Maastricht University

Inland Waterway Efficiency Through Skipper Collaboration and Joint Speed Optimization

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.
Link to publication

Inland Waterway Efficiency Through Skipper Collaboration and Joint Speed Optimization

Christof Defryn[1], Julian Golak[1], Alexander Grigoriev[1(✉)], and Veerle Timmermans[2]

[1] Maastricht University School of Business and Economics,
P.O.Box 616, 6200 MD Maastricht, The Netherlands
{c.defryn,j.golak,a.grigoriev}@maastrichtuniversity.nl
[2] RWTH Aachen, Department of Management Science,
Kackertstrasse 7, 52072 Aachen, Germany
veerle.tantimmermans@oms.rwth-aachen.de

Abstract. We address the problem of minimizing the aggregated fuel consumption by the vessels in an inland waterway (a river) with a single lock. The fuel consumption of a vessel depends on its velocity and the slower it moves, the less fuel it consumes. Given entry times of the vessels into the waterway and the deadlines before which they need to leave the waterway, we decide on optimal velocities of the vessels that minimize their private fuel consumption. Presence of the lock and possible congestions on the waterway make the problem computationally challenging. First, we prove that in general Nash equilibria might not exist, i.e., if there is no supervision on the vessels velocities, there might not exist a strategy profile from which no vessel can unilaterally deviate to decrease its private fuel consumption. Next, we introduce simple supervision methods to guarantee existence of Nash equilibria. Unfortunately, though a Nash equilibrium can be computed, the aggregated fuel consumption of such a stable solution is high compared to the consumption in a social optimum, where the total fuel consumption is minimized. Therefore, we propose a mechanism involving payments between vessels, guaranteeing Nash equilibria while minimizing the fuel consumption. This mechanism is studied for both the offline setting, where all information is known beforehand, and online setting, where we only know the entry time and deadline of a vessel when it enters the waterway.

Keywords: Lock scheduling · Congestions · Social welfare · Mechanism design · Online scheduling

1 Introduction

The high fuel prices, a congested road network and the increasing demand for transport due to globalization put a high pressure on the existing transportation

network, especially road transport. The growing sense of resource scarcity and climate change motivates companies to rethink their logistical operations and, if possible shift towards a more sustainable transport mode. In comparison to other transportation modes, the use of barges is more sustainable (less greenhouse gas emission) and relatively cheap (due to economies of scale). Moreover, as a single barge can replace over 100 trucks, increased use of the water network is likely to reduce congestion and the number of accidents on the road network. The Netherlands, located around the mouth of multiple important European rivers, has a dense network of over 4600 km of navigable inland waterways [2], on which 36% of all freight transport (in tonne-kilometre) takes place [3].

Besides longer travel times, mainly due to the relatively low density of the network, the high uncertainty in arrival time is one the major drawbacks of freight transport over inland waterways. This uncertainty is caused by the presence of many river obstacles, such as low bridges, narrow river segments, harbors and locks, which gives rise to unexpected congestion and waiting time. This requires the skipper, the person in charge of the boat, to increase the speed afterwards to guarantee an on-time arrival at the destination. However, the operational cost for the skipper is largely determined by the fuel consumption, which is related directly to the required power and, therefore, the speed of the vessel. The required speeding up results therefore in a direct increase of operational costs for the skipper.

In this paper, we investigate how coordination and scheduling of all movement around these river obstacles can help to reduce congestion and waiting times, and therefore increase the efficiency of inland waterway transport. Moreover, by optimizing a recommended speed for each barge between two consecutive obstacles, one can control the arrival times of the vessels at each obstacle, guaranteeing the minimal throughput time and at the same time the minimal total fuel consumption. For a single lock, the strategy of reducing the speed of the vessel to avoid waiting time has resulted in significant economic benefits [14].

2   Literature Review

Lock Scheduling. Existing research on the optimization of river obstacles is mainly focused on lock scheduling. In a single lock scheduling problem, the operating times of a single lock are optimized for a set of vessels with given arriving time at the lock. By batching the vessels together and determining the optimal service time for each batch, the goal is to reduce overall waiting time at the lock.

Passchyn et al. [8] provide a polynomial time algorithm to optimally solve the single lock scheduling problem, given the arrival times of the boats and the capacity of the lock. Passchyn, Briskorn and Spieksma present in [6] a complexity analysis of this problem and provide a polynomial time algorithm that applies to special cases for the single lock scheduling problem with multiple parallel chambers. The problem of physically placing vessels inside the chamber of the lock has been addressed by Verstichel et al. [16,17]. The joint optimization of multiple

sequential locks on the river is considered by Passchyn et al. [7] and Prandtstetter et al. [10]. Here, Prandtstetter et al. propose a Variable Neighborhood Search for solving the problem. Passchyn et al. propose an MILP to find an exact solution, which is also used in the current work. In all the contributions above the vessel speeds and arrival times in the river segment are deterministic and given. In an optimal lock schedule the aim is always to minimize the aggregated fuel cost or emissions, and selfish behavior of skippers is not addressed.

There are also multiple case studies conducted for the lock scheduling problem, focused on specific lock sequences on important waterways in the world. Petersen et al. [9] consider the *Welland Canal* in North America for which they provide a heuristic that employs optimal dynamic programming submodels for scheduling individual locks in order to determine operating schedules for the lock sequence. Smith et al. [13] present a simulation model to evaluate the quality of different heuristics on lock operations on the *Upper Mississippi River* in the US. This research has been extended by Smit et al. [12]. Here, the authors propose a MIP model to solve the lock scheduling problem with sequence-dependent setup- and processing times. On the same river segment, Nauss [5] incorporated the malfunctioning of locks in order to efficiently resolve a queue of vessels that might arise due to the malfunctioning. Also, a model for the lock scheduling problem with multiple parallel chambers for this river layout has been investigated by Ting and Schonfeld [15]. Finally, the *Kiel Canal* is considered by Günter, Lübbecke and Möring [4]. They incorporate collision of ships in their model and provide a heuristic to determine a routing and scheduling to fleet of ships in a collision-free manner.

