

Designing robust liner shipping schedules: Optimizing recovery actions and buffer times

Judith Mulder¹, Rommert Dekker¹ and Mehdi Sharifyazdi²

¹Econometric Institute, Erasmus University Rotterdam, the Netherlands

²Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Sweden

Econometric Institute Report EI 2012-30

Abstract

Maritime transport is an important mode of transport in international trade. It is important for liner shipping companies to maintain cost efficient and robust liner shipping networks. Regularly, they set up pro-forma schedules, yet it is difficult to stay on time and we consider the problem to manage the delays. Therefore, we need to determine an optimal recovery policy and buffer time distribution of the ship route in order to minimize the total costs associated to delays and recovery actions, such as increasing sailing speed. We will introduce a mixed integer programming formulation and two heuristical methods to solve this problem. We found a cost decrease of 14.8% after optimizing the buffer time distribution based on an existing liner shipping route compared to the cost of the current route schedule.

1 Introduction

In liner shipping networks ships follow a fixed route within a fixed time schedule. However, ships can encounter delays both when they are sailing between ports and when they are berthing in

a port. Shipping companies buy a certain time slot in terminals in ports for a specified number of berths. Because terminals have their own planning regarding time and berths, the delay of a ship has consequences for both ships and ports. For ships, the consequences will depend on the amount of delay and on the number of berths they need.

Additional costs are associated with a delayed arrival of a ship in a certain port. These costs can be categorized in three different groups. First, the terminal can encounter extra costs, for example because the berth schedule has to be adapted. The containers that have to be loaded on the ship, are placed near the scheduled berth. If the ship is delayed and has to arrive at another berth, the costs of transporting the container to the ship are higher than planned. The terminal can charge the shipping company for these extra costs. Next, the containers that are unloaded in the port have to be transported to their destination. When a ship is delayed, the containers can be unloaded too late, so that they miss their connection. Finally, ships can encounter demurrage costs. Demurrage is an additional cost that represents liquidated damages for delays. Demurrage occurs when the ship is prevented from (un)loading the containers within the scheduled time.

Delay can be caused by several reasons, including terminal operations delay, port access delay, maritime passage delay and chance (Notteboom 2006). Because of the delay, a shipping company will encounter additional costs, as explained above. Besides, the probability that the ship will also have a delay in the next ports to visit is increasing. Furthermore, the incurred delay can increase during the round tour, because delayed ships have to wait until they can enter a port or because new delays are encountered. When ships are delayed, they can reduce their delay by taking recovery actions against certain costs. Examples of recovery actions are increasing the sailing speed or port handling capacity. Furthermore, a buffer time can be incorporated in the route to capture (a part of) the delay. The total buffer time of a route depends on the time needed to perform one round tour of the route. Since liner shipping routes are usually serviced once a week, the round tour time is rounded to an integer number of weeks. The total buffer time is chosen in such a way that the time of a single round tour including buffer times equals an integer number of weeks.

The goal of the problem studied in this paper is to determine a recovery policy and buffer time allocation that minimizes the costs associated with delays and recovery actions for a given liner shipping route.

1.1 Literature review

Notteboom (2006) analyzes the effects of delays on liner shipping networks. The author provides insights in the causes of delay and mentions some actions that can be taken to recover from the obtained delays. One of the proposed recovery actions is to increase the sailing speed on the sea leg. Notteboom and Vernimmen (2009) discuss the effect of high fuel costs on liner shipping networks. They conclude that high bunker prices have a significant influence on the route design in liner shipping networks. For example, liner shipping companies decided to increase their fleet size in order to be able to reduce the sailing speed on the route network. This also results in more buffer times in ports, however the authors do not study the influence of the distribution of the buffer times over the ports.

In the last period, research on minimizing fuel costs and emissions by decreasing the sailing speed has become more popular (see for example Corbet et al. 2009, Fagerholt et al. 2010 and Cariou 2011). From these studies it follows that the sailing speed of ships can have significant effect on fuel costs and emissions. However, these studies do not consider situations in which ships can encounter delays.

