research Part E

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

The generalized lock scheduling problem: An exact approach

Jannes Verstichel [a,*], Patrick De Causmaecker [b], Frits Spieksma [c], Greet Vanden Berghe [c]

[a] KU Leuven, Department of Computer Science, CODeS & iMinds-ITEC, Gebroeders De Smetstraat 1, 9000 Gent, Belgium
[b] KU Leuven, Department of Computer Science, CODeS & iMinds-ITEC, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
[c] KU Leuven, Faculty of Economics and Business, ORSTAT, Naamsestraat 69, 3000 Leuven, Belgium

ARTICLE INFO

Keywords:
Lock scheduling problem
Integer programming
Packing
Scheduling

ABSTRACT

The present paper introduces an integrated approach to solving the generalized lock scheduling problem. Three interrelated sub problems can be discerned: ship placement, chamber assignment and lockage operation scheduling. In their turn, these are closely related to the 2D bin packing problem, the assignment problem and the (parallel) machine scheduling problem respectively. In previous research, the three sub problems mentioned were considered separately, often using (heuristic) interaction between them to obtain better solutions. A mixed integer linear programming model is presented and applied to instances from both inland locks and locks in a tide independent port. The experiments show that small instances incorporating a wide range of real-life constraints can be solved to optimality.

1. Introduction

Growing maritime traffic leads to an ever increasing pressure on the infrastructure of both ports and waterways. Ports are forced to reduce their handle times and heighten their flexibility to maintain or increase their market share. Inland waterways need to reduce the waiting times at their infrastructure to a minimum if they want to increase the share of waterway transport in multimodal transportation. Locks play a key role in these ports and waterways: a vessel must be processed by at least one lock when entering/leaving a tide independent port; barges may pass several locks while transporting cargo on a waterway. A lock consists of at least one chamber, in which ships can be transferred from one water level to another. When more than one chamber is available, the chambers can be paired (i.e. they are operated together) or they can be operated independently. While some locks consist of several identical chambers, others have chambers of different dimensions and properties. Depending on the size of the chamber, one or more ships can be transferred together in a single lockage operation. Processing a ship in a lock may take up to three decisions, each with a significant impact on the quality of service: selecting the chamber that will transfer the ship, determining a position for the ship in that chamber and setting a starting time for the lockage operation. At present, lock scheduling is performed by human experts, with little or no support from optimization software. Therefore, one could conclude there is a scope for introducing specialized decision support optimization software.

A wide range of port related operations has been researched, such as berth or quay crane related problems (Chang et al., 2010; Chen et al., 2012; Lee et al., 2012) and yard operations (Cao et al., 2010; Lee and Kim, 2010; Petering, 2011), but the lock operations attracted only a limited interest. For an overview, we refer to Stahlbock and Voß (2008). Previous research on the lock scheduling problem has mainly focussed on the Upper Mississippi River (UMR). Here, single chamber locks are used to transport tows that are often larger than the chamber itself. Those have to be split into different groups of barges which

* Corresponding author.
E-mail address: jannes.verstichel@cs.kuleuven.be (J. Verstichel).

1366-5545/$ - see front matter
http://dx.doi.org/10.1016/j.tre.2013.12.010

are transferred one at a time, and rejoined for the next part of the journey. Several techniques for solving congestion for the chain of 600-foot locks on the UMR are presented and validated using a simulation tool in Smith and Nauss (2010) and Smith et al. (2011). Optimal sequencing of the tows after a disruption at a lock on the UMR is presented in Nauss (2008). Coene et al. (2011) study the lockmaster's problem, which focusses exclusively on the scheduling sub problem. They identify it as a batch scheduling problem that can, under certain assumptions, be solved in polynomial time using a dynamic programming algorithm. In Verstichel and Vanden Berghe (2009), inland locks with multiple parallel chambers where several ships can be transferred together are considered. To solve this problem, a problem specific heuristic is combined with a late acceptance hill climber. Instances with a single chamber, identical parallel chambers and multiple heterogeneous chambers can be tackled by this approach. Scheduling lockages for a lock with identical parallel chambers is investigated in Verstichel et al. (2011a). The authors identify the problem as the identical parallel machine scheduling problem with sequence dependent setup times and release dates and present a mathematical model. A heuristic solution method is presented for solving the lock scheduling in this specific setting and its performance is compared with that of a first-come-first-served policy. As only the scheduling part is solved, another solution method is needed to place the ships in the chambers. This ship placement problem is tackled for a single chamber type using exact and heuristic methods in Verstichel et al. (in press).

The present paper discusses an integrated approach to solving the generalized lock scheduling problem. This problem consists of a ship placement problem, a variant of two dimensional bin packing, a chamber assignment problem, and a lockage operation scheduling problem, which is closely related to the parallel machine scheduling problem. When the lock has heterogeneous chambers, a chamber type assignment problem has to be solved as well, leading to a total of three sub problems. In previous research, these sub problems were considered separately, often using heuristic interaction between the parts in order to obtain better solutions. We will present an exact approach that solves all three sub problems together, both for an inland setting and in ports.

In Section 2 the generalized lock scheduling problem and its applicability are discussed. A mathematical model for the problem is presented in Section 3, together with several methods for improving the model's scalability. The experiments are described in Section 4, followed by conclusions in Section 5.

2. The lock scheduling problem

When a ship has to be transferred by a lock, a wide range of constraints has to be taken into account. In this section we will describe these constraints for locks that can transfer one or more ships in a single lockage operation. This type of lock is

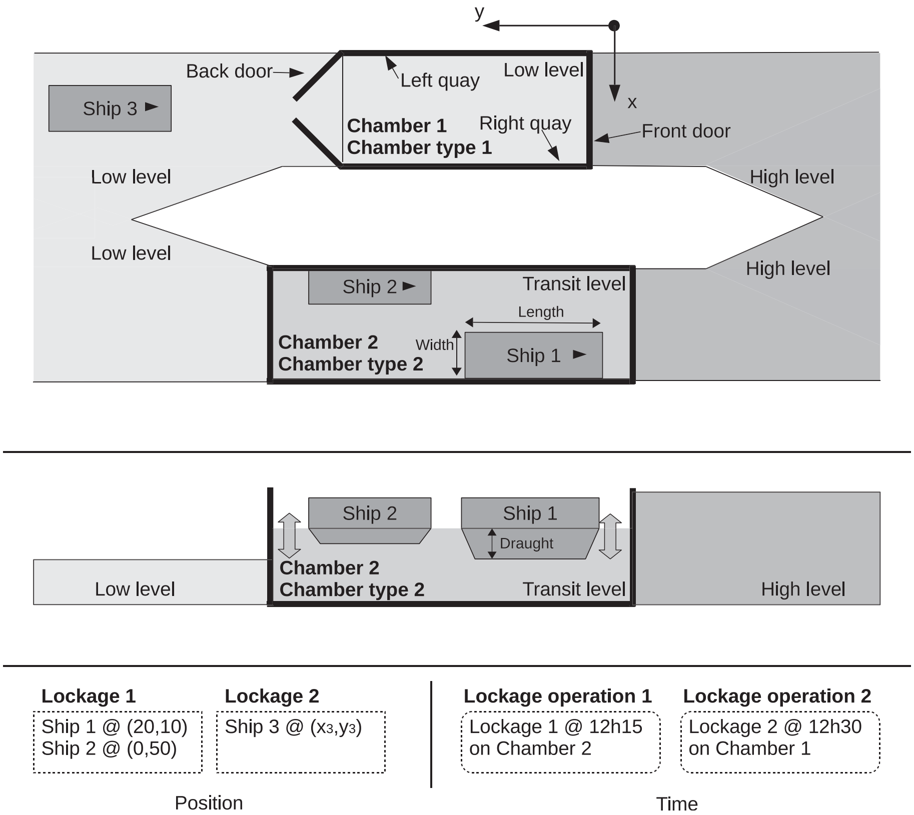

Fig. 1. Visual explanation of the lock scheduling specific terms used throughout the paper.

used in many ports and waterways in Belgium and other Western–European countries. The different constraints and their applicability to port and inland locks will be explained for each sub problem individually. The lock scheduling specific terms that are used throughout this paper are elucidated in Fig. 1. The first part of this figure depicts the top view of a lock with two chambers. Each chamber has two doors (front and back), a left and right quay and a certain chamber type which determines its dimensions and minimal lockage duration. The figure also shows the directions of the x and y axis used to determine a ship's position. The middle part of Fig. 1 shows a side view of the second chamber. The lower part of this figure clarifies the terms lockage and lockage operation. A lockage is basically a solution for the ship placement part of the problem. It defines the ships that are transferred together, and where they are positioned in the chamber. A lockage operation defines the scheduling part of the solution. It contains the timing information for moving the vessels into the lock, changing the water level, and enabling the ships to leave the chamber.

With respect to the chamber type assignment problem, the constraints concern the dimensions of the ship and the chamber type. The ship must be smaller than the chamber type, both with respect to width, length and draught.

The ship placement problem is subject to quite a few more constraints, which are thoroughly discussed in Verstichel et al. (in press), and summarized in Figs. 2 and 3. First there are the standard 2D bin packing constraints, violated in Fig. 2(a) and (b). Rotating ships is also prohibited (2(c)). The other subfigures represent violations of the mooring constraint. When a ship is transferred in a lockage, it must be secured either to the right or to the left quay (i.e. the right or left hand side of the chamber). This can be achieved by mooring a ship directly to the quay, or by mooring it to a larger ship that is already attached to either quay (violated in Fig. 2(d) and (e)). An additional limitation when mooring a smaller ship to a larger ship requires the smaller ship to be contained within the length of the larger ship (Fig. 2(f)). When several types of vessels can be processed together, limitations may occur as to which vessel types a ship can moor to. In ports, for example, sea ships can only be moored directly to the quay, and no other ships are allowed to moor to sea ships. This is visualised in Fig. 2(g), where the barge may not moor to the sea ship, although this would be feasible for the other constraints.

Another set of constraints deals with the typical safety distances of the ship placement problem. Some manoeuvring space must be allocated to ships, preventing collisions while sailing in and out of the chamber, and in case of a possible minor accident during the lockage operation. While these safety distances can be considered equal for all ship pairs in an inland setting, traffic in sea ports requires a more individualised approach. The safety distance constraints for different ship pairs are visualised in Fig. 3. Based on the dimensions and type of both ships and their tugboat requirements, a minimal lateral safety distance can be calculated (Fig. 3(a)). When both ships use tugboats a sufficiently large lateral distance or 'corridor' must be present to enable the tugboats to sail between the ships when leaving the chamber before the lockage operation starts. The longitudinal safety distance between ships depends on their type and dimensions (Fig. 3(b)). The distance between each ship and the doors of the chamber (Fig. 3(c)) determines a third set of distance constraints. These are required both for safety (i.e. to avoid collisions with the doors) and practical reasons (position of the mooring poles to which the ship can moor). This distance is dependent on both the chamber type and the ship's dimensions.