In contrast to the previous literature, only Passchyn et al. [7] take into account that skippers can choose the speed of their boat, and hence influence the time in which they arrive at the lock. They minimize overall CO_2 emissions by optimizing the speed at which vessels have to approach the locks using a MILP formulation. Although this approach is closely related to the problem addressed in this paper, the authors of [7] focus on minimizing the aggregated emissions without considering the fact that each skipper is mainly interested in minimizing his personal fuel cost and emissions. As a consequence, skippers might deviate from the proposed solution and increase their individual utility. In this paper, we view this problem from a game-theoretic point of view, and propose a schedule in which no skipper can profitably deviate from the proposed solution.

Fuel Reduction. Academic literature on fuel savings has been extensive in the context of ocean vessels. We refer to [11] for a more detailed survey. Though, inland waterways are significantly different compared to the ocean, as there are no 'river' obstacles in the ocean. Research on fuel consumption in inland waterways is sparse. Ting et al. [14] found that the strategy of reducing vessels speed to avoid idle time has resulted in significant economic benefits for a single lock. This may been seen as a key observation for the motivation of the current work. The fact that fuel consumption grows non-linearly in the vehicle's speed is corroborated by Bialystockia and Konovessis [1].

Our Contributions. Previous research on the lock scheduling is based on the assumption that lock operators have the full power to determine the operating schedule for the lock. In practice, this schedule is typically determined using the first come first serve (FIFO) principle based on the order at which vessels arrive at the lock. Skippers that know this have the incentive to speed up when approaching a lock in order to pass their predecessors and get served first. This action leads to longer waiting times before the locks, and increases the operational cost for these skippers due to the higher fuel consumption that is caused by maintaining a higher speed.

In this paper, we aim to minimize the aggregated fuel consumption by the vessels in the river, while keeping in mind that each skipper is a rational individual with the sole goal of minimizing his personal fuel cost or emissions. In the solutions we present, we determine an optimal speed for each individual boat and for each river segment. The positive relation between vessel speed and fuel consumption leads to the observation that maintaining the slowest speed—yet meeting the arrival deadline at the destination harbour—minimizes the total fuel consumption of a single vessel. Unfortunately, even a single lock on the river becomes a source of congestion and the speeds of the vessels have to be adjusted accordingly.

The paper is structured as follows. In Sect. 3, we model the problem as a non-cooperative game and discuss a variety of priority rules that can be used by the lock operators in case multiple vessels approach the lock (possibly in the opposite directions). Moreover, we discuss the existence of *Nash equilibria*— situations in which no skipper can unilaterally deviate from the proposed solution and decrease its individual cost. In Sect. 4, we introduce a *cooperative game* perspective on the traffic optimization problem at hand. We assume that binding contracts between different skippers are possible and propose a mechanism based on monetary payments. This situation will give rise to new Nash equilibria. We design an algorithm that computes these Nash equilibria while minimizing total fuel consumption on the river. Finally, in Sect. 5, we extend this algorithm to comply with an online setting.

3   Non-cooperative Game for Traffic Optimization at River Obstacles

3.1   Mathematical Notation of the System

Without loss of generality, we assume a waterway with a single lock L. Let this lock be defined by its capacity C, i.e., the number of boats that can be leveled up or down simultaneously, and its current state P, indicating whether the level of the water is high (equal to the upstream level) or low (equal to the downstream level). Let T be the time to change the lock state from high to low or vice versa. If a batch of vessels is processed, an additional T_i times units are required for each vessel i in the batch. That time represents the loading and unloading of vessels and varies across different types and sizes of vessels [13].

The total processing time of a batch of vessels is the sum of lockage time T and the individual processing times T_i for every vessel i in the batch. Moreover, let L_u and L_d be the distances between the upstream and downstream end points of the waterway respectively and the lock. From the moment that a vessel is within that distance from the lock, we consider it to be in the system. The complete system is, therefore, determined by the tuple $L = \{C, P, T, L_u, L_d\}$.

Now, let U and D be sets of vessels, that sail upstream or downstream respectively and let $S = U \cup D$ be the set of all vessels. Let $n = |S|$ be the size of the entire fleet. For each vessel $i \in U$, we are given an arrival time at the upstream end point of the river, denoted by a_i, and a deadline d_i, the latest time when the vessel has to reach the downstream end point of the waterway. Similarly, a_j and d_j are defined for each vessel $j \in D$, sailing in the opposite direction. Furthermore, we assume that vessels in set S are ordered according to their arrival times and that between any two sequential vessel arrivals at least ε time elapses. Finally, let $v_{i,p}$ denote the speed of vessel i along river segment $p \in \{u, d\}$, where u and d represent the upstream and downstream segments respectively. We assume the minimum and the maximum speed for any vessel is bounded by $v_{\min}$ and $v_{\max}$.

3.2   Model Definition

In the game, each vessel $i \in U \cup D$ decides on $v_{i,d}$ and $v_{i,u} \in [v_{\min}, v_{\max}]$, such that $v_i = (v_{i,d}, v_{i,u})$. Furthermore, let v_{-i} denote the strategy profile of every player in the game except for i and let $\mathbf{v} = (v_i, v_{-i})$. Note that only constant speeds have been specified for both, upstream and downstream, waterway segments. Due to the convexity of the cost function, defined below, skippers will have no incentive to alter their speed midway of the segments. The assumption of constant speeds is relaxed, when an online setting of the game is considered, in Sect. 5. To illustrate the game, consider the following example.