Wang and Meng (2012a) and Wang and Meng (2012b) consider the problem of designing liner shipping schedules when there is uncertainty in the port and sailing times. In their models, they try to adjust the speed in order to capture the delays. A trade-off is made between the round tour duration (number of ships needed) and fuel cost associated with the required speed to maintain these round tours. Qi and Song (2012) propose a model to minimize the fuel emissions when port times are uncertain. Again, delays are captured by adjusting the sailing speed on the next sea leg. The authors compare the on-time arrivals under different conditions. Furthermore, the authors provide some managerial insights regarding the uncertain port times. In all these articles, the

effect of the buffer times distribution on the uncertainty and fuel costs is not considered yet.

Brouer et al. (2012) consider a problem with disruption scenarios. They introduce a time horizon after which the vessel has to be back on schedule. Ships can adjust their sailing speed, omit port calls or swap the order of port calls in order to return to the planned time schedule. The model is solved within 10 seconds using a mixed integer programming solver for a standard problem. However, for larger problems, the computational time can exceed the 10 seconds.

Finally the problem of optimal allocation of buffer times also obtained attention in the railway sector, where it was framed as running time supplements. Kroon et al. (2007) and Kroon et al. (2008) use a stochastic optimization model to solve the problem of optimally allocating running time supplements in a railway network. Kroon et al. (2007) consider the problem of a single train for which they found that in an optimal allocation supplements are mainly allocated to the middle part of the trip and not much to the first and last parts of the trip. Kroon et al. (2008) extended this research for multiple trains and a cyclic timetable. Both studies show that the punctuality (percentage of trains that arrive within 3 minutes of the schedule time) can be increased by reallocating the time supplements. Since trains usually already run at maximum speed and dwell times in a station are needed to let passengers enter and leave the train, it is not possible to include recovery actions in a railway network.

1.2 Contribution

Very little research has been performed on developing robust routes in liner shipping. In this current research, delays can only be reduced by increasing the sailing speed on the next sea leg. Furthermore, it is often assumed that the sailing speed can be adjusted in such a way that the total amount of delay can be captured on this single leg. However, since very high costs are incurred when the sailing speed has to be increased to the maximum, it is probably cheaper to use multiple sea legs to capture large delays (especially when the next sea leg is relatively small). Furthermore, the effect of the buffer time distribution on delays is not studied yet.

In this article, we propose a model that can be used to find the optimal recovery policy and buffer time distribution at the same time. To the best of our knowledge, such a formulation does not exist yet in the literature. The recovery policy can include both actions in ports and actions between port. Furthermore, it is also possible to include actions that will take more time, such as decreasing the sailing speed, in order to minimize the cost of the liner shipping company.

2 Problem description

2.1 Ship delay recovery problem

In the ship delay recovery problem, the ship schedule and vessels operating the schedule are known on forehand. Thus, we know for every port on the route when a ship is to arrive in this port and when it should leave the port again. Due to unexpected events, ships can encounter delay both when they are in a port and when they are sailing between ports. Delays of ships have consequences on for example port terminal planning and customer satisfaction. Hence, liner shipping companies incur costs when ship arrive delayed in ports. Therefore, the objective of the liner shipping company is to reduce the cost associated to delay as much as possible. When ships are delayed, they can reduce their delay by taking recovery actions against certain costs.

The goal of the ship delay recovery problem is to determine a recovery policy that minimizes the costs associated with delays and recovery actions for a given liner shipping route schedule. Thus, the buffer time allocation is known on forehand in the ship delay recovery problem.

2.1.1 Markov decision process

The ship delay recovery problem will be formulated using a Markov decision process. The recovery policy can then be defined as a policy that describes for every possible state of the Markov process which action has to be performed given that the process is in that specific state. The states of the Markov process denote the position of the ship and the amount of delay with respect to the original schedule encountered by the ship. The position of the ship contains both the port in which the ship is berthing and whether it is arriving or leaving the port. Furthermore,

since a finite number of possible states are needed in the Markov process, we discretize delay and fix a maximum amount of delay that can be encountered by a ship.