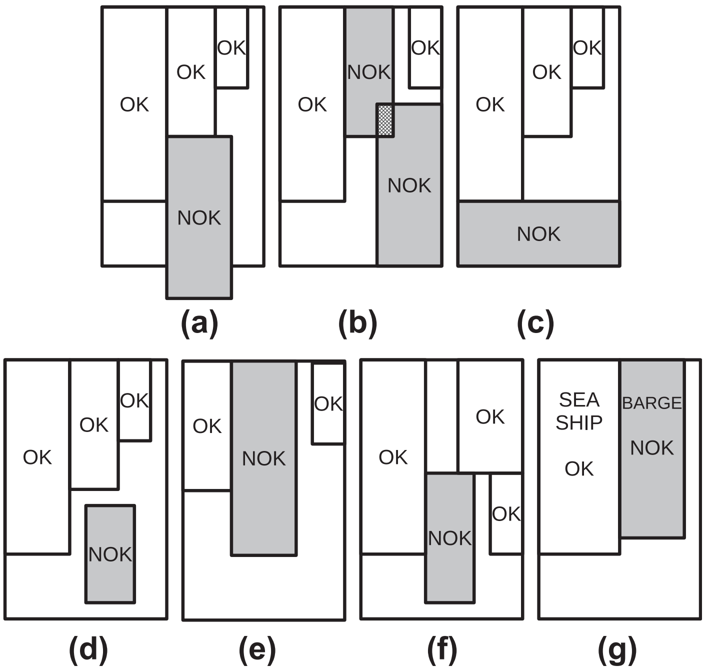

Fig. 2. A visual representation of the hard constraints for the ship placement problem. (OK: ship placed in a correct way, NOK: ship not placed correctly. Removing a NOK ship would make the packing feasible.)

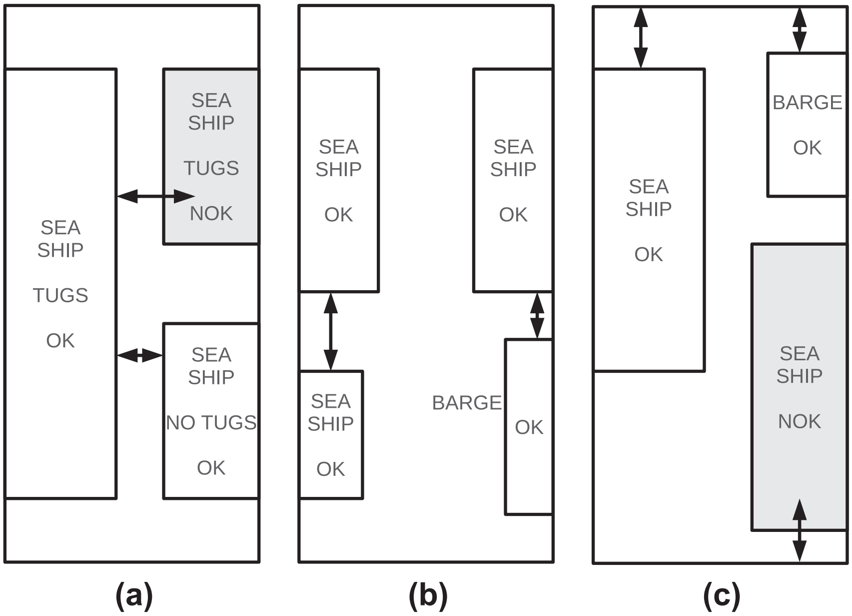

Fig. 3. A visual representation of the safety distance constraints. Subfigure (a) shows the 'width' safety distances between ships, (b) shows the 'length' safety distances between ships, and (c) shows the safety distances between ships and the chamber doors. (OK: ship placed in a correct way, NOK: ship not placed correctly.)

The lockage operation scheduling sub problem can be modeled as a parallel machine scheduling problem where the chambers are machines and the lockages are jobs. Further, there are sequence dependent setup times, release dates, machine eligibility restrictions and time windows. This results in the following standard notation for the scheduling sub problem: $Q_m|r_j, s_{jk}, M_j|\sum w_jC_j$.

There is one additional constraint that enforces a first come first served policy with respect to ship arrival times, either for all ships or for a subset. As multiple chambers of different chamber types are allowed, the machines may have different speeds. The sequence dependent setup times are required for two reasons. Firstly, when two lockages in the same direction are processed directly after each other on the same chamber, an empty lockage in the opposite direction must be processed in between these lockages, leading to a setup time between the two lockages. Secondly, when dealing with large ships, approach times must be taken into account for each vessel entering the lock, as well as a departure time for clearing the lock. Furthermore, ships with tug boats also require some additional time for (dis) engaging the tug boats before and after the actual lockage operation. We define the pre-processing time of a ship as the time required for the ship to approach the lock, moor to the quay or to another ship, disengage the tug boats (if any) and allow for the tug boats to leave the lock. The post-processing time is the time required for the tug boats (if any) to enter the lock and engage the ship, and for the ship to unmoor and clear the lock. Both the approach time and departure time are defined with respect to a given 'coordination point' near the lock. As each lockage operation is feasible on one chamber type only, we need machine eligibility restrictions. Otherwise, a lockage defined for a large chamber type might be processed on a chamber of a much smaller type. The processing times can be ship dependent, for example when a lot of barges are processed in one lockage operation, an additional processing time must be taken into account. The usage of time windows corresponds to the tidal windows for large sea vessels that can only approach or leave a port during high tide.

Depending on the type of lock and the lock policies, one or more of the aforementioned constraints may be removed from the problem. Examples are the constraints with respect to ship types (inland locks), draught (inland locks) and first come first served (port locks).

Parameters and sets:

N:	Set of ships, indexed by $i, j \in \{1, \ldots, n\}$.
w_i, l_i, d_i:	Width, length and draught of ship i.
dF_i, dB_i:	Minimal distance between ship i and the front, back of the chamber.
sW_{ij}, sL_{ij}:	Minimal safety distance between ships i and j when they are lying next to, or behind each other.
$MOOR_i$:	Set of ships to which ship i can moor.
$pre_i, post_i$:	Pre- and post-processing times of ship i.

(*continued on next page*)

p_i:	Additional processing time induced by ship i.
r_i:	Arrival time of ship i at the coordination point.
$c_{min,i}, c_{max,i}$:	Lower and upper limit on the completion time of ship i.
wct_i:	Weight of the completion time of ship i.
$TIDAL$:	Set of all ships with a tidal window.
$FCFS$:	Set of all ships that must be processed first-come-first-served with respect to their arrival times.
$TYPES$:	Set of available chamber types at the lock, indexed by t.
M:	Set of the lockages, indexed by k, l.
M_t:	Subset of M, reserved for lockages performed on the chamber (s) of type t.
U_t:	Set of physical chambers of type t, indexed by u.
W_t, L_t, D_t:	Width, length and draught of the chamber (s) of type t.
W, T, D:	Maximal width, length and draught over all chambers.
$ship_0$:	The left quay of each chamber.
$ship_{n+t}$:	The right quay of each chamber of type t.
p:	Minimal duration of a lockage.
$setup_t$:	Minimal setup time between two lockages (in the same direction) when they are processed by the same chamber of type t.
C_{max}:	Upper bound for the completion time (Big_M constant, is reduced using heuristics).
K_1, K_2, K_3:	Weights of the number of lockage operations, weighted completion time and maximum tardiness.

Variables:

$\boldsymbol{x_i}, \boldsymbol{y_i}$:	Integer variables that define the x and y position of ship i (front left corner).
$\boldsymbol{left_{ij}}$:	Binary variable, $left_{ij} = 1 \Rightarrow$ship i is completely to the left of ship j.
$\boldsymbol{b_{ij}}$:	Binary variable, $b_{ij} = 1 \Rightarrow$ ship i is completely behind ship j.
$\boldsymbol{ml_{ij}}, \boldsymbol{mr_{ij}}$:	Binary variables, 1 when ship i is moored to ship j's left, respectively right, 0 otherwise.
$\boldsymbol{z_k}$:	Binary variable that indicates whether lockage k is used (1) or not (0).
$\boldsymbol{f_{ik}}$:	Binary variable that indicates whether ship i is processed in lockage k (1) or not (0).
$\boldsymbol{v_{ij}}$:	Binary variable, 1 when ship i and j are processed in the same lockage, 0 otherwise.
$\boldsymbol{c_i}$:	Departure time of ship i (completion time of the lockage).
$\boldsymbol{C_k}$	Completion time of lockage k.
$\boldsymbol{P_k}$	Processing time of lockage k.
$\boldsymbol{s_{kl}}$:	Ship dependent setup time for processing lockage l immediately after lockage k in the same chamber.
$\boldsymbol{seq_{kl}}$:	Binary variable, 1 when lockage k precedes lockage l in the same chamber, 0 otherwise.
$\boldsymbol{proc_{ku}}$:	Binary variable that indicates whether lockage k is processed in chamber u (1) or not (0).
$\boldsymbol{T}_{max}$:	Maximum lock transit time over all ships.

3. MILP model

This section introduces a mixed integer linear programming model for the generalized lock scheduling problem. To improve the readability of this section, the presented model assumes that all ships travel upstream. The downstream parameters, variables and constraints (which are identical to their upstream counterparts) were omitted. The complete model including downstream ships is added in Appendix A.

The ship placement and chamber type assignment parts are based on the model for 2D bin packing with multiple bin sizes of Pisinger and Sigurd (2005). Constraints (2)–(31) define the chamber assignment and ship placement parts of the problem. The model contains additional ships that represent the left ($ship_0$) and right ($ship_{n+t}$) quays of each chamber of type $t \in T$ (see Fig. 4). These 'quay' ships enable a straightforward implementation of the mooring constraints as the quays can now be seen as large ships with a fixed position to which a ship can be moored. The scheduling part is based on the model for early/tardy scheduling with sequence dependent setup times on uniform parallel machines by Balakrishnan et al. (1999). It is defined by constraints (32)–(44). Constraints (45)–(57) define the variables.

The objective is a weighted sum of the number of lockage operations, the weighted completion times of the ships and the maximum tardiness. These weights can be changed depending on the problem at hand. In case of a drought, for example, a high weight can be given to the first part of the objective, making sure the number of lockages used for transferring the ships is minimal. When a large queue of ships is waiting at the lock, the weight of the maximal tardiness can be increased, so that none of the ships has to wait for an excessive amount of time.

$$minimize\, K_1 \sum_{k \in M} \boldsymbol{z_k} + K_2 \sum_{i \in N} wct_i \boldsymbol{c_i} + K_3 \boldsymbol{T}_{max} \tag{1}$$

s.t.

The first block of constraints (2)–(31) models the ship placement part of the upstream-only lock scheduling problem. Constraints (2)–(4) ensure that two ships transferred in the same lockage do not overlap (Pisinger and Sigurd, 2005).

$$\boldsymbol{left_{ij}} + \boldsymbol{left_{ji}} + \boldsymbol{b_{ij}} + \boldsymbol{b_{ji}} + (1 - \boldsymbol{f_{ik}}) + (1 - \boldsymbol{f_{jk}}) \geqslant 1 \tag{2}$$

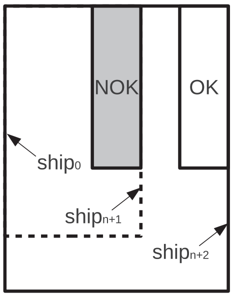

Fig. 4. A visual representation of the constraints for mooring to the right quay of the different chamber types. The ship is transferred in a chamber of type 2, therefore it is not allowed to moor to the right quay of a chamber of type 1 ($ship_{n+1}$).

$$\forall i < j, \quad i,j \in N,\; k \in M$$

$$\boldsymbol{x_i} - \boldsymbol{x_j} + W\boldsymbol{left_{ij}} \leqslant W - w_i, \quad \forall i,j \in N \tag{3}$$

$$\boldsymbol{y_i} - \boldsymbol{y_j} + L\boldsymbol{b_{ij}} \leqslant L - l_i, \quad \forall i,j \in N \tag{4}$$

In constraint (5) we model the safety distance between two adjacent ships, see Fig. 3(a). This safety distance depends on both ships, and on their tug boat requirements. The minimal safety distance between two vessels both requiring tug boats is larger than that between two ships of which only one needs tug boats.