Example 1. Assume three vessels (see also Fig. 1): 1 and 2 sailing upstream and 3 sailing downstream. The waterway is 20 km long, and the lock is placed in the middle of the waterway. As a result, $L_u = L_d = 10$. The lock has an infinite capacity and $T = T_1 = T_2 = T_3 = 0.5$. The entry/arrival times of the vessels are as follows: $a_1 = 0$, $a_2 = \varepsilon$ and $a_3 = 2\varepsilon$. Moreover, we know that $(v_{1,u}, v_{1,d}) = (5, 5)$, $(v_{2,u}, v_{2,d}) = (10, 5)$ and $(v_{3,u}, v_{3,d}) = (5, 10)$. Given the current speeds, vessel 1 arrives at the lock at time 2, vessel 2 at time $1 + \varepsilon$ and vessel 3 is expected to arrive at the lock at time $1 + 2\varepsilon$.

The total fuel consumption is given by the function $E(v)$, where v represents the speed of the vessel. The function is measured in tons per kilometer. We assume that fuel consumption is equal to zero if the vessel is not moving, i.e., its speed is equal to zero, and vessels are only standing still inside the lock. Following the conventions from the related literature, we assume convexity of $E(v)$, $v > 0$ (see [7]).
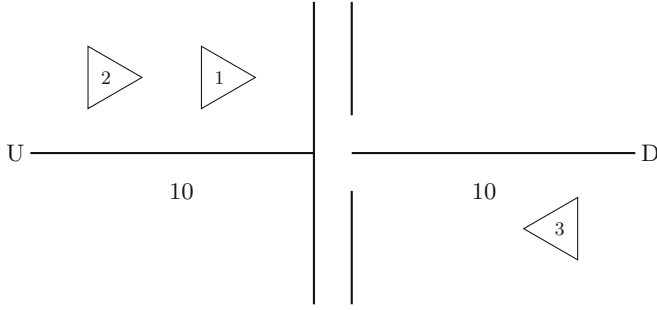
Fig. 1. The setup of locks and vessel for Example 1

To further simplify notations, and without loss of generalization, we consider the fuel consumption function to be the same for every vessel and equal to

$$E_i(v_i) = L_u E(v_{i,u}) + L_d E(v_{i,d}). \tag{1}$$

The fuel consumption of the entire fleet can therefore be written as

$$E_{tot}(\mathbf{v}) = \sum_{i \in S} E_i(v_i). \tag{2}$$

Each skipper i aims to minimize its total fuel consumption $E_i(v_i)$, given its deadline (denoted as d_i) on the arrival time at the destination. This is considered a hard constraint. Arriving at the destination after the predefined deadline is considered infeasible, represented by an infinite penalty cost. In case the deadline is unrestrictive for the vessel, it will sail at the minimum speed $v_{\min}$. Therefore, we define the cost function for skipper $i \in S$ by

$$C_i(\mathbf{v}) = \begin{cases} E_i(v_i) & \text{if } a_i + L_u/v_{i,u} + L_d/v_{i,d} + q_i(\mathbf{v}) \leq d_i; \\ \infty & \text{otherwise} , \end{cases} \tag{3}$$

where $q_i(\mathbf{v})$ is the total processing time of vessel i at the lock, i.e., waiting time before entering the lock plus the lock re-level time T and the individual loading times. This waiting time depends on the congestion induced by the strategy profile, i.e., individual speeds of all vessels in the system.

We now define the *social cost* $C(\mathbf{v})$ of a strategy profile $\mathbf{v}$ as the aggregated cost of all players, defined as

$$C(\mathbf{v}) = \sum_{i \in S} C_i(\mathbf{v}). \tag{4}$$

The strategy profile $\mathbf{v}$ that minimizes the social cost is called the *social optimum*, and has a social cost of

$$C_{opt} = \min_{\mathbf{v}} C(\mathbf{v}). \tag{5}$$

3.3   Nash Equilibrium and Queuing Discipline at the Lock

In a *non-cooperative* game (without binding contracts between the skippers), we assume that skippers act selfishly and aim to minimize their individual costs. One of the most important tools that game theorists have at their disposal is the *Nash equilibrium*: a strategy profile $\mathbf{v}^*$ where no vessel can unilaterally deviate from its current strategy v_i^* and decrease its current cost. More formally, $\mathbf{v}^*$ is a Nash equilibrium if and only if

$$C_i(v_i^*, v_{-i}^*) \leq C_i(v_i, v_{-i}^*), \forall v_i \in \mathcal{V}_i. \tag{6}$$

The importance of the Nash equilibrium comes from the natural observation that agents/players/skippers are rather interested in selfishly minimizing their individual costs than reducing the social cost, i.e., the total cost of the entire fleet. The Nash equilibrium is calculated by minimizing the regret of the individual players, where regret is defined as the cost they could have saved by altering the strategy.

The existence of the Nash Equilibrium is dependent on the waiting time of vessels in front of the locks. In turn, this waiting time is subject to the *queuing discipline* of the lock. This queuing discipline dictates the order in which vessels are served by the lock operator. As the waiting time impacts the optimal (required) speed after the lock, the queuing discipline directly affects the cost of each skipper. Therefore, different lock mechanisms yield different characteristics of the game. We consider the following three simple lock mechanisms:

Mechanism 1: Lock FIFO. For any $i, j \in U \cup D$, vessel i is served by the lock before vessel j if i arrives at the lock before j. If vessels i and j arrive at the lock at the same time, i will be served first if $a_i < a_j$.

Mechanism 2: System FIFO. For any $i, j \in U \cup D$, vessel i is served by the lock before vessel j if $a_i < a_j$.