Next, the probability of a transition from one state to another can be determined for all combinations of states. The definition of the states allows to make use of the Markov property, which says that the transition probabilities of one state to another are independent on the past states of the process. In other words, the transition probability of the current state to any other state only depends on the current delay. Furthermore, in each state of the Markov process a decision is made about which recovery action to take in that state. Thus, the transition probabilities also depend on the chosen recovery actions. This decision process is referred to as a Markov decision process. Now, we introduce the following sets:

- \mathcal{P} : set of possible positions of a ship (port name and arriving/leaving).
- \mathcal{D} : set of possible units of delay.
- \mathcal{I} : set of possible states of the Markov process, $\mathcal{I} = \mathcal{P} \times \mathcal{D}$.
- \mathcal{K} : set of possible actions, which can be performed when in a specific state.

A Markov decision process can be defined as a discrete time Markov chain in which after each transition an action $k \in \mathcal{K}$ has to be chosen from a set of available actions (Hillier and Lieberman 2001). A cost C_{ik} is associated to performing recovery action $k \in \mathcal{K}$ in state $i \in \mathcal{I}$. As already mentioned, the probability of a transition from state $i \in \mathcal{I}$ to state $j \in \mathcal{I}$ only depends on the current state i and the recovery action $k \in \mathcal{K}$ chosen in state i . These transition probabilities are denoted by p_{ijk} .

Usually, the (long-run) expected average cost per unit time is used as a performance measure in Markov decision processes. The underlying assumption in calculating the expected average cost per time unit is that a transition between states takes always place after a fixed time interval. However, in this model the order in which the ports are visited is fixed, while the time needed in a state varies over the states. Since costs are only incurred once in each state, the (long-run)

average cost per ship position can be computed as

$$\mathbb{E}[C] = \sum_{i \in \mathcal{I}, k \in \mathcal{K}} \pi_i C_{ik} D_{ik},$$

where D_{ik} denotes the current policy, $D_{ik} = 1$ if action k is chosen in state k and 0 otherwise, and π_i represents the steady-state probability of being in state i under the evaluated policy.

The next section describes a linear programming formulation that can be used to solve a Markov decision process. The linear programming formulation is obtained from Hillier and Lieberman (2001). They also describe a few iterative algorithms that can be used to solve a Markov decision process, such as policy improvement.

2.1.2 Linear programming formulation

The Markov decision process can be formulated as a linear programming model if randomized policies are allowed. In a randomized policy the D_{ik} variables are allowed to be continuous variables instead of integer variables. The definition of D_{ik} then becomes

$$D_{ik} = \mathbb{P}(\text{decision} = k | \text{state} = i).$$

In other words, D_{ik} is the probability that action k is selected given that the process is in state i . The decision variables of the linear programming model are

$$\pi_{ik} = \mathbb{P}(\text{state} = i \text{ and action} = k).$$

Thus, π_{ik} is the steady-state unconditional probability that the process is in state i and decision k is made (Hillier and Lieberman 2001). Note that π_{ik} and D_{ik} are closely related to each other. It follows that

$$\mathbb{P}(\text{state} = i \text{ and action} = k) = \mathbb{P}(\text{action} = k | \text{state} = i) \mathbb{P}(\text{state} = i),$$

or

$$\pi_{ik} = D_{ik} \pi_i,$$

with π_i the steady-state probability that the Markov chain is in state i :

$$\pi_i = \sum_{k \in \mathcal{K}} \pi_{ik}.$$

From the previous formulas it can be concluded that

$$D_{ik} = \frac{\pi_{ik}}{\pi_i} = \frac{\pi_{ik}}{\sum_{l \in \mathcal{K}} \pi_{il}}.$$

The linear programming model can now be formulated as:

$$\min \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} C_{ik} \pi_{ik} \tag{1}$$

$$\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} \pi_{ik} = 1. \tag{2}$$

$$\sum_{k \in \mathcal{K}} \pi_{jk} - \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} \pi_{ik} p_{ijk} = 0 \quad j \in \mathcal{I}. \tag{3}$$

$$\pi_{ik} \geq 0 \quad i \in \mathcal{I}, \quad k \in \mathcal{K}. \tag{4}$$

The objective of the model (1) is to minimize the average expected cost, Constraint (2) ensures that the probabilities of being in a state sum up to 1 and Constraints (3) are the transition constraints. Finally, Constraints (4) make sure that all π_{ik} take nonnegative values.