$$\boldsymbol{x_j} - \boldsymbol{x_i} + (W + sW_{ij})(1 - \boldsymbol{left_{ij}} + \boldsymbol{b_{ij}}) \geqslant w_i + sW_{ij}, \quad \forall i,j \in N \tag{5}$$

When ships are laying behind each other, constraint (6) ensures that the safety distance requirements are met. This distance depends on the dimensions of both ships, and on the ship types, see Fig. 3(b).

$$\boldsymbol{y_j} - \boldsymbol{y_i} + (L + sL_{ij})(1 - \boldsymbol{b_{ij}} + \boldsymbol{left_{ij}}) \geqslant l_i + sL_{ij}, \quad \forall i,j \in N \tag{6}$$

Each ship must be placed within the dimensions of the chamber type in which it will be transferred, which is modeled by constraints (7)–(9) (Pisinger and Sigurd, 2005). These constraints are examples of how the $\boldsymbol{f_{ik}}$ variables connect the $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$ variables to a specific lockage.

$$\boldsymbol{x_i} + w_i \leqslant W_t + (1 - \boldsymbol{f_{ik}})W, \quad \forall i \in N,\; t \in TYPES,\; k \in M_t \tag{7}$$

$$\boldsymbol{y_i} + l_i \leqslant L_t + (1 - \boldsymbol{f_{ik}})L, \quad \forall i \in N,\; t \in TYPES,\; k \in M_t \tag{8}$$

$$d_i \leqslant D_t + (1 - \boldsymbol{f_{ik}})D, \quad \forall i \in N,\; t \in TYPES,\; k \in M_t \tag{9}$$

Constraints (10) and (11) ensure that the minimal safety distance between a ship and the front and back door of the chamber is respected. These safety distances depend on the ship's type and dimensions. The constraints are depicted in Fig. 3(c).

$$\boldsymbol{y_i} \geqslant dF_i, \quad \forall i \in N \tag{10}$$

$$\boldsymbol{y_i} + l_i \leqslant L_t - dB_i + (1 - \boldsymbol{f_{ik}})L, \quad \forall i \in N,\; t \in TYPES,\; k \in M_t \tag{11}$$

Constraint (12) ensures that each ship is transferred by exactly one lockage.

$$\sum_{k \in M} \boldsymbol{f_{ik}} = 1, \quad \forall i \in N \tag{12}$$

Constraint (13) models that each lockage that transfers a ship must be used (Pisinger and Sigurd, 2005).

$$\boldsymbol{f_{ik}} \leqslant \boldsymbol{z_k}, \quad \forall i \in N,\; k \in M \tag{13}$$

We now present the lock scheduling specific mooring constraints (14)–(31).

Constraints (14) and (15) model that ship i can only moor to ship j's left side when it is contained within ship j's length.

$$\boldsymbol{y_j} - \boldsymbol{y_i} \leqslant (1 - \boldsymbol{ml_{ij}})L, \quad \forall i \in N,\; j \in MOOR_i \tag{14}$$

$$\boldsymbol{y_i} - \boldsymbol{y_j} \leqslant l_j - l_i + (1 - \boldsymbol{ml_{ij}})L, \quad \forall i \in N,\; j \in MOOR_i \tag{15}$$

An additional requirement for mooring ship i to ship j's left side, is that both ships are adjacent. This is modeled by constraints (16) and (17).

$$\boldsymbol{x_j} - \boldsymbol{x_i} \leqslant w_i + (1 - \boldsymbol{ml_{ij}})W, \quad \forall i \in N,\; j \in MOOR_i \tag{16}$$

$$\boldsymbol{x_j} - \boldsymbol{x_i} \geqslant w_i - (1 - \boldsymbol{ml_{ij}})W, \quad \forall i \in N,\; j \in MOOR_i \tag{17}$$

A ship can also moor to the left side of the right quay (modeled as $ship_{n+t}$) of the chamber (of type t) in which it is transferred. Mooring a ship to the right quay is modeled by constraints (18) and (19), keeping in mind that $ship_{n+t}$, ($t \in TYPES$) denote the right quays for the different chamber types. Constraint (20) ensures that a ship cannot be moored to the right quay of another chamber type than the one it is transferred in. When, for example, two chamber types are available at the lock, there will be two right quay ships ($n+1$ and $n+2$). A ship transferred in chamber type 2 may moor to $ship_{n+2}$, but cannot moor to $ship_{n+1}$ (the right quay of chamber type 1). These constraints are depicted in Fig. 4.

$$\boldsymbol{x_{n+t}} - \boldsymbol{x_i} \leqslant w_i + (1 - \boldsymbol{ml_{i,n+t}})W, \quad \forall i \in N,\ t \in TYPES \tag{18}$$

$$\boldsymbol{x_{n+t}} - \boldsymbol{x_i} \geqslant w_i - (1 - \boldsymbol{ml_{i,n+t}})W, \quad \forall i \in N,\ t \in TYPES \tag{19}$$

$$\boldsymbol{ml_{i,n+t}} \leqslant \sum_{k \in M_t} \boldsymbol{f_{ik}}, \quad \forall i \in N,\ t \in TYPES \tag{20}$$

Constraints (21) and (22) model that ship i can only moor to ship j's right side when it is contained within ship j's length.

$$\boldsymbol{y_j} - \boldsymbol{y_i} \leqslant (1 - \boldsymbol{mr_{ij}})L, \quad \forall i \in N,\ j \in MOOR_i \tag{21}$$

$$\boldsymbol{y_i} - \boldsymbol{y_j} \leqslant l_j - l_i + (1 - \boldsymbol{mr_{ij}})L, \quad \forall i \in N,\ j \in MOOR_i \tag{22}$$

An additional requirement for mooring ship i to ship j's right side, is that both ships are adjacent. This is modeled by constraints (23) and (24).

$$\boldsymbol{x_j} - \boldsymbol{x_i} \leqslant -w_j + (1 - \boldsymbol{mr_{ij}})W, \quad \forall i \in N,\ j \in MOOR_i \tag{23}$$

$$\boldsymbol{x_j} - \boldsymbol{x_i} \geqslant -w_j - (1 - \boldsymbol{mr_{ij}})W, \quad \forall i \in N,\ j \in MOOR_i \tag{24}$$

A ship can also moor to the right side of the left quay (modeled as $ship_0$) of the chamber by which it is transferred. Mooring a ship to the left quay is modeled by constraints (25) and (26), keeping in mind that $ship_0$ is the left quay for all chambers (see Fig. 4).

$$\boldsymbol{x_0} - \boldsymbol{x_i} \leqslant w_i + (1 - \boldsymbol{ml_{i,0}})W, \quad \forall i \in N \tag{25}$$

$$\boldsymbol{x_0} - \boldsymbol{x_i} \geqslant w_i - (1 - \boldsymbol{ml_{i,0}})W, \quad \forall i \in N \tag{26}$$

All ships have to be moored and this is modeled by constraint (27). A ship can be moored to another ship, the left quay or one of the right quays.

$$\sum_{j \in N: j \neq i} (\boldsymbol{ml_{ij}} + \boldsymbol{mr_{ij}}) + \boldsymbol{mr_{i,0}} + \sum_{t \in TYPES} \boldsymbol{ml_{i,n+t}} \geqslant 1, \quad \forall i \in N \tag{27}$$

When $ship_i$ is moored to $ship_j$, $ship_j$ cannot be moored to $ship_i$ and vice versa. This only happens when ships have the same length, at which point they could moor to one another, leaving both ships unattached to the quay. Constraint (28) disables this kind of 'fake' mooring.

$$\boldsymbol{ml_{ij}} + \boldsymbol{mr_{ji}} \leqslant 1, \quad \forall i \neq j,\ i,j \in N \tag{28}$$

When $ship_i$ and $ship_j$ are in different lockages, they cannot moor to each other. Constraints (29)–(31) ensure that the mooring constraints are only valid for two ships that are transferred in the same lockage.

$$\boldsymbol{f_{ik}} - \boldsymbol{f_{jk}} \leqslant (1 - \boldsymbol{v_{ij}}), \quad \forall i < j,\ i,j \in N,\ k \in M \tag{29}$$

$$\boldsymbol{f_{jk}} - \boldsymbol{f_{ik}} \leqslant (1 - \boldsymbol{v_{ij}}), \quad \forall i < j,\ i,j \in N,\ k \in M \tag{30}$$

$$\boldsymbol{ml_{ij}} + \boldsymbol{mr_{ij}} + \boldsymbol{ml_{ji}} + \boldsymbol{mr_{ji}} \leqslant \boldsymbol{v_{ij}}, \quad \forall i < j,\ i,j \in N \tag{31}$$

The second block of constraints (32)–(44) models the scheduling part of the lock scheduling problem with only upstream ships.

Constraints (32) and (33) ensure that the lockage completion time for $ship_i$ is equal to the completion time of the lockage in which it is transferred. C_{max} is a Big_M constant and should be sufficiently large.

$$\boldsymbol{c_i} \geqslant C_{max}(\boldsymbol{f_{ik}} - 1) + \boldsymbol{C_k}, \quad \forall i \in N,\ k \in M \tag{32}$$

$$\boldsymbol{c_i} \leqslant C_{max}(1 - \boldsymbol{f_{ik}}) + \boldsymbol{C_k}, \quad \forall i \in N,\ k \in M \tag{33}$$

When a ship is restricted by a tidal window, it has to be processed before its tidal window ends. This is modeled by constraint (34).

$$c_{min,i} \leqslant \boldsymbol{c_i} \leqslant c_{max,i}, \quad \forall i \in TIDAL \tag{34}$$

Constraint (35) ensures that the processing time of upstream lockage k is equal to the standard lockage duration for the lockage's chamber type, increased by the additional processing times of each ship that is transferred by that lockage.

$$\boldsymbol{P_k} \geqslant p_t \boldsymbol{z_k} + \sum_{i \in N} \boldsymbol{f_{ik}} p_i, \quad \forall k \in M_t,\ t \in TYPES \tag{35}$$

The setup time between lockage operations k and l depends on their direction and on the ships that are processed. This is modeled in constraint (36). When k and l process ships in the same direction, the minimal setup time will be equal to the empty lockage operation time for the used chamber type, $setup_t$. This is the time required for changing the water level when the chamber is empty. If the lockage operations would be in opposite directions, the value would be smaller, as the water level does not need to be changed in that case. Here, we can use $setup_t$ instead of $setup_{kl}$ as all lockages are in the upstream direction. The next part of the constraint states that both the post processing times of lockage operation k and the pre-processing times of lockage operation l are taken into account.

$$\boldsymbol{s_{kl}} \geqslant setup_t + \sum_{i \in N} \boldsymbol{f}_{ik} post_i + \sum_{i \in N} \boldsymbol{f}_{il} pre_i \quad \forall l \neq k,\ k,l \in M_t,\ t \in TYPES \tag{36}$$

Constraint (37) ensures that every used lockage is assigned to one of the physical chambers corresponding to the lockage's chamber type (Balakrishnan et al., 1999). Lockages that do not transfer any ships, are not assigned to a physical chamber.

$$\sum_{u \in U_t} \boldsymbol{proc_{ku}} = \boldsymbol{z_k}, \quad \forall k \in M_t,\ t \in TYPES \tag{37}$$

Constraint (38) ensures that two lockages can be sequenced after each other iff they are processed by the same physical chamber (Balakrishnan et al., 1999).

$$\boldsymbol{proc_{ku}} + \sum_{v \in U_t, v \neq u} \boldsymbol{proc_{lv}} + \boldsymbol{seq_{kl}} \leqslant 2, \quad \forall l > k,\ k,l \in M_t,\ u \in U_t,\ t \in TYPES \tag{38}$$

The completion times of lockages that are processed by the same physical chamber are modeled using constraints (39) and (40) (Balakrishnan et al., 1999).