Mechanism 3: System FIFO with filling idle time. Consider vessel $i \in U \cup D$. Assume that skippers choose strategies sequentially and all $(v_j)_{j=1,\ldots,i-1}$ are given. For any $i, j \in U \cup D$ such that $j < i$, vessel i is served before j if it does not affect the time of departure of vessel j determined by the strategy profile $(v_j)_{j=1,\ldots,i-1}$.

The following example illustrates how these three mechanisms work and how they affect the payoff of a strategy profile.

Example 2. Consider again the setup of Example 1. Let us remind that the entry/arrival times of the vessels were $a_1 = 0$, $a_2 = \varepsilon$ and $a_3 = 2\varepsilon$. Furthermore, given the current speeds of the vessels, the arrival times at the locks are 2, $1 + \varepsilon$ and $1 + 2\varepsilon$, for vessels 1, 2 and 3, respectively.

First, if the lock operates under Mechanism 1, only the arrival times at the lock are relevant. Note that vessel 2 arrives at the lock first, vessel 3 second and

vessel 1 is the last one. As vessels are processed in order of arrival time, the waiting times under the strategy profile are $2 + \varepsilon, 1, 2 - \varepsilon$ for vessel 1, 2 and 3 respectively.

Second, under mechanism 2, only the arrival times into the system are relevant. Note that vessel 1 arrives first in the system, vessel 2 second and vessel 3 last. The waiting times are $1, 2 - \varepsilon, 3 - 2\varepsilon$ for vessel 1, 2 and 3 respectively.

Lastly, when Mechanism 3 is applied, the arrival times into the system and at the locks are relevant. Note that if vessel 2 or 3 is served before vessel 1, the exit from the lock of vessel 1 would be delayed. Since vessel 1 arrives first into the system, it has priority and therefore it is processed first. Once vessel 1 is processed, the lock is open to the downstream side and vessels 2 and 3 are waiting on the upstream and downstream segments, respectively. Vessel 2 arrives first into the system, therefore it has priority. However, when vessel 1 has been processed, the lock is on the side of vessel 3. Thus, serving vessel 3 does not affect the waiting time of vessel 2. Therefore, under this mechanism, vessel 3 is processed second and vessel 2 is processed last. The waiting times are now equal to $1, 4 - \varepsilon, 3 - 2\varepsilon$ for vessel 1, 2 and 3 respectively.

Since the choice of a lock mechanism influences the behavior of vessels, it also influences the existence of equilibria. Under the assumption of Mechanism 1, where the priority of vessels is determined by the arrival of vessels at the lock, equilibria might not exist, which is shown in the following example.

Example 3 (Mechanism 1). Assume there are two vessels: vessel 1 sailing upstream and vessel 2 sailing downstream. The complete river segment is again 20 km long, and the lock is placed in the middle of the waterway, hence, $L_u = L_d = 10$. The lock has capacity of 1 (though, any positive capacity will do) and its duration T and loading times T_1 and T_2 are set to 0.5. We assume that the fuel consumption function $E(v)$ is convex, non-negative and increasing in speeds $v_{i,p} \in [5, 10]$, $p \in \{u, d\}$. We assume that the lock starts on the upstream side, but can switch to the downstream side in time whenever vessel 2 is the first one to arrive at the lock. We assume the arrival times in the system are given by $a_1 = 0$ and $a_2 = \epsilon$ and the deadlines are $d_1 = 4$ and $d_2 = 4 + \epsilon$. Note that whenever a vessel has decided on its speed up to the lock, there is a unique speed after the lock that minimizes the fuel consumption such that the deadline, if possible, will not be exceeded. Therefore, the strategy of the vessels can be expressed in their speed before the lock (denoted by v_1 for vessel 1, and v_2 for vessel 2). We divide all possible speed scenarios into six cases, presented in Table 1. We see that in every strategy profile, there is a skipper that can decrease its fuel consumption by changing its speed. Hence, there does not exist a Nash equilibrium. Note that for this example, $v_{opt} = 6.\bar{6}$, i.e., the optimal speed for each vessel if it would be the only vessel on this waterway segment.

Under lock operating mechanisms 2 and 3, however, the Nash equilibrium does exist as the order in which the vessels enter the lock is determined solely by the order in which they arrive into the system. Hence, it cannot occur that vessels race each other to the lock, which is the main idea behind our previous example. Under these two mechanisms, vessels cannot affect the costs of vessels

Table 1. Speed scenarios for example 3.

Scenario	v_1	v_2	Improving move
1	10	$[5, v_{opt}]$	Player 1 should decrease v_1 to v_{opt}
2	10	$(v_{opt}, 10]$	Player 2 should decrease v_2 to 5
3	$(5, 10)$	$v_2 \leq v_1$	Player 2 should increase v_2 to 10
4	$(5, 10)$	$v_2 \geq v_1$	Player 1 should increase v_1 to 10
5	5	$(v_{opt}, 10]$	Player 2 should decrease v_2 to v_{opt}
6	5	$[5, v_{opt}]$	Player 1 should increase v_1 to v_{opt}

that entered the river section earlier. This implies that vessels can sequentially choose a best response, taking into account the arrival times of the previous vessels. We prove this statement more formally in the next theorem.

Theorem 1. *Consider the single lock scheduling problem, where the lock operates under Mechanism 2 or 3. Then, each game possesses at least one Nash equilibrium.*

Proof. We provide a generic construction of a strategy profile and show that this strategy profile constitutes a Nash equilibrium. Observe that under both Mechanism 2 and 3, for any speed v_i, the waiting time of vessel i, $q_i(\mathbf{v})$, only depends on the vessels arriving earlier in the system than vessel i. Consequently, knowing the strategies $v_1, \ldots, v_{i-1}$ is sufficient to determine optimal strategy v_i.