After solving the linear programming model, the values of D_{ik} can be obtained using the values of π_{ik} :

$$D_{ik} = \frac{\pi_{ik}}{\sum_{l \in \mathcal{K}} \pi_{il}}.$$

The optimal solution of the model will contain $|\mathcal{I}|$ basic variables $\pi_{ik} > 0$. Since it can be shown that $\pi_{ik} > 0$ for at least one $k \in \mathcal{K}$ for each $i \in \mathcal{I}$, it follows that $\pi_{ik} > 0$ for only one $k \in \mathcal{K}$ for each $i \in \mathcal{I}$. Therefore, each D_{ik} has an integer value and the optimal policy is deterministic instead of randomized (Hillier and Lieberman 2001).

2.2 Ship delay recovery problem with variable buffer times

Next, we consider the ship delay recovery problem in which the buffer times are variable instead of fixed. In this problem, the ship route and vessels operating the route are known on forehand. Thus, now we only know which ports are visited and in which order they are visited by the ships. Based on this information, we can determine the minimum time needed between an arrival and departure in the same port and between a departure in one port and an arrival in the next port. However, since liner shipping routes are usually serviced once a week, the round tour time is rounded to an integer number of weeks. This means that there is probably some time left when we only use the minimum times needed in the schedule. We will call this additional time to guarantee a round tour time of an integer number of weeks the total amount of buffer time in the model.

In the ship delay recovery problem with variable buffer time distribution, we still want to determine an optimal recovery policy. Hence, an additional decision regarding the buffer time is included in the problem. We want to determine for each port which amount of buffer can best be included in the time schedule to minimize the costs associated with delays and recovery actions. Note that contrary to the recovery policy, the buffer time should be determined on forehand and should be independent on the amount of delay incurred in a port. Thus, buffer times can also be modeled as an action in a Markov decision process. However, we lose the property that actions in different states (that is, different amount of delay) can be taken independent of each other. Although policy improvement can still be applied, it does not necessarily result in improvements in each step and hence it does not necessarily result in the optimal policy. In general it is unknown how to address such a problem in Markov decision chains. We were able to formulate a mixed integer programming model by adding additional (integer) constraints to the linear programming formulation of the Markov decision process.

2.2.1 Mixed integer formulation

First, we introduce the following sets needed to formulate the new model:

- \mathcal{B} : set of possible values of buffer time per ship position.

- \mathcal{A} : set of possible actions in the new Markov decision problem, $\mathcal{A} = \mathcal{K} \times \mathcal{B}$.

The ship delay recovery problem with variable buffer times can now be formulated as:

$$\min \sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}} C_{ia} \pi_{ia} \quad (5)$$

$$\sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}} \pi_{ia} = 1. \quad (6)$$

$$\sum_{a \in \mathcal{A}} \pi_{ja} - \sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}} \pi_{ia} p_{ija} = 0 \quad j \in \mathcal{I}. \quad (7)$$

$$\sum_{p \in \mathcal{P}} \sum_{b \in \mathcal{B}} B_b y_{pb} \leq M. \quad (8)$$

$$\sum_{b \in \mathcal{B}} y_{pb} = 1 \quad p \in \mathcal{P}. \quad (9)$$

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}} \pi_{(pd), (kb)} \leq y_{pb} \quad p \in \mathcal{P}, \quad b \in \mathcal{B}. \quad (10)$$

$$\pi_{ia} \geq 0 \quad i \in \mathcal{I}, \quad k \in \mathcal{K}. \quad (11)$$

$$y_{pb} \in \{0, 1\} \quad p \in \mathcal{P}, \quad b \in \mathcal{B}. \quad (12)$$