$$\boldsymbol{C_l} - \boldsymbol{C_k} + 2C_{max}(3 - \boldsymbol{seq_{kl}} - \boldsymbol{proc_{ku}} - \boldsymbol{proc_{lu}}) \geqslant \boldsymbol{P_l} + \boldsymbol{s_{kl}} \quad \forall k < l,\ k,l \in M_t,\ u \in U_t,\ t \in TYPES \tag{39}$$

$$\boldsymbol{C_k} - \boldsymbol{C_l} + 2C_{max}(2 + \boldsymbol{seq_{kl}} - \boldsymbol{proc_{ku}} - \boldsymbol{proc_{lu}}) \geqslant \boldsymbol{P_k} + \boldsymbol{s_{lk}} \quad \forall k < l,\ k,l \in M_t,\ u \in U_t,\ t \in TYPES \tag{40}$$

Lockage k cannot start before all ships in the lockage have arrived at the coordination point (Constraints 41).

$$\boldsymbol{C_k} - \boldsymbol{P_k} \geqslant \boldsymbol{f}_{ik} r_i, \quad \forall i \in N,\ k \in M \tag{41}$$

Constraint (42) ensures that a lockage is only scheduled when it transfers at least one ship.

$$\boldsymbol{z_k} \leqslant \sum_{i \in N} \boldsymbol{f}_{ik}, \quad \forall k \in M \tag{42}$$

The maximum lock transit time over all ships is defined by constraint (43).

$$\boldsymbol{T}_{max} \geqslant \boldsymbol{c_i} - r_i, \quad \forall i \in N \tag{43}$$

Constraint (44) can be used to implement priority due to operational, economic or safety policies. It ensures that the transfer of $ship_j$ may not be completed before that of $ship_i$, when $ship_i$ and $ship_j$ have a first-come-first-served restriction ($i < j$).

$$\boldsymbol{c_i} \leqslant \boldsymbol{c_j}, \quad \forall i < j,\ i,j \in FCFS \tag{44}$$

Constraints (45)–(57) formulate bounds and integrality constraints on the variables.

$$\boldsymbol{left_{ij}}, \boldsymbol{b_{ij}}, \boldsymbol{ml_{ij}}, \boldsymbol{mr_{ij}} \in \{0,1\}, \quad \forall i,j \in N \tag{45}$$

$$\boldsymbol{v_{ij}} \in \{0,1\}, \quad \forall i < j,\ i,j \in N \tag{46}$$

$$0 \leqslant \boldsymbol{x_i} \leqslant W, \quad \forall i \in N \tag{47}$$

$$0 \leqslant \boldsymbol{y_i} \leqslant L, \quad \forall i \in N \tag{48}$$

$$0 \leqslant \boldsymbol{c_i} \leqslant C_{max}, \quad \forall i \in N \tag{49}$$

$$\boldsymbol{f}_{ik} \in \{0,1\}, \quad \forall i \in N,\ k \in M \tag{50}$$

$$0 \leqslant \boldsymbol{C_k} \leqslant C_{max}, \quad \forall k \in M \tag{51}$$

$$\boldsymbol{P_k} \geqslant 0, \quad \forall k \in M \tag{52}$$

$$\boldsymbol{z_k} \in \{0,1\}, \quad \forall k \in M \tag{53}$$

$$0 \leqslant \boldsymbol{s_{kl}} \leqslant C_{max}, \quad \forall k,l \in M \tag{54}$$

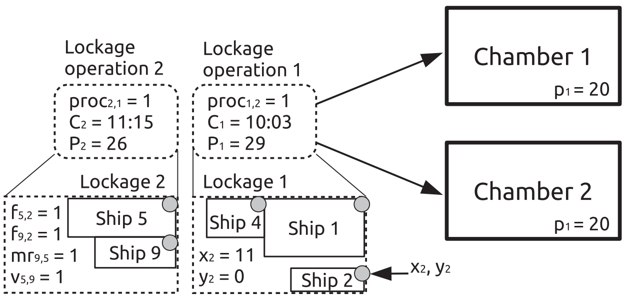

Fig. 5. A visual representation of how the main MILP variables are mapped to a solution for the lock scheduling problem.

$$\boldsymbol{seq_{kl}} \in \{0,1\}, \quad \forall k, l \in M \tag{55}$$

$$\boldsymbol{proc_{ku}} \in \{0,1\}, \quad \forall k \in M_t, \ u \in U_t, \ t \in TYPES \tag{56}$$

$$\boldsymbol{T}_{max} \geqslant 0 \tag{57}$$

The above model describes the generalized lock scheduling problem. It can be applied to both locks in ports and locks on inland waterways.

Some assumptions are made with respect to safety distances and pre/post-processing times. Safety distances are independent of the chamber type, and are defined only by the interaction between ships that are transferred together. We assume that the weather conditions are constant for the entire lock operation. This allows us to consider pre/post-processing times that depend only on the ships. When considering the pre/post-processing times, a distinction can be made between ships with and ships without tug boats. Ships with tug boats will always be the first to be positioned in the chamber for any given lockage and their pre/post-processing times can be considered a constant value for a given weather condition. Ships without tug boats will be positioned later, and their pre/post-processing times may depend on the number of ships already in the chamber and on the remaining free space. In the model, we consider these much smaller times to be constant as well, using averages based on the experience of lock masters.

Depending on the type of lock and on the traffic, several of the constraints in the model may be removed or reduced. When all ships have a draught that is smaller than the smallest chamber draught, constraint (9) is obsolete. When there are no first come first served limitations, constraint (44) can be removed.

Another example of redundant constraints applies to inland locks. The processing times at these locks are not ship dependent, which greatly reduces the impact of constraints (35) and (36).

The number of ships in the $MOOR_i$ sets varies greatly between ship types. For ocean going vessels, for example, $MOOR_i$ will contain only the quay ships. Barges, on the other hand, may moor to any other barge, so their $MOOR_i$ will also contain all barges that are at least as long as the barge itself.

When constructing the solution for a lock scheduling instance, the values of five variable groups need to be retrieved from the MILP solution:

- The lockage in which each ship is transferred ($\boldsymbol{f}_{\boldsymbol{ik}}$).
- The chamber used to process this lockage ($\boldsymbol{proc_{ku}}$).
- The ship's position in this chamber ($\boldsymbol{x_i}, \boldsymbol{y_i}$).
- The time at which the lockage operation ends ($\boldsymbol{C_k}$ or $\boldsymbol{c_i}$).
- The lockage operation duration ($\boldsymbol{P_k}$), enabling the calculation of the lockage start time.

Given these variables' values, the proposed lock scheduling solution can be constructed unambiguously.

Fig. 5 shows how the main MILP variables are mapped to a solution for the lock scheduling problem. Each lockage contains at least one ship, each of which is placed at a specific location, defined by the x_i and y_i variables (gray dots). An example of the interaction between ships in the same lockage is shown through the $mr_{9,5}$ and $v_{5,9}$ variables, which define how ship 9 is moored to ship 5, and that ships 5 and 9 are transferred together. The $f_{5,2}$ variable tells us that ship 5 is transferred in upstream lockage 2. When looking at the scheduling part, the figure visualizes how the $proc_{ku}$ variables map each lockage to a lockage operation that is transferred by a physical chamber. For example: lockage 1 is transferred in a lockage operation in physical chamber 2 ($proc_{1,2} = 1$) starting at 09:34 ($C_1 - P_1$).

4. Experiments

We have applied the mixed integer linear programming model for the generalized lock scheduling problem to several instances generated from historical data. These instances can be split into two different sets, one for an inland setting, and one for locks at a port. All experiments were run on an Intel Core2Duo E8400 cpu with 4 GB memory running Windows XP SP3 and Gurobi 4.5.2 under an academic license. The Gurobi solver was set to time-out after 12 h of calculation time, returning the best solution found.

4.1. Model speed ups

Several adaptations can be made to the model to reduce the size of the solution space and speed the solution process up. Firstly, the $\boldsymbol{z_k}$ and $\boldsymbol{v_{ij}}$ variables can be defined as continuous variables with values between 0 and 1. The nature of the model is such that omitting the zero-one constraint on the $\boldsymbol{z_k}$ and $\boldsymbol{v_{ij}}$ variables does not lead to incorrect fractional values in feasible solutions. Indeed, given that in a feasible solution the value of the $\boldsymbol{f}_{ik}$ variables is either 0 or 1, Constraints (13) and (42) force binary values for the $\boldsymbol{z_k}$ variables. Whenever Constraint (31) is active (otherwise $\boldsymbol{v_{ij}}$ is irrelevant) for a specific *i*-*j* combination, the binary values of $\boldsymbol{f}_{ik}, \boldsymbol{mr_{ij}}$ and $\boldsymbol{ml_{ij}}$ and Constraints (29)–(31) force a binary value for the corresponding $\boldsymbol{v_{ij}}$ variable. For a large set of instances that were solved in less than 12 h, the introduction of these continuous variables reduces the computation time by an average of 33.75%.

A second optimization can be obtained by forcing an ordering in the lockages and lockage operation completion times. The first constraint (Constraint 58) states that lockage l of type t can only be activated if lockage $k < l$ of the same type is already active. This symmetry breaking constraint strongly reduces computation time when the difference between the optimal solution and the lower bound is significant. The second constraint states that the completion time of lockage operation k of type t must be less than or equal to the completion time of lockage operation l of type t, where $k < l$ (Constraint 59). Both restrictions are modeled through transitive constraints, which only put limitations on the subsequent lockage (operation) $k+1$ of each lockage (operation) k. The addition of these transitive constraints decimated computation time on all tested instances, with even larger reductions on some large instances.

$$\boldsymbol{z_{k+1}} \leqslant \boldsymbol{z_k}, \quad \forall k \in M_t, \ \forall t \in T \tag{58}$$

$$\boldsymbol{C_k} \leqslant \boldsymbol{C_{k+1}}, \quad \forall k \in M_t, \ \forall t \in T \tag{59}$$

The third speedup is obtained for instances where a first-come-first-served policy is used with respect to the ship arrival times. This is the case for the locks on the canal called Albertkanaal, where the current policy requires a first-come-first-served order when transferring ships through the lock. Here, we can add an ordering constraint with respect to the lockage index of each ship. When considering a single chamber lock, this constraint states that for any two ships i and j ($i < j$), the index (k) of the lockage that transfers ship i must be less than or equal to the index (l) of the lockage that transfers ship j (Constraint 60). In the case of multiple identical chambers, this constraint is relaxed, requiring that $k \leqslant l + nrOfChambers$ (Constraint 61). This relaxation is necessary as ships can be processed at the same time, but in different chambers. Fig. 6 shows an example where the original constraint (60) would exclude the, otherwise feasible, solution. Using the original constraint, ship PRIMA and GERONIMO can be processed together iff WISDOM is transferred in the same lockage. As the presented solution does not violate the first-come-first-served principle, this constraint should be replaced by constraint (61) to avoid excluding such feasible solutions. When multiple chamber types are available at the lock, the constraint is once again changed to make sure that the first-come-first-served rule is only applied to ships that are processed in the same chamber type. This constraint is added in constraint (62). By using the *FCFS* sets instead of all the ships, these constraints can also be applied when only a subset of the ships are subject to first-come-first-served constraints.