By construction of the strategy profile, it is apparent that each vessel i chooses its best possible strategy with respect to the early arriving vessels. Also, strategies of vessels that arrive later cannot influence the costs experienced by vessel i. Hence, vessel i can not decrease its private cost and therefore the resulting strategy profile is a Nash equilibrium.

Note, that the difference between the two mechanisms occurs in the individual optimization of strategies: under Mechanism 3 the waiting times caused by profile $\mathbf{v}$ might be different from the waiting times under Mechanism 2 using the same vector $\mathbf{v}$. However, the implications and the arguments stay the same: the cost for vessel i is only affected by the strategies of the first $i - 1$ vessels.   □

A central authority could guarantee the existence of a Nash equilibrium by forcing the lock operators to use Mechanism 2 or Mechanism 3. However, the fact that a Nash equilibrium exists does not tell us anything about its cost efficiency. Selfish decision making may lead to a Nash equilibrium with a high social cost, which then leads to a waste of resources and high pollution on rivers. In Mechanism 2 and 3, individual costs highly depend on the strategies taken by the previous vessels. Therefore, selfish decision making may lead to the scenario in which later vessels are unable to cross the river segment before their deadline, resulting in a Nash equilibrium with an infinitely high social cost. Such scenario indicates that the *price of anarchy* of this game (the ratio between the highest

social cost of any Nash equilibrium and the minimal social cost) is unbounded. This becomes apparent in the following example.

Example 4. We consider the same instance as in Example 3. However, this time we assume that the lock operates under Mechanism 2. We construct a Nash equilibrium with the procedure described in the proof of Theorem 1. This implies that $v_1^* = (20/3, 20/3)$. Note that there is no strategy in the strategy space of vessel 2, such that it passes the river segment before its deadline. Thus the social cost of this instance is infinitely high.

There is a strategy profile such that both vessels cross the river before their deadlines. More precisely, $\mathbf{v} = (5, 10)$ leads to a finite costs for both vessels. Because of this, the price of anarchy of the game at hand is unbounded. Note that the same results hold, when the lock is assumed to operate under Mechanism 3.

The goal of this section was to show that, though the concept of a Nash equilibrium seems appealing, in the non-cooperative setting it might not exist or it might be extremely inefficient compared to a socially optimal strategy profile. In the next section, we review the problem from a cooperative game point of view as we introduce the possibility to make binding contracts between the vessels.

4 Cooperative Game for Traffic Optimization at River Obstacles

We now assume that the vessels can make binding contracts and allow *payments* between skippers. As a result, the agents/skippers can incentivize their counter-agents to adapt their speeds by reimbursing their extra costs. We aim to find a solution concept that is cost optimal while making sure that no player can profit from a unilateral deviation from the social optimum. More precisely, we introduce a payment system that fulfills two criteria:

1. By participating in the payment system, the cost of a player can never be higher than when he/she did not participate.
2. The payment system should give a vessel an incentive to behave as in the social optimum.

In this section, we consider full information about the lock, river segments and vessels that will enter the system to be known in advance. An online variant of this problem is presented in Sect. 5, in which only the information about the river segment and the lock are publicly known while information about the vessels becomes only available when a vessel physically enters the waterway. First, we propose an algorithm that returns for each vessel a speed v_i, and the payment scheme $P_{i,j}$ indicating payment of skipper i to skipper j for the requested velocity adjustment. Second, we prove that the solution proposed by the algorithm satisfies the two criteria mentioned above.

4.1   Iterative Payment Scheme Algorithm

The algorithm sequentially determines optimal speeds and payments in the order of vessels arrival by considering all vessels 1 through i, denoted by the set $\bar{S}_i$. In the first iteration, only vessel 1 is considered and its optimal speed is determined. Let ζ_1 be the operating cost associated with this strategy such that $\zeta_1 = C_1(v_1)$. During future iterations, it will be ensured that the cost for this skipper will not go above the cost of this benchmark situation. To do this, other skippers should fully reimburse any cost increase that results from changing the strategy for the skipper.

Now, let $P^*_{j,j'}$ be the payment scheme for all $j' < j < i$ at iteration i. Moreover, all guaranteed costs ζ_j are considered to be known for all $j < i$. To determine the speeds v_j for all $j \in \bar{S}_i$ and payments $P_{i,j}$ for all $j < i$, we solve the following optimization problem: determine new velocities of the vessels from $\bar{S}_i$ such that the sum of the costs and payments for vessel i is minimized, while the total cost of each vessel $j < i$ is at most ζ_j. Then, we compute the value of the guaranteed cost ζ_i of player i. More formally, we define following relations.

$$C_{opt,k}(\bar{S}_i) := C_k\big((v^*_j)_{j\in\bar{S}_i}\big) \quad k \in \bar{S}_i, \tag{7}$$

$$P^*_{i,k} := C_{opt,k}(\bar{S}_i) - \zeta_k - \sum_{j\in\bar{S}_{i-1}:j>k} P^*_{jk} \quad k \in \bar{S}_{i-1}, \tag{8}$$

$$\zeta_i := C_{opt,i}(\bar{S}_i), \tag{9}$$

where v^* and P^* are the solutions to the following optimization problem. For a given vessel $i > 1$, having computed all optimal values P^* for all $i' < i$, the mathematical program reads

$$\min_{(v_j)_{j\in\bar{S}_i};P_{i,j}} \left(C_i\big((v_j)_{j\in\bar{S}_i}\big) + \sum_{k\in\bar{S}_{i-1}} P_{i,k} \right) \tag{10}$$

$$C_k\big((v_j)_{j\in\bar{S}_i}\big) - P_{i,k} - \sum_{\substack{j\in\bar{S}_{i-1}\\ j>k}} P^*_{j,k} \leq \zeta_k, \qquad k \in \bar{S}_{i-1}. \tag{11}$$

Algorithm 1 represents the payment system which outputs both optimal speeds and payments for all skippers. Note that the optimization problem has been replaced by a computation of the social optimal speeds. This is a valid substitution due to Theorem 2 below.