The objective (5) of the model is still to minimize the average expected cost and Constraints (6) and (7) are also still the same as in the ship delay recovery problem. However, using only these constraints will result in multiple buffer times for a ship position, since for each value of delay another buffer time can be chosen. Therefore, additional constraints have to be added that ensure that the same buffer time is chosen for all different values of delay for each ship position. Thereto, the binary decision variables y_{pb} are introduced, which take the value 1 if buffer time $b \in \mathcal{B}$ is allocated to ship position $p \in \mathcal{P}$ and 0 otherwise. Constraints (8) make sure that the total allocated buffer time may not exceed the maximum amount of buffer time available (M). Furthermore, exactly one possible buffer time has to be allocated to each ship position, which is ensured by Constraints (9). Finally, Constraints (10) states that the steady-state probabilities π_{ia} can only be positive for combinations of ship position and buffer times for which $y_{pb} = 1$. The right-hand side of these constraints can be strengthened to $y_{pb}/|\mathcal{P}|$ if ports are always visited in the same order (i.e. actions that skip ports are not considered), since ships will then be with equal probability in each of the ports.

Note that the above mixed integer programming model simplifies to the linear programming model discussed in the previous section when the buffer time is fixed. For a feasible fixed buffer time distribution constraints (8) and (9) always hold. Furthermore, only states for which $\xi_{pb} = 1$ can be visited in this case, which ensures that also constraints (10) are automatically satisfied when a feasible fixed buffer time distribution is given.

In the mixed integer programming (MIP) model all feasible buffer time distributions can be formed using the ξ_{pb} variables. Since for a fixed ξ_{pb} the mixed integer programming model simplifies to the linear programming model discussed in the previous section, the mixed integer programming model minimizes the costs over all feasible buffer time distributions. Thus, the mixed integer programming model will indeed return the buffer time distribution that results into the lowest cost of delay and recovery actions. Furthermore, the optimal recovery policy for this buffer time distribution is returned by the model.

3 Structural results

Before we introduce two heuristics to solve the ship delay recovery problem with variable buffer times, we prove the correctness of two theorems about the structure of solutions.

Theorem 1

1. *The total cost incurred in the ship delay recovery problem with variable buffer times is non-increasing in the total amount of buffer time that can be allocated.*
2. *If the total amount of buffer time has to be allocated, the total cost incurred in the ship delay recovery problem is non-increasing in this total amount.*

First, we proof part 1 of Theorem 1. Let (π_B^*, y_B^*) be an optimal solution to the problem with B units of buffer. Let z_B^* be the costs corresponding to this solution. Then (π_B^*, y_B^*) is also a feasible solution to the problem with $B + 1$ units of buffer. Thus, it holds that

$$z_B^* \leq z_{B+1}^*,$$

where z_{B+1}^* is the cost corresponding to the optimal solution of the problem with $B + 1$ units of buffer.

Now, we proof part 2 of Theorem 1. Let (π_B^*, y_B^*) be an optimal solution to the problem with B units of buffer. Let z_B^* be the costs corresponding to this solution and let k_B^* denote the optimal actions taken in each of the states. Now, we can randomly select an initial state and construct a sample path s_B by randomly selecting a delay for each port position. The costs of s_B can now be calculated and compared to the costs of a sample path s_{B+1} that has the same initial state, actions and delays. The buffer time distribution of s_{B+1} is also the equal to that of s_B , except for one port p for which the buffer time is one unit higher than it is for s_B . Since the sample paths s_B and s_{B+1} are obtained using the same actions, delays and buffer (except for port p), the costs of s_B and s_{B+1} are equal until the ship arrives in port p . If the ship arrives according to s_B with a delay of d units in port p ($d > 0$), the ship will have $d - 1$ units of delay in this port according to sample path s_{B+1} . Thus, in sample path s_B a cost of $C_{(p,d),k}$ is incurred, while for sample path s_{B+1} a cost of $C_{(p,d-1),k}$ is incurred. Since the costs are monotonically increasing in the amount of delay, we know that $C_{(p,d-1),k} \leq C_{(p,d),k}$. If, on the other hand, the ship arrives without delay in port p , then the ship will also have no delay in port p according to sample path s_{B+1} , so the costs of both sample paths are equal in this case. Now, for the remaining ports, the delay of sample path s_B will be larger than or equal to the delay of sample path s_{B+1} . Since, the costs are non-decreasing in the amount of delay for all port positions, this means that the total costs of sample path s_{B+1} will be at most as high as the total costs of sample path s_B . Thus, we have found a feasible solution to the problem with $B + 1$ units of buffer time in which all buffer time is allocated and the total costs of a randomly generated sample path are at most as high as the total costs of the corresponding sample path associated with the optimal solution of the problem with B units of buffer time. Since this holds for every sample path, the total costs of the problem with $B + 1$ units of buffer time (z_{B+1}) will be at most as high as the optimal costs of the problem with B units of buffer. Thus, $z^{B+1} \leq z_B^*$ and the total costs are thus monotonically decreasing when the amount of buffer time is increasing.