$$\sum_{k<c,k\in M} (\boldsymbol{f}_{ik} - \boldsymbol{f}_{jk}) \geqslant 0, \quad \forall i<j, \ i,j \in N, \ c \in M \tag{60}$$

$$\sum_{k<c+|U|,k\in M} \boldsymbol{f}_{ik} - \sum_{k=1}^{c} \boldsymbol{f}_{jk} \geqslant 0, \quad \forall i<j, \ i,j \in N, \ c \in M \tag{61}$$

$$-2\sum_{k<c,k\in M_t} \boldsymbol{f}_{jk} - \sum_{k>c,k\in M_t} (\boldsymbol{f}_{ik} + \boldsymbol{f}_{jk}) + \sum_{c+1<k<c+|U_t|,k\in M_t} \boldsymbol{f}_{ik} \geqslant -2 \quad \forall i<j, \ i,j \in N, \ c = 1,\ldots,|M_t|, \ t \in TYPES \tag{62}$$

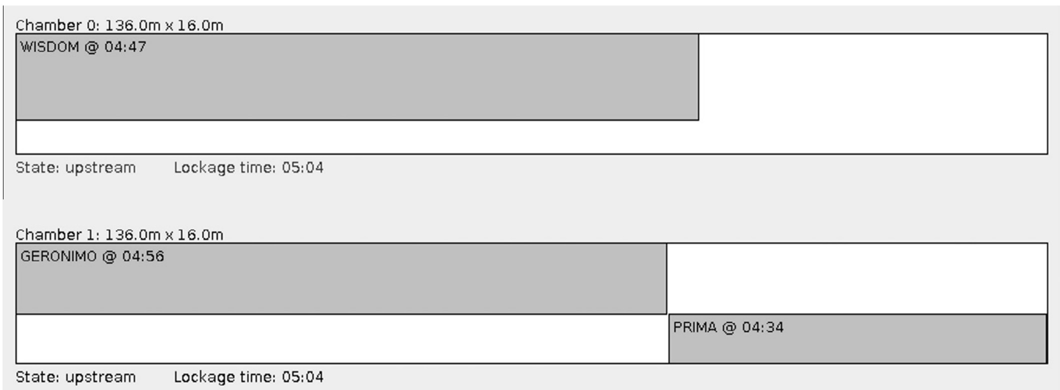

Fig. 6. An example of a feasible solution for which the original index ordering constraint (60) does not hold.

4.2. Inland locks

The inland test set is based on data from the waterway traffic on the Albertkanaal in Belgium. The locks on this canal have two identical small chambers and one large chamber, all of which can be operated independently. Table 1 presents the properties of these locks. The traffic at the locks was generated based on actual traffic from 2008 and the settings from Table 2 and is available online (Verstichel, 2012). The instance properties can be identified with the following convention: 'mean inter arrival time'-'number of ships'-'fraction of ships traveling upstream'. The ship sizes were extracted from the traffic from 2008, while the inter arrival times were generated randomly between zero and two times the given mean inter arrival time. A first-come-first-served policy is used with respect to the ship arrival times to maintain fairness among the ships. This first-come-first-served policy also corresponds to the actual practice at the locks. Therefore, the speedups described in Section 4.1 were added to the model. We assume that all safety distances are included in the ship and chamber dimensions and that the number of ships in a lockage does not influence the processing time.

First, we compare the performance of the model in four different test settings. In the 'single small' setting only one chamber of type 1 is available at the lock. In the 'single large' setting, one chamber of type 2 is available at the lock. The 'parallel small' setting assumes a lock with two parallel chambers of type 1, while the 'real' setting considers the actual lock from the Albertkanaal (Table 1). These settings allow for a thorough comparison of the computation times and waiting times for different lock settings. The difference between using a single small and a single large chamber also gives an indication of the influence of the ship placement part on the calculation time. Table 3 shows an overview of the results for these four different inland test settings. The results were obtained under a weighted completion time objective ($K_{wt} = 1.0$), with the minimization of the maximum waiting time as a secondary objective ($K_T = 0.1$). The total number of lockages was not taken into account for these experiments ($K_z = 0.0$). All single chamber instances were solved to optimality in less than 20 s, while the large chamber instances took up to 340 s to be solved to optimality. This suggests that the ship placement part of the lock scheduling problem strongly influences the hardness of an instance. While the instances with parallel chambers took significantly more time for solving compared to the single small chamber cases, a reduction of the waiting time of at least 66.8% was obtained, with a maximum of 100%. When comparing the waiting times for a single large chamber and the parallel small chambers, the difference over all instances is not statistically significant ($p - value = 0.1$), but it is highly dependent on the average inter arrival time. The single large chamber is on average 14% better in case of a small inter arrival time (5 min). When the inter arrival time increases to 10, 15 and 30 min, the lock with parallel chambers obtains an average reduction of the waiting times of 15%, 73% and 90% respectively. When looking at the 'real' setting, the optimal solutions are found slower compared to using a single chamber type, but the resulting reduction in waiting time is still significant. Careful examination of the search logs provided by Gurobi did however show that the optimal solution was found early in the search in almost all cases, and that the remaining computation time was used to prove optimality.

A second set of experiments was performed on the smallest test instances for inland locks. We removed the first-come-first-served policy from the model for the 'single small', 'parallel small' and 'real' locks. This means that the lockage index constraints (60)–(62) could not be added for gaining a speedup in calculation time. The results for these experiments are presented in Table 4, from which it is clear that the FCFS constraints strongly reduce the calculation time. For the single chamber setting, the model with FCFS constraints is between 43 and 293 times faster, while the waiting time increases between 0% en 27.4%. For the parallel chambers, the FCFS model is between 1.5 and 159 times faster, while the total waiting time is increased for just one instance. When looking at the real-life lock, both approaches are the fastest on 50% of the instances. In the first two lock configurations, the number of explored nodes was much larger for these experiments, compared to the ones the first-come-first-served and lockage index constraints. For the 'single small' setting the average number of explored nodes is 903 and 5 respectively, while the 'parallel small' setting results in an average 15,394 and 3071 explored nodes. When looking at the 'real' setting, we find that the average number of explored nodes is 45,177 and 3096 respectively.

Table 1
Attributes of the locks on the Albertkanaal.

	Width (m)	Length (m)	p (min)	Chamber ID
Chamber type 1	16.0	136.0	16	1, 2
Chamber type 2	24.0	200.0	16	3

Table 2
Properties of the instances generated for the Albertkanaal.

Traffic properties	Instances
Mean inter arrival time (min)	5, 10, 15, 30
Number of ships	10, 20, 30
Upstream/downstream traffic	50/50, 30/70
Safety distances	Included in ship dimensions
Example instance name	5-20-0.3

Table 3
Results for the traffic on the Albertkanaal. '#' Shows the number of lockages, 'WT' the total waiting time in minutes and 'Calc' the calculation time in seconds.

Instance	SingleSmall			SingleLarge			ParallelSmall			Real		
	#	WT	Calc	#	WT	Calc	#	WT	Calc	#	WT	Calc
5-10-0.3	6	218	0.48	4	110	2.75	6	66	2.33	6	21	41.73
5-10-0.5	7	292	0.11	4	74	2.73	7	52	2.28	8	5	28.02
10-10-0.3	10	467	0.09	6	107	0.90	9	115	1.14	9	11	33.63
10-10-0.5	8	252	0.08	5	103	1.49	9	54	1.26	9	15	30.64
15-10-0.3	9	148	0.09	7	64	1.65	10	2	0.28	10	0	25.69
15-10-0.5	7	182	0.13	7	81	2.41	9	17	0.33	10	0	25.63
30-10-0.3	8	15	0.2	8	15	1.71	10	0	0.11	10	0	24.45
30-10-0.5	10	48	0.64	10	48	1.71	10	0	0.11	10	0	26.34
5-20-0.3	15	1829	1.28	6	297	251.51	16	606	353.91	14	59	$\sim 3.9h$
5-20-0.5	13	1350	0.83	7	211	10.32	13	379	$\sim 2h$	13	37	$\sim 5h$
10-20-0.3	16	1352	1.08	8	203	11.16	16	241	177.03	18	17	1305.73
10-20-0.5	16	1025	0.59	8	163	9.34	17	130	478.62	17	16	3207.56
15-20-0.3	14	600	19.58	10	148	10.00	17	77	215.21	18	9	$\sim 6.2h$
15-20-0.5	16	863	1.27	13	171	18.65	20	7	4.42	20	0	29.72
30-20-0.3	19	679	1.19	12	117	10.12	19	33	5.05	20	2	$> 12h$
30-20-0.5	17	182	1.17	15	123	10.43	19	14	1.7	20	0	30.12
5-30-0.3	18	2034	6.64	8	350	157.74	19	422	$> 12h$	19	75	$> 12h$
5-30-0.5	15	1017	6.31	9	340	45.36	17	183	$> 12h$	20	52	$> 12h$
10-30-0.3	23	2407	11.78	13	188	56.65	26	136	$\sim 2.8h$	24	15	$> 12h$
10-30-0.5	22	1308	3.7	16	290	336.32	23	227	$> 12h$	28	13	$> 12h$
15-30-0.3	23	1410	4.36	17	218	123.51	26	111	$> 12h$	28	3	$> 12h$
15-30-0.5	24	795	4.59	18	197	135.54	26	57	$\sim 1h$	27	10	$> 12h$
30-30-0.3	26	119	6.11	26	98	121.34	29	12	89.39	29	2	454.58
30-30-0.5	27	114	3.66	26	82	48.89	29	7	40.83	28	1	83.47

Table 4
Results for the experiments without first-come-first-served policy on the Albertkanaal instances. '#' Shows the number of lockages, 'WT' the total waiting time in minutes and 'Calc' the calculation time in seconds.

Instance	SmallSingle			SmallParallel			Real		
	#	WT	Calc	#	WT	Calc	#	WT	Calc
5-10-0.3	6	218	23.39	6	66	34.34	5	21	65.61
5-10-0.5	7	258	27.23	7	52	4.44	8	5	30.19
10-10-0.3	9	339	27.16	9	81	23.33	9	11	1681.97
10-10-0.5	8	220	21.94	9	54	26.41	9	15	45.36
15-10-0.3	9	136	27.58	10	2	0.47	10	0	19.25
15-10-0.5	7	150	27.45	9	17	19.08	10	0	23.52
30-10-0.3	8	15	27.50	10	0	17.42	10	0	0.72
30-10-0.5	10	48	28.02	10	0	17.28	10	0	0.70

The efficiency of the lockage index constraint decreases with each relaxation of the constraint. Where the original constraint reduced the number of explored nodes by a factor of 180, this drops to a factor of 5 after the first relaxation. However, in both cases the influence on the calculation time is considerable. The second relaxation of the constraint reduced the number of explored nodes by a factor of 15, while the influence on the calculation time is less pronounced as in the other two settings.

4.3. Port locks

The test set for locks in ports is based on historical data from the Port of Antwerp. This port has four different lock complexes that connect the docks to the river Scheldt. Our test instances are based on historical data for the Berendrecht–Zandvliet (BE–ZV) lock complex, and the Van Cauwelaert (VC) lock. These instances contain between 10 and 28 ships, and were solved with the same relative costs as for the inland waterways experiments: $K_{wt} = 1.0,\ K_z = 0.0,\ K_T = 0.1$. The Van Cauwelaert lock only transferred barges, while the Berendrecht–Zandvliet complex handled a mix of sea vessels and barges. The instances were provided to the authors under a non-disclosure agreement and can therefore not be published online. The properties of the locks are provided in Table 5.

The results, both with and without the first-come-first-served constraints, are presented in Table 6. These results show that the ship dependent safety distances, setup times and processing times significantly increase the calculation time, even for the smallest instances, compared to the inland case. Adding a first-come-first-served policy once again strongly reduces the calculation time and has a limited impact on the weighted completion time objective. For one instance, VC-28, the first-come-first-served solution is even better than the normal solution, which is due to the fact that the objective was still

Table 5
Attributes of the selected locks from the Port of Antwerp.