Input: $(L := (C, T, P, L_u, L_d), U, D, (a_i, d_i, v_{\min}, v_{\max})_{i \in U \cup D})$
Output: Optimal set of speeds and payments.
$\bar{S}_1 = \{1\}$;
$\zeta_1 = C_{opt}(S_i)$;
for i *from 2 to n* **do**

> $\bar{S}_i = \bar{S}_{i-1} \cup \{i\}$;
> Compute $C_{opt}(\bar{S}_i)$ and let $(v_j^*)_{j \in \bar{S}_i}$ be the optimal parameters;
> $C_{opt,k}(\bar{S}_i) := C_k\big((v_j^*)_{j \in \bar{S}_i}\big) \quad \forall k \in \bar{S}_i$;
> $P_{i,k}^* := C_{opt,k}(\bar{S}_i) - \zeta_k - \sum_{j \in \bar{S}_{i-1}:j>k} P_{jk}^* \quad \forall \, k \in \bar{S}_{i-1}$;
> $\zeta_i := C_{opt,k}(\bar{S}_i)$;

end
return $\big((v_j^*)_{j \in S}, (P_{ij}^*)_{i,j \in S}\big)$

Algorithm 1. Payment mechanism

The subroutine computing of $C_{opt}(\bar{S}_i)$ can be implemented in various ways. In the Appendix, we provide a MIP-formulation to solve the lock scheduling problem to optimality. This formulation is based on the model in [7] and has been adjusted to comply with our problem statement. Moreover, we show that the problem is NP-complete in the strong sense, this way motivating design of MIP-formulations and approximation algorithms for the problem. Regarding existence of good approximation algorithms, we leave this question open but stress that any α-approximation algorithm directly leads to an α-approximate Nash equilibrium.

Given a solution to the optimization problem above, in Theorem 2, we show that the optimal speeds in that problem are equivalent to the speeds in the social optimum computed on vessels in the set $\bar{S}_i$.

Theorem 2. $(v_j^*)_{j \in \bar{S}_i} = \operatorname{argmin}_{(v_j)_{j \in \bar{S}_i}} \sum_{k \in \bar{S}_i} C_k(v_j)_{j \in \bar{S}_i}$.

Proof. For each $k \in \bar{S}_{i-1}$,

$$P_{i,k} = C_k\big((v_j)_{j \in \bar{S}_i}\big) - \zeta_k - \sum_{\substack{j \in \bar{S}_{i-1} \\ j>k}} P_{j,k}^* \tag{12}$$

and therefore the optimization problem can be written as

$$(v_j^*)_{j \in \bar{S}_i} = \operatorname*{argmin}_{(v_j)_{j \in \bar{S}_i}} C_i\big((v_j)_{j \in \bar{S}_i}\big) + \sum_{k \in \bar{S}_{i-1}} P_{i,k} \tag{13}$$

$$= \operatorname*{argmin}_{(v_j)_{j \in \bar{S}_i}} C_i\big((v_j)_{j \in \bar{S}_i}\big) + \sum_{k \in \bar{S}_{i-1}} \left(C_k\big((v_j)_{j \in \bar{S}_i}\big) - \zeta_k - \sum_{\substack{j \in \bar{S}_{i-1} \\ j>k}} P_{j,k}^* \right)$$

$$\tag{14}$$

$$= \underset{(v_j)_{j \in \bar{S}_i}}{\operatorname{argmin}} \sum_{k \in \bar{S}_i} C_k\big((v_j)_{j \in \bar{S}_i}\big) - \sum_{k \in \bar{S}_{i-1}} \Big(\zeta_k + \sum_{\substack{j \in \bar{S}_{i-1} \\ j > k}} P_{j,k}^*\Big) \tag{15}$$

$$= \underset{(v_j)_{j \in \bar{S}_i}}{\operatorname{argmin}} \sum_{k \in \bar{S}_i} C_k\big((v_j)_{j \in \bar{S}_i}\big), \tag{16}$$

$\square$

Lastly, in Theorem 3, we show that in the i-th iteration of the algorithm the best response for skipper i is to obey the payment mechanism. This means that the guaranteed cost of vessel i plus the payments this skipper has to pay to all other skipper is lower than the cost of any strategy not involving the payments.

Theorem 3. *In Algorithm 1 for each $\bar{S}_i$, it holds that*

$$\zeta_i + \sum_{k \in \bar{S}_{i-1}} P_{i,k} \le C_i(v_i, (v_j^*)_{j \in \bar{S}_{i-1}}) \text{ for all } v_i \in V_i. \tag{17}$$

Proof. Note that after every iteration i, it holds for every $k \in \bar{S}_{i-1}$

$$P_{i,k}^* = C_{opt,k}(\bar{S}_i) - \zeta_k - \sum_{\substack{j \in \bar{S}_{i-1} \\ j > k}} P_{jk}^* \tag{18}$$

$$C_{opt,k}(\bar{S}_i) = \zeta_k + \sum_{\substack{j \in \bar{S}_i \\ j > k}} P_{jk}^* \tag{19}$$

This leads to the following equality.

$$\zeta_i + \sum_{k \in \bar{S}_{i-1}} P_{ik}^* = C_{opt,i}(\bar{S}_i) + \sum_{k \in \bar{S}_{i-1}} \Big(C_{opt,k}(\bar{S}_i) - \zeta_k - \sum_{\substack{j \in \bar{S}_{i-1} \\ j > k}} P_{jk}^*\Big) \tag{20}$$

$$= \sum_{k \in \bar{S}_i} C_{opt,k}(\bar{S}_i) - \sum_{k \in \bar{S}_{i-1}} \Big(\zeta_k + \sum_{\substack{j \in \bar{S}_{i-1} \\ j > k}} P_{jk}^*\Big) \tag{21}$$

$$= \sum_{k \in \bar{S}_i} C_{opt,k}(\bar{S}_i) - \sum_{k \in \bar{S}_{i-1}} C_{opt,k}(\bar{S}_{i-1}) \tag{22}$$

$$= C_{opt}(\bar{S}_i) - C_{opt}(\bar{S}_{i-1}) \tag{23}$$

Furthermore, we know that

$$C_{opt}(\bar{S}_i) \le C_i(v_i, (v_j^*)_{j \in \bar{S}_{i-1}}) + C_{opt}(\bar{S}_{i-1}) \text{ for all } v_i \in V_i \tag{24}$$

$$C_{opt}(\bar{S}_i) - C_{opt}(\bar{S}_{i-1}) \le C_i(v_i, (v_j^*)_{j \in \bar{S}_{i-1}}) \text{ for all } v_i \in V_i \tag{25}$$

$$\zeta_i + \sum_{k \in \bar{S}_{i-1}} P_{ik}^* \le C_i(v_i, (v_j^*)_{j \in \bar{S}_{i-1}}) \text{ for all } v_i \in V_i \tag{26}$$

$\square$

From Theorems 2 and 3, it follows that the stated criteria for an efficient payment mechanism are fulfilled by Algorithm 1.

5   Online Setting

The assumption of perfect information on arrival times is likely to be violated in real-life. That is, there is no information prior to the arrival of the vessels at the boundaries of the system. Each time a vessel enters, the optimal speed and payments are recomputed taking into account the location of the vessels already present on the waterway. Note that the definition of a social optimum and a best response of a player are dependent on the information setting of the game. Therefore, we have to dynamically redefine/adjust these quantities in an online setting.

Let the distance between vessel i and the exit of the waterway at time t be denoted by h_i^t. The best response of vessel i, given the strategies of the other vessels, is defined as the strategy that minimizes the cost of vessel i conditional on the position of the other vessels at time t. The cost of vessel i under strategy profile $\mathbf{v}$ conditional on the position of all vessels in set $\bar{S}$ at time t is denoted as $C_i\big((v_j)_{j\in\bar{S}}|(h_j^t)_{j\in\bar{S}}\big)$. The social optimum is defined as a strategy profile, which provides the lowest possible cost given the positions of vessels in $\bar{S}$ at time t.

Similar to the offline setting, the algorithm sequentially determines optimal speeds and payments at the arrival of each vessel. In each iteration, a set $\bar{S}_i$ is constructed containing all vessels currently in the system. Assume that vessel i arrives and vessels in $\bar{S} = \{k, k+1, \ldots, i\}$ have not left the waterway yet. Assume that the payments $P_{j,j'}^*$ for all $k \leq j' < j < i$ and the guaranteed costs ζ_j for all $k \leq j < i$ are given. Since each vessel is at a different position, payments and costs are normalized to units per kilometers. Therefore, we solve the following optimization problem: determine new velocities of the vessels from $\bar{S}_i$ such that the sum of the costs and payments for vessel i is minimized, while the normalized total cost of each vessel $k \leq j < i$ is at most the normalized guaranteed cost. Given the following relations

$$C_{opt,k}^{a_i}(\bar{S}_i) = C_k\big((v_j^*)_{j\in\bar{S}_i}|(h_j^{a_i})_{j\in\bar{S}_i}\big) \ \ \forall\, k \in \bar{S}_i, \tag{27}$$

$$P_{i,k}^* := C_{opt,k}^{a_i}(\bar{S}_i) - h_k^{a_i}\Big(\sum_{\substack{j\in\bar{S}_i\setminus\{i\}\\ j>k}} \Big(\frac{P_{j,k}^*}{h_k^{a_j}}\Big) + \frac{\zeta_k}{l_d+l_u}\Big) \ \ \forall\, k \in \bar{S}_i\setminus\{i\}, \tag{28}$$

$$\zeta_i := C_{opt,i}^{a_i}(\bar{S}_i) \tag{29}$$

we define the online optimization problem as

$$(v_j^*)_{j\in\bar{S}_i} \in \operatorname*{argmin}_{(v_j)_{j\in\bar{S}_i}:P_{i,j}} \left(C_i\big((v_j)_{j\in\bar{S}_i}|(h_j^{a_i})_{j\in\bar{S}_i}\big) + \sum_{k\in\bar{S}_i\setminus\{i\}} P_{i,k} \right) \tag{30}$$

s.t

$$\frac{C_k\big((v_j)_{j\in\bar{S}_i}|(h_j^{a_i})_{j\in\bar{S}_i}\big)}{h_k^{a_i}} - \frac{P_{i,k}}{h_k^{a_i}} - \sum_{\substack{j\in\bar{S}_i\setminus\{i\}\\ j>k}} \frac{P_{j,k}^*}{h_k^{a_j}} \leq \frac{\zeta_k}{l_d+l_u} \quad k \in \bar{S}_i\setminus\{i\}. \tag{31}$$

Again, it can be shown that the two conditions for an efficient payment mechanism are fulfilled in the online setting. The proof is similar to the one discussed in the offline case. The resulting algorithm is given in Algorithm 2.