	Width (m)	Length (m)	p (min)	Chamber ID
Berendrecht	68.0	500.0	30	BE
Zandvliet	57.0	500.0	30	ZV
Van Cauwelaert	35.0	270.0	30	VC

Table 6
Results for the experiments for the port instances. 'Calc' shows the calculation time in seconds, 'Cost' the objective cost of the solution. Best results are indicated in bold.

Lock	#Ships	No FCFS		FCFS	
		Cost	Calc	Cost	Calc
BE-ZV	10	**3076.5**	4917.96	3086.8	23.15
BE-ZV	12	**2493.7**	$> 12h$	2623.5	72.28
BE-ZV	13	**3611.8**	$> 12h$	3615.3	129.78
VC	10	**2920.2**	18.50	**2920.2**	17.40
VC	12	**5683.3**	20.37	**5683.3**	17.83
VC	23	**16522.3**	3906.45	**16522.3**	76.00
VC	28	32634.6	$> 12h$	**32589.5**	2628.35

Table 7
Results for the comparison between the heuristic scheduler Verstichel et al. (2011a) and the exact approach. 'Cost' represents the objective cost of the solution. Best results are indicated in bold.

	Heuristic Cost	Optimal Cost
R-5-20-0.3	80,221	**50,409**
R-5-20-0.5	50,392	**50,300**
R-10-20-0.3	70,344	**50,405**
R-10-20-0.5	60,473	**40,551**
R-15-20-0.3	70,477	**40,707**
R-15-20-0.5	80,808	**50,837**
R-30-20-0.3	100,773	**51,850**
R-30-20-0.5	61,035	**51,213**

improving after the 12 h time-out on the calculations. Due to the limited influence of the first-come-first-served policy on the results and its large decrease in calculation time, it might be worthwhile to consider applying these constraints. Even when applied to only a subset of the ships, the resulting decrease in computation time may be considerably large.

4.4. Comparison with other results

In (Verstichel et al., 2011a) a heuristic method was proposed for the lock scheduling problem in an inland setting. In this section we compare the results of that heuristic with the ones obtained by our model. The cost function in (Verstichel et al., 2011a) considers $K_z = 10,000$, $K_{wt} = 1.0$ and $K_T = 0.0$, a weight $wt\boldsymbol{c_i} = 10$ for all ships with increased priority and $wt\boldsymbol{c_i} = 1$ for all other ships. No first-come-first-served policy is used, and all safety distances are assumed to be included in the ship dimensions. We compare the solution quality for the eight instances with 20 ships each. Table 7 show the objective cost of the solutions found by both approaches. The average improvement gained by the mixed integer linear programming model compared to the heuristic is 30.3%. It should however be noted that the heuristic method generated all the solutions in less than 10 s, while the exact approach required more than 12 h to reach optimality on all but one instance. These results show that there is still a lot of room for improving the heuristics for the lock scheduling problem.

5. Conclusion

We have described the generalized lock scheduling problem and presented a mathematical model. The work builds on previous models for the ship placement part and a simplified version of the scheduling part. Up till now no combined model was available, and hence instances of the lock scheduling problem were never solved to optimality. The model represents lock scheduling for both inland and port locks, under a wide range of real-life constraints, and using a weighted objective function. The model is very flexible, as its modular structure allows for a very easy addition and removal of constraints and objective terms, depending on the situation at hand. This includes specific settings for locks in ports or on inland waterways, but also environmental problems such as drought and operational issues like large queues. Experiments on data instances generated from historical data show that instances with up to 30 ships can be solved to optimality in less than 12 h, even for a real-life lock with three chambers and two different chamber types. The model is further improved by adding constraints that reduce calculation time and tree size by several orders of magnitude when a first-come-first served policy is applicable. These constraints are introduced for single chamber locks and relaxed for usage with identical parallel chambers and locks with several chamber types.

Acknowledgements

Research funded by a Ph.D. grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). This work is supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office.

We would like to thank the Scheepvaartmanagement of the Port of Antwerp for sharing their experience and historical data of the lock scheduling problem in ports.

We would also like to thank nv De Scheepvaart and IT-Bizz for providing real-life data for inland locks.

Appendix A. Full MILP model

Ship parameters:

N, N':	Set of upstream, downstream ships, indexed by $i,j \in \{1,\ldots,n^{(\prime)}\}$.
M, M':	Set of the upstream, downstream lockages, indexed by k, l.
w_i, l_i, d_i:	Width, length and draught of upstream ship i.
w'_i, l'_i, d'_i:	Width, length and draught of downstream ship i.
dF_i, dB_i:	Minimal distance between upstream ship i and the front, back of the chamber.
dF'_i, dB'_i:	Minimal distance between downstream ship i and the front, back of the chamber.
sW_{ij}, sL_{ij}:	Minimal safety distance between upstream ships i and j when they are lying next to, or behind each other.
sW'_{ij}, sL'_{ij}:	Minimal safety distance between downstream ships i and j when they are lying next to, or behind each other.
$MOOR_i, MOOR'_i$:	Set of ships to which upstream, downstream ship i can moor.
$pre_i, post_i$:	Pre- and post-processing times of upstream ship i.
$pre'_i, post'_i$:	Pre- and post-processing times of downstream ship i.
p_i, p'_i:	Additional processing time induced by upstream, downstream ship i.
r_i, r'_i:	Arrival time of upstream, downstream ship i at the coordination point.
$c_{min,i}, c_{max,i}$:	Lower and upper limit on the completion time of upstream ship i.
$c'_{min,i}, c'_{max,i}$:	Lower and upper limit on the completion time of downstream ship i.
wct_i, wct'_i:	Weight of the completion time of upstream, downstream ship i.
$TIDAL, TIDAL'$:	Set of all upstream, downstream ships with a tidal window.
$FCFS, FCFS'$:	Set of all upstream, downstream ships that must be processed first-come-first-served with respect to their arrival times.

Lock parameters:

$TYPES$:	Set of available chamber types at the lock, indexed by t.
M_t, M'_t:	Subset of M, reserved for upstream, downstream lockages performed on the chamber (s) of type t.
U_t:	Set of physical chambers of type t, indexed by u.
W_t, L_t, D_t:	Width, length and draught of the chamber (s) of type t.
W, T, D:	Maximal width, length and draught over all chambers.
$ship_0, ship'_0$:	The upstream, downstream left quay of each chamber.
$ship_{n+t}, ship'_{n'+t}$:	The upstream, downstream right quay of each chamber of type t.
p:	Minimal duration of a lockage.
$setup_{kl}$:	Minimal setup time between lockages k and l when they are processed by the same chamber of type t. Depends on the chamber type t and the direction of k and l.
C_{max}:	Upper bound for the completion time (Big_M constant, is reduced using heuristics).
K_1, K_2, K_3:	Weights of the number of lockage operations, weighted completion time and maximum tardiness.

$$minimize\, K_1 \sum_{k \in M \cup M'} \boldsymbol{z_k} + K_2\Big(\sum_{i \in N} wct_i \boldsymbol{c_i} + \sum_{i \in N'} wct'_i \boldsymbol{c'_i}\Big) + K_3 \boldsymbol{T}_{max} \tag{A.1}$$

s.t.

The first block of constraints (A.2)–(A.30) and (A.31) models the ship placement part of the upstream ships.

$$\boldsymbol{left_{ij}} + \boldsymbol{left_{ji}} + \boldsymbol{b_{ij}} + \boldsymbol{b_{ji}} + (1 - \boldsymbol{f}_{\boldsymbol{ik}}) + (1 - \boldsymbol{f}_{\boldsymbol{jk}}) \geqslant 1 \quad \forall i < j,\ i,j \in N,\ k \in M \tag{A.2}$$

$$\boldsymbol{x_i} - \boldsymbol{x_j} + W\boldsymbol{left_{ij}} \leqslant W - w_i, \quad \forall i,j \in N \tag{A.3}$$

$$\boldsymbol{y_i} - \boldsymbol{y_j} + L\boldsymbol{b_{ij}} \leqslant L - l_i, \quad \forall i,j \in N \tag{A.4}$$

$$\boldsymbol{x_j} - \boldsymbol{x_i} + (W + sW_{ij})(1 - \boldsymbol{left_{ij}} + \boldsymbol{b_{ij}}) \geqslant w_i + sW_{ij}, \quad \forall i,j \in N \tag{A.5}$$

$$\boldsymbol{y_j} - \boldsymbol{y_i} + (L + sL_{ij})(1 - \boldsymbol{b_{ij}} + \boldsymbol{left_{ij}}) \geqslant l_i + sL_{ij}, \quad \forall i,j \in N \tag{A.6}$$

$$\boldsymbol{x_i} + w_i \leqslant W_t + (1 - \boldsymbol{f}_{\boldsymbol{ik}})W, \quad \forall i \in N,\ t \in TYPES,\ k \in M_t \tag{A.7}$$

Variables:

$\boldsymbol{x_i}, \boldsymbol{y_i}$:	Integer variables that define the x and y position of upstream ship i (front left corner).
$\boldsymbol{x'_i}, \boldsymbol{y'_i}$:	Integer variables that define the x and y position of downstream ship i (front left corner).
$\boldsymbol{left_{ij}}, \boldsymbol{left'_{ij}}$:	Binary variable indicating whether ship i is to the left of ship j (1) or not (0). (upstream, downstream)
$\boldsymbol{b_{ij}}, \boldsymbol{b'_{ij}}$:	Binary variable indicating whether ship i is behind ship j (1) or not (0). (upstream, downstream)
$\boldsymbol{ml_{ij}}, \boldsymbol{mr_{ij}}$:	Binary variables, 1 when ship i is moored to ship j's left, respectively right, 0 otherwise. (upstream)
$\boldsymbol{ml'_{ij}}, \boldsymbol{mr'_{ij}}$:	Binary variables, 1 when ship i is moored to ship j's left, respectively right, 0 otherwise. (downstream)
$\boldsymbol{z_k}$:	Binary variable that indicates whether lockage k is used (1) or not (0).
$\boldsymbol{f}_{\boldsymbol{ik}}, \boldsymbol{f}'_{\boldsymbol{ik}}$:	Binary variable that indicates whether upstream, downstream ship i is processed in lockage k (1) or not (0).
$\boldsymbol{v_{ij}}, \boldsymbol{v'_{ij}}$:	Binary variable, 1 when ship i and j are processed in the same lockage, 0 otherwise. (upstream, downstream)
$\boldsymbol{c_i}, \boldsymbol{c'_i}$:	Departure time of upstream, downstream ship i (completion time of the lockage).
$\boldsymbol{C_k}$	Completion time of lockage k.
$\boldsymbol{P_k}$	Processing time of lockage k.
$\boldsymbol{s_{kl}}$:	Ship dependent setup time for processing lockage l immediately after lockage k in the same chamber.
$\boldsymbol{seq_{kl}}$:	Binary variable, 1 when lockage k precedes lockage l in the same chamber, 0 otherwise.
$\boldsymbol{proc_{ku}}$:	Binary variable that indicates whether lockage k is processed in chamber u (1) or not (0).
$\boldsymbol{T}_{max}$:	Maximum lock transit time over all ships.

$$\boldsymbol{y_i} + l_i \leqslant L_t + (1 - \boldsymbol{f}_{\boldsymbol{ik}})L, \quad \forall i \in N,\ t \in TYPES,\ k \in M_t \tag{A.8}$$

$$d_i \leqslant D_t + (1 - \boldsymbol{f}_{\boldsymbol{ik}})D, \quad \forall i \in N,\ t \in TYPES,\ k \in M_t \tag{A.9}$$

$$\boldsymbol{y_i} \geqslant dF_i, \quad \forall i \in N \tag{A.10}$$

$$\boldsymbol{y_i} + l_i \leqslant L_t - dB_i + (1 - \boldsymbol{f}_{\boldsymbol{ik}})L, \quad \forall i \in N,\ t \in TYPES,\ k \in M_t \tag{A.11}$$

$$\sum\nolimits_{k \in M} \boldsymbol{f}_{\boldsymbol{ik}} = 1, \quad \forall i \in N \tag{A.12}$$

$$\boldsymbol{f}_{\boldsymbol{ik}} \leqslant \boldsymbol{z_k}, \quad \forall i \in N,\ k \in M \tag{A.13}$$

$$\boldsymbol{y_j} - \boldsymbol{y_i} \leqslant (1 - \boldsymbol{ml_{ij}})L, \quad \forall i \in N,\ j \in MOOR_i \tag{A.14}$$

$$\boldsymbol{y_i} - \boldsymbol{y_j} \leqslant l_j - l_i + (1 - \boldsymbol{ml_{ij}})L, \quad \forall i \in N,\ j \in MOOR_i \tag{A.15}$$

$$\boldsymbol{x_j} - \boldsymbol{x_i} \leqslant w_i + (1 - \boldsymbol{ml_{ij}})W, \quad \forall i \in N,\ j \in MOOR_i \tag{A.16}$$

$$\boldsymbol{x_j} - \boldsymbol{x_i} \geqslant w_i - (1 - \boldsymbol{ml_{ij}})W, \quad \forall i \in N,\ j \in MOOR_i \tag{A.17}$$

$$\boldsymbol{x_{n+t}} - \boldsymbol{x_i} \leqslant w_i + (1 - \boldsymbol{ml_{i,n+t}})W, \quad \forall i \in N,\ t \in TYPES \tag{A.18}$$

$$\boldsymbol{x_{n+t}} - \boldsymbol{x_i} \geqslant w_i - (1 - \boldsymbol{ml_{i,n+t}})W, \quad \forall i \in N,\ t \in TYPES \tag{A.19}$$

$$\boldsymbol{ml_{i,n+t}} \leqslant \sum_{k \in M_t} \boldsymbol{f}_{\boldsymbol{ik}}, \quad \forall i \in N,\ t \in TYPES \tag{A.20}$$

$$\boldsymbol{y_j} - \boldsymbol{y_i} \leqslant (1 - \boldsymbol{mr_{ij}})L, \quad \forall i \in N,\ j \in MOOR_i \tag{A.21}$$

$$\boldsymbol{y_i} - \boldsymbol{y_j} \leqslant l_j - l_i + (1 - \boldsymbol{mr_{ij}})L, \quad \forall i \in N,\ j \in MOOR_i \tag{A.22}$$

$$\boldsymbol{x_j} - \boldsymbol{x_i} \leqslant -w_j + (1 - \boldsymbol{mr_{ij}})W, \quad \forall i \in N,\ j \in MOOR_i \tag{A.23}$$

$$\boldsymbol{x_j} - \boldsymbol{x_i} \geqslant -w_j - (1 - \boldsymbol{mr_{ij}})W, \quad \forall i \in N,\ j \in MOOR_i \tag{A.24}$$

$$\boldsymbol{x_0} - \boldsymbol{x_i} \leqslant w_i + (1 - \boldsymbol{ml_{i,0}})W, \quad \forall i \in N \tag{A.25}$$

$$\boldsymbol{x_0} - \boldsymbol{x_i} \geqslant w_i - (1 - \boldsymbol{ml_{i,0}})W, \quad \forall i \in N \tag{A.26}$$

$$\sum\nolimits_{j \in N: j \neq i} (\boldsymbol{ml_{ij}} + \boldsymbol{mr_{ij}}) + \boldsymbol{mr_{i,0}} + \sum_{t \in TYPES} \boldsymbol{ml_{i,n+t}} \geqslant 1, \quad \forall i \in N \tag{A.27}$$

$$\boldsymbol{ml_{ij}} + \boldsymbol{mr_{ji}} \leqslant 1, \quad \forall i \neq j,\ i,j \in N \tag{A.28}$$

$$\boldsymbol{f}_{\boldsymbol{ik}} - \boldsymbol{f}_{\boldsymbol{jk}} \leqslant (1 - \boldsymbol{v_{ij}}), \quad \forall i < j,\ i,j \in N,\ k \in M \tag{A.29}$$

$$\boldsymbol{f}_{\boldsymbol{jk}} - \boldsymbol{f}_{\boldsymbol{ik}} \leqslant (1 - \boldsymbol{v_{ij}}), \quad \forall i < j,\ i,j \in N,\ k \in M \tag{A.30}$$

$$\boldsymbol{ml_{ij}} + \boldsymbol{mr_{ij}} + \boldsymbol{ml_{ji}} + \boldsymbol{mr_{ji}} \leqslant \boldsymbol{v_{ij}}, \quad \forall i < j,\ i,j \in N \tag{A.31}$$

The second block of constraints (A.32)–(A.60) and (A.61) models the ship placement part of the downstream ships.

$$\boldsymbol{left'_{ij}} + \boldsymbol{left'_{ji}} + \boldsymbol{b'_{ij}} + \boldsymbol{b'_{ji}} + (1 - \boldsymbol{f}'_{\boldsymbol{ik}}) + (1 - \boldsymbol{f}'_{\boldsymbol{jk}}) \geqslant 1 \quad \forall i < j,\ i,j \in N',\ k \in M' \tag{A.32}$$

$$\boldsymbol{x'_i} - \boldsymbol{x'_j} + W\boldsymbol{left'_{ij}} \leqslant W - w'_i, \quad \forall i,j \in N' \tag{A.33}$$

$$\boldsymbol{y'_i} - \boldsymbol{y'_j} + L\boldsymbol{b'_{ij}} \leqslant L - l'_i, \quad \forall i,j \in N' \tag{A.34}$$

$$\boldsymbol{x'_j} - \boldsymbol{x'_i} + (W + sW'_{ij})(1 - \boldsymbol{left'_{ij}} + \boldsymbol{b'_{ij}}) \geqslant w'_i + sW'_{ij}, \quad \forall i,j \in N' \tag{A.35}$$

$$\boldsymbol{y'_j} - \boldsymbol{y'_i} + (L + sL'_{ij})(1 - \boldsymbol{b'_{ij}} + \boldsymbol{left'_{ij}}) \geqslant l'_i + sL'_{ij}, \quad \forall i,j \in N' \tag{A.36}$$

$$\boldsymbol{x'_i} + w'_i \leqslant W_t + (1 - \boldsymbol{f}'_{\boldsymbol{ik}})W, \quad \forall i \in N',\ t \in TYPES,\ k \in M'_t \tag{A.37}$$

$$\boldsymbol{y'_i} + l'_i \leqslant L_t + (1 - \boldsymbol{f}'_{\boldsymbol{ik}})L, \quad \forall i \in N',\ t \in TYPES,\ k \in M'_t \tag{A.38}$$

$$d'_i \leqslant D_t + (1 - \boldsymbol{f}'_{\boldsymbol{ik}})D, \quad \forall i \in N',\ t \in TYPES,\ k \in M'_t \tag{A.39}$$

$$\boldsymbol{y'_i} \geqslant dF'_i, \quad \forall i \in N' \tag{A.40}$$

$$\boldsymbol{y'_i} + l'_i \leqslant L'_t - dB'_i + (1 - \boldsymbol{f}'_{\boldsymbol{ik}})L, \quad \forall i \in N',\ t \in TYPES,\ k \in M'_t \tag{A.41}$$

$$\sum_{k \in M} \boldsymbol{f}'_{\boldsymbol{ik}} = 1, \quad \forall i \in N' \tag{A.42}$$

$$\boldsymbol{f}'_{\boldsymbol{ik}} \leqslant \boldsymbol{z'_k}, \quad \forall i \in N',\ k \in M' \tag{A.43}$$

$$\boldsymbol{y'_j} - \boldsymbol{y'_i} \leqslant (1 - \boldsymbol{ml'_{ij}})L, \quad \forall i \in N',\ j \in MOOR'_i \tag{A.44}$$

$$\boldsymbol{y'_i} - \boldsymbol{y'_j} \leqslant l'_j - l'_i + (1 - \boldsymbol{ml'_{ij}})L, \quad \forall i \in N',\ j \in MOOR'_i \tag{A.45}$$

$$\boldsymbol{x'_j} - \boldsymbol{x'_i} \leqslant w'_i + (1 - \boldsymbol{ml'_{ij}})W, \quad \forall i \in N',\ j \in MOOR'_i \tag{A.46}$$

$$\boldsymbol{x'_j} - \boldsymbol{x'_i} \geqslant w'_i - (1 - \boldsymbol{ml'_{ij}})W, \quad \forall i \in N',\ j \in MOOR'_i \tag{A.47}$$

$$\boldsymbol{x'_{n'+t}} - \boldsymbol{x'_i} \leqslant w'_i + (1 - \boldsymbol{ml'_{i,n'+t}})W, \quad \forall i \in N',\ t \in TYPES \tag{A.48}$$

$$\boldsymbol{x'_{n'+t}} - \boldsymbol{x'_i} \geqslant w'_i - (1 - \boldsymbol{ml'_{i,n'+t}})W, \quad \forall i \in N',\ t \in TYPES \tag{A.49}$$

$$\boldsymbol{ml'_{i,n'+t}} \leqslant \sum_{k \in M'_t} \boldsymbol{f}'_{\boldsymbol{ik}}, \quad \forall i \in N',\ t \in TYPES \tag{A.50}$$

$$\boldsymbol{y'_j} - \boldsymbol{y'_i} \leqslant (1 - \boldsymbol{mr'_{ij}})L, \quad \forall i \in N',\ j \in MOOR'_i \tag{A.51}$$

$$\boldsymbol{y'_i} - \boldsymbol{y'_j} \leqslant l'_j - l'_i + (1 - \boldsymbol{mr'_{ij}})L, \quad \forall i \in N',\ j \in MOOR'_i \tag{A.52}$$

$$\boldsymbol{x'_j} - \boldsymbol{x'_i} \leqslant -w'_j + (1 - \boldsymbol{mr'_{ij}})W, \quad \forall i \in N',\ j \in MOOR'_i \tag{A.53}$$

$$\boldsymbol{x'_j} - \boldsymbol{x'_i} \geqslant -w'_j - (1 - \boldsymbol{mr'_{ij}})W, \quad \forall i \in N',\ j \in MOOR'_i \tag{A.54}$$

$$\boldsymbol{x'_0} - \boldsymbol{x'_i} \leqslant w'_i + (1 - \boldsymbol{ml'_{i,0}})W, \quad \forall i \in N' \tag{A.55}$$

$$\boldsymbol{x'_0} - \boldsymbol{x'_i} \geqslant w'_i - (1 - \boldsymbol{ml'_{i,0}})W, \quad \forall i \in N' \tag{A.56}$$

$$\sum_{j \in N' : j \neq i} (\boldsymbol{ml'_{ij}} + \boldsymbol{mr'_{ij}}) + \boldsymbol{mr'_{i,0}} + \sum_{t \in TYPES} \boldsymbol{ml'_{i,n'+t}} \geqslant 1, \quad \forall i \in N' \tag{A.57}$$

$$\boldsymbol{ml'_{ij}} + \boldsymbol{mr'_{ji}} \leqslant 1, \quad \forall i \neq j,\ i,j \in N' \tag{A.58}$$

$$\boldsymbol{f}'_{\boldsymbol{ik}} - \boldsymbol{f}'_{\boldsymbol{jk}} \leqslant (1 - \boldsymbol{v'_{ij}}), \quad \forall i < j,\ i,j \in N',\ k \in M' \tag{A.59}$$

$$\boldsymbol{f}'_{\boldsymbol{jk}} - \boldsymbol{f}'_{\boldsymbol{ik}} \leqslant (1 - \boldsymbol{v'_{ij}}), \quad \forall i < j,\ i,j \in N',\ k \in M' \tag{A.60}$$

$$\boldsymbol{ml'_{ij}} + \boldsymbol{mr'_{ij}} + \boldsymbol{ml'_{ji}} + \boldsymbol{mr'_{ji}} \leqslant \boldsymbol{v'_{ij}}, \quad \forall i < j,\ i,j \in N' \tag{A.61}$$