Input: $(L := (C, T, P, L_u, L_d), U, D, (a_i, d_i, v_{\min}, v_{\max})_{i \in U \cup D})$
Output: Optimal set of speeds and payments.
vessel i arrives in the system at time a_i:
For each vessel currently present in the waterway, update the distance to the destination;
Let $\bar{S}_i$ be the set of vessels in the waterway at time a_i;
if $\bar{S}_i \neq \emptyset$ **then**

> Compute $C_{opt}^{a_i}(\bar{S}_i)$ and let $(v_j^*)_{j \in \bar{S}_i}$ be the optimal parameters;
>
> $C_{opt,k}^{a_i}(\bar{S}_i) = C_k\big((v_j^*)_{j \in \bar{S}_i} | (h_j^{a_i})_{j \in \bar{S}_i}\big) \ \ \forall \, k \in \bar{S}_i;$
>
> $P_{i,k}^* := C_{opt,k}^{a_i}(\bar{S}_i) - h_k^{a_i} \Big(\sum_{\substack{j \in \bar{S}_i \setminus \{i\} \\ j > k}} \Big(\frac{P_{j,k}^*}{h_k^{a_j}}\Big) + \frac{\zeta_k}{l_d + l_u}\Big) \ \ \forall \, k \in \bar{S}_i \setminus \{i\};$
>
> $\zeta_i := C_{opt,i}^{a_i}(\bar{S}_i);$

else

> $\zeta_i = \min_{v_i} C_i(v_i);$

end

Algorithm 2. Payment mechanism Online Setting

Note that whenever a vessel enters the lock, its total fuel cost and payments to the other vessels are known, and will not change anymore. Hence, the lock operator can also operate as a bank: whenever a vessel crosses the lock, it pays (or receives) the payments. This implies that the lock operator needs a cash reserve, as it is likely that the first vessels entering the lock receive money from the vessels that did not arrive at the lock yet. Clearly, this cash reserve needs to be at most the cost of an optimal profile minus the minimum fuel cost of all earlier vessels. In the journal version of the paper we give a simple and insightful example where the cash reserve is actually completely needed.

References

1. Bialystockia, N., Konovessis, K.: On the estimation of vessel's fuel consumption and speed curve: a statistical approach. J. Ocean Eng. Sci. **1**(2), 157–166 (2016)
2. Eurostat: Navigable inland waterways, by horizontal dimensions of vessels and pushed convoys (2016). http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=iww_if_hordim&lang=en. Accessed 1 Apr 2019

3. Inland Navigation in Europe, Market Observation. Central commission for the navigation of the Rhine, annual report (2017). https://www.inland-navigation-market.org/wp-content/uploads/2017/09/CCNR_annual_report_EN_Q2_2017_BD_-1.pdf. Accessed 1 Apr 2019
4. Günther, E., Lübbecke, M.E., Möhring, R.H.: Vessel traffic optimization for the Kiel canal. TRISTAN VII Book of Extended Abstracts 104 (2010)
5. Nauss, R.M.: Optimal sequencing in the presence of setup times for tow/barge traffic through a river lock. Eur. J. Oper. Res. **187**(3), 1268–1281 (2008)
6. Passchyn, W., Briskorn, D., and Spieksma, F.C.R.: No-wait scheduling for locks. Technical Report KBI_1605, KU Leuven, Research group Operations Research and Business Statistics, Leuven, Belgium (2016)
7. Passchyn, W., Briskorn, D., Spieksma, F.C.R.: Mathematical programming models for lock scheduling with an emission objective. Eur. J. Oper. Res. **248**(3), 802–814 (2016)
8. Passchyn, W., Coene, S., Briskorn, D., Hurink, J.L., Spieksma, F.C.R., Vanden Berghe, G.: The lockmaster's problem. Eur. J. Oper. Res. **251**(2), 432–441 (2016)
9. Petersen, E.R., Taylor, A.J.: An optimal scheduling system for the Welland Canal. Transp. Sci. **22**(3), 173–185 (1988)
10. Prandtstetter, M., Ritzinger, U., Schmidt, P., Ruthmair, M.: A variable neighborhood search approach for the interdependent lock scheduling problem. In: Ochoa, G., Chicano, F. (eds.) EvoCOP 2015. LNCS, vol. 9026, pp. 36–47. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16468-7_4
11. Psaraftis, H.N., Kontovas, C.A.: Speed models for energy-efficient maritime transportation: a taxonomy and survey. Transp. Res. Part C: Emerg. Technol. **26**, 331–351 (2013)
12. Smith, L.D., Nauss, R.M., Mattfeld, D.C., Li, J., Ehmke, J.F., Reindl, M.: Scheduling operations at system choke points with sequence-dependent delays and processing times. Transp. Res. Part E: Logistics Transp. Rev. **47**(5), 669–680 (2011)
13. Smith, L.D., Sweeney, D.C., Campbell, J.F.: Simulation of alternative approaches to relieving congestion at locks in a river transportion system. J. Oper. Res. Soc. **60**(4), 519–533 (2009)
14. Ching-Jung, T., Schonfeld, P.: Effects of speed control on tow travel costs. J. Waterw. Port Coastal Ocean Eng. **125**(4), 203–206 (1999)
15. Ching-Jung, T., Schonfeld, P.: Control alternatives at a waterway lock. J. Waterw. Port Coastal Ocean Eng. **127**(2), 89–96 (2001)
16. Verstichel, J., De Causmaecker, P., Spieksma, F.C.R., Vanden Berghe, G.: Exact and heuristic methods for placing vessels in locks. Eur. J. Oper. Res. **235**(2), 387–398 (2014)
17. Verstichel, J., De Causmaecker, P., Spieksma, F.C.R., Vanden Berghe, G.: The generalized lock scheduling problem: an exact approach. Transp. Res. Part E: Logistics Transp. Rev. **65**, 16–34 (2014)