The third block of constraints (A.62)–(A.81) and (A.82) models the scheduling part of the lock scheduling problem.

$$\boldsymbol{c_i} \geqslant C_{max}(\boldsymbol{f_{ik}} - 1) + \boldsymbol{C_k}, \quad \forall i \in N,\ k \in M \tag{A.62}$$

$$\boldsymbol{c_i} \leqslant C_{max}(1 - \boldsymbol{f_{ik}}) + \boldsymbol{C_k}, \quad \forall i \in N,\ k \in M \tag{A.63}$$

$$\boldsymbol{c'_i} \geqslant C_{max}(\boldsymbol{f}'_{\boldsymbol{ik}} - 1) + \boldsymbol{C_k}, \quad \forall i \in N',\ k \in M' \tag{A.64}$$

$$\boldsymbol{c'_i} \leqslant C_{max}(1 - \boldsymbol{f}'_{\boldsymbol{ik}}) + \boldsymbol{C_k}, \quad \forall i \in N',\ k \in M' \tag{A.65}$$

$$c_{min,i} \leqslant \boldsymbol{c_i} \leqslant c_{max,i}, \quad \forall i \in TIDAL \tag{A.66}$$

$$c'_{min,i} \leqslant \boldsymbol{c'_i} \leqslant c'_{max,i}, \quad \forall i \in TIDAL' \tag{A.67}$$

$$\boldsymbol{P_k} \geqslant p_t \boldsymbol{z_k} + \sum_{i \in N} \boldsymbol{f_{ik}} p_i, \quad \forall k \in M_t,\ t \in TYPES \tag{A.68}$$

$$\boldsymbol{P_k} \geqslant p'_t \boldsymbol{z_k} + \sum_{i \in N'} \boldsymbol{f}'_{\boldsymbol{ik}} p'_i, \quad \forall k \in M'_t,\ t \in TYPES \tag{A.69}$$

$$\boldsymbol{s_{kl}} \geqslant setup_{kl} + \sum_{i \in N} \boldsymbol{f_{ik}} post_i + \sum_{i \in N} \boldsymbol{f_{il}} pre_i \tag{A.70}$$

$$\forall l \neq k,\ k,l \in M_t \cup M'_t,\ t \in TYPES$$

$$\textstyle\sum_{u \in U_t} \boldsymbol{proc_{ku}} = \boldsymbol{z_k}, \quad \forall k \in M_t \cup M'_t,\ t \in TYPES \tag{A.71}$$

$$\boldsymbol{proc_{ku}} + \sum_{v \in U_t, v \neq u} \boldsymbol{proc_{lv}} + \boldsymbol{seq_{kl}} \leqslant 2, \tag{A.72}$$

$$\forall l > k,\ k,l \in M_t \cup M'_t,\ u \in U_t,\ t \in TYPES$$

$$\boldsymbol{C_l} - \boldsymbol{C_k} + 2C_{max}(3 - \boldsymbol{seq_{kl}} - \boldsymbol{proc_{ku}} - \boldsymbol{proc_{lu}}) \geqslant \boldsymbol{P_l} + \boldsymbol{s_{kl}} \tag{A.73}$$

$$\forall k < l,\ k,l \in M_t \cup M'_t,\ u \in U_t,\ t \in TYPES$$

$$\boldsymbol{C_k} - \boldsymbol{C_l} + 2C_{max}(2 + \boldsymbol{seq_{kl}} - \boldsymbol{proc_{ku}} - \boldsymbol{proc_{lu}}) \geqslant \boldsymbol{P_k} + \boldsymbol{s_{lk}} \tag{A.74}$$

$$\forall k < l,\ k,l \in M_t \cup M'_t,\ u \in U_t,\ t \in TYPES$$

$$\boldsymbol{C_k} - \boldsymbol{P_k} \geqslant \boldsymbol{f}_{\boldsymbol{ik}} r_i, \quad \forall i \in N,\ k \in M \tag{A.75}$$

$$\boldsymbol{C_k} - \boldsymbol{P_k} \geqslant \boldsymbol{f}'_{\boldsymbol{ik}} r'_i, \quad \forall i \in N',\ k \in M' \tag{A.76}$$

$$\boldsymbol{z_k} \leqslant \sum_{i \in N} \boldsymbol{f}_{\boldsymbol{ik}}, \quad \forall k \in M \tag{A.77}$$

$$\boldsymbol{z_k} \leqslant \sum_{i \in N'} \boldsymbol{f}'_{\boldsymbol{ik}}, \quad \forall k \in M' \tag{A.78}$$

$$\boldsymbol{T}_{max} \geqslant \boldsymbol{c_i} - r_i, \quad \forall i \in N \tag{A.79}$$

$$\boldsymbol{T}_{max} \geqslant \boldsymbol{c'_i} - r'_i, \quad \forall i \in N' \tag{A.80}$$

$$\boldsymbol{c_i} \leqslant \boldsymbol{c_j}, \quad \forall i < j,\ i,j \in FCFS \tag{A.81}$$

$$\boldsymbol{c'_i} \leqslant \boldsymbol{c'_j}, \quad \forall i < j,\ i,j \in FCFS' \tag{A.82}$$

Constraints (A.83)–(A.100) and (A.101) formulate bounds and integrality constraints on the variables.

$$\boldsymbol{left_{ij}}, \boldsymbol{b_{ij}}, \boldsymbol{ml_{ij}}, \boldsymbol{mr_{ij}} \in \{0,1\}, \quad \forall i,j \in N \tag{A.83}$$

$$\boldsymbol{left'_{ij}}, \boldsymbol{b'_{ij}}, \boldsymbol{ml'_{ij}}, \boldsymbol{mr'_{ij}} \in \{0,1\}, \quad \forall i,j \in N' \tag{A.84}$$

$$\boldsymbol{v_{ij}} \in \{0,1\}, \quad \forall i < j,\ i,j \in N \tag{A.85}$$

$$\boldsymbol{v'_{ij}} \in \{0,1\}, \quad \forall i < j,\ i,j \in N' \tag{A.86}$$

$$0 \leqslant \boldsymbol{x_i} \leqslant W, \quad \forall i \in N \tag{A.87}$$

$$0 \leqslant \boldsymbol{x'_i} \leqslant W, \quad \forall i \in N' \tag{A.88}$$

$$0 \leqslant \boldsymbol{y_i} \leqslant L, \quad \forall i \in N \tag{A.89}$$

$$0 \leqslant \boldsymbol{y'_i} \leqslant L, \quad \forall i \in N' \tag{A.90}$$

$$0 \leqslant \boldsymbol{c_i} \leqslant C_{max}, \quad \forall i \in N \tag{A.91}$$

$$0 \leqslant \boldsymbol{c'_i} \leqslant C_{max}, \quad \forall i \in N' \tag{A.92}$$

$$\boldsymbol{f}_{\boldsymbol{ik}} \in \{0,1\}, \quad \forall i \in N,\ k \in M \tag{A.93}$$

$$\boldsymbol{f}'_{\boldsymbol{ik}} \in \{0,1\}, \quad \forall i \in N',\ k \in M' \tag{A.94}$$

$$0 \leqslant \boldsymbol{C_k} \leqslant C_{max}, \quad \forall k \in M \cup M' \tag{A.95}$$

$$\boldsymbol{P_k} \geqslant 0, \quad \forall k \in M \cup M' \tag{A.96}$$

$$\boldsymbol{z_k} \in \{0,1\}, \quad \forall k \in M \cup M' \tag{A.97}$$

$$0 \leqslant \boldsymbol{s_{kl}} \leqslant C_{max}, \quad \forall k,l \in M \cup M' \tag{A.98}$$

$$\boldsymbol{seq_{kl}} \in \{0,1\}, \quad \forall k,l \in M \cup M' \tag{A.99}$$

$$\boldsymbol{proc_{ku}} \in \{0,1\}, \quad \forall k \in M_t \cup M'_t,\ u \in U_t,\ t \in TYPES \tag{A.100}$$

$$\boldsymbol{T}_{max} \geqslant 0 \tag{A.101}$$

References

Balakrishnan, N., Kanet, J.J., Sridharan, V., 1999. Early/tardy scheduling with sequence dependent setups on uniform parallel machines. Comput. Oper. Res. 26, 127–141.

Cao, J.X., Lee, D.H., Chen, J.H., Shi, Q., 2010. The integrated yard truck and yard crane scheduling problem: Benders decomposition-based methods. Transport. Res. Part E: Logist. Transport. Rev. 46, 344–353.

Chang, D., Jiang, Z., Yan, W., He, J., 2010. Integrating berth allocation and quay crane assignments. Transport. Res. Part E: Logist. Transport. Rev. 46, 975–990.

Chen, J.H., Lee, D.H., Cao, J.X., 2012. A combinatorial benders cuts algorithm for the quayside operation problem at container terminals. Transport. Res. Part E: Logist. Transport. Rev. 48, 266–275.

Coene, S., Spieksma, F., Vanden Berghe, G., 2011. The Lockmaster's Problem. Technical Report KBI-1126. KU Leuven. Available at: https://lirias.kuleuven.be/handle/123456789/318524.

Lee, B.K., Kim, K.H., 2010. Optimizing the block size in container yards. Transport. Res. Part E: Logist. Transport. Rev. 46, 120–135.

Lee, D.H., Jin, J.G., Chen, J.H., 2012. Terminal and yard allocation problem for a container transshipment hub with multiple terminals. Transport. Res. Part E: Logist. Transport. Rev. 48, 516–528.

Nauss, R.M., 2008. Optimal sequencing in the presence of setup times for tow/barge traffic through a river lock. Eur. J. Oper. Res. 187, 1268–1281.

Petering, M.E., 2011. Decision support for yard capacity, fleet composition, truck substitutability, and scalability issues at seaport container terminals. Transport. Res. Part E: Logist. Transport. Rev. 47, 85–103.

Pisinger, D., Sigurd, M., 2005. The two-dimensional bin packing problem with variable bin sizes and costs. Discrete Optim. 2, 154–167.

Smith, L., Nauss, R., 2010. Investigating strategic alternatives for improving service in an inland waterway transportation system. Int. J. Strategic Dec. Sci. 1, 6281.

Smith, L., 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. Transport. Res. Part E: Logist. Transport. Rev..

Stahlbock, R., Voß, S., 2008. Operations research at container terminals: a literature update. OR Spect. 30, 1–52.

Verstichel, J., De Causmaecker, P., Spieksma, F., Vanden Berghe, G., in press. Exact and heuristic methods for placing ships in locks. Eur. J. Oper. Res.. http://dx.doi.org/10.1016/j.ejor.2013.06.045.

Verstichel, J., De Causmaecker, P., Vanden Berghe, G., 2011a. Scheduling algorithms for the lock scheduling problem. Proc.– Soc. Behav. Sci. 20, 806–815.

Verstichel, J., Vanden Berghe, G., 2009. A late acceptance algorithm for the lock scheduling problem. Logist. Manage., 457–478.

Verstichel, J., 2012. <http://allserv.kahosl.be/∼jannes/lockplanning/index.html>.