For the next theorem we need to make some assumptions.

Assumption 1 *The cost function is convex in the amount of delay.*

Assumption 2 *The recovery policy is the same for every buffer time distribution.*

Theorem 2 *If Assumptions 1 and 2 are satisfied, then the reduction in total cost is monotonically decreasing in the amount of buffer time that can be allocated.*

We consider a situation in which we have a fixed buffer time distribution of B units, a fixed recovery policy and three different sample paths s_B , s_{B+1} and s_{B+2} . The first sample path, s_B , is generated as follows. First, we randomly select an initial state and then we construct a sample path by randomly selecting a delay for each ship position and a certain recovery policy k_B . The second and third sample path, s_{B+1} and s_{B+2} , are generated using the same recovery policy and delays as in s_B . Only the buffer time distribution differs over the three sample paths: in path s_{B+1} the buffer distribution is the same as that of sample path s_B with the exception of one port for which the buffer time is one unit higher than in s_B and in path s_{B+2} the buffer distribution is the same as that of path s_{B+1} with the exception of one port for which the buffer time is one unit higher than in s_{B+1} . Let the delay in a ship position p of sample path s_B be denoted by D_p^B . Similarly, D_p^{B+1} and D_p^{B+2} are the delays in port position p of sample paths s_{B+1} and s_{B+2} respectively. Since the sample paths s_B and s_{B+1} are obtained using the same actions, delays and buffer (except for port p^*), the costs of s_B and s_{B+1} are equal until the ship arrives in port p^* . If the ship arrives according to s_B with a delay of d units in port p^* ($d > 0$), the ship will have $d - 1$ units of delay in this port according to sample path s_{B+1} . A similar reasoning can be given for sample paths s_{B+1} and s_{B+2} . Thus, we know that both $D_p^{B+2} - D_p^{B+1}$ and $D_p^{B+1} - D_p^B$ are either equal to 0 or to 1. Furthermore, we know that if $D_p^{B+1} - D_p^B = 0$ then also $D_p^{B+2} - D_p^{B+1} = 0$, because the delay in port position p will only be the same if the ship encounters no delay in this port position. Thus, by increasing the buffer time with one unit, we either find only a cost reduction for adding the first unit of buffer (in which case the cost reduction of adding the first unit of buffer is clearly larger than the cost reduction of adding the second unit of buffer) or we find both a cost reduction for adding the first unit of buffer and for adding the second unit of buffer time. In the last case, the cost reduction of adding the first unit of buffer is equal to $C_{(p,d),k} - C_{(p,d-1),k}$ and the cost reduction of adding the second unit of buffer is equal to $C_{(p,d-1),k} - C_{(p,d-2),k}$. Since the cost function is convex in the amount of delay,

we know that $C_{(p,d),k} - C_{(p,d-1),k} \geq C_{(p,d-1),k} - C_{(p,d-2),k}$. Thus, the cost reduction of adding the first unit of buffer is larger than that of adding the second unit of buffer. Again, since this holds for all port positions and sample paths, we have proven that the cost reduction of adding consecutively units of buffer to an existing buffer time distribution when the recovery policy is fixed, is monotonically decreasing in the amount of buffer that is added.

4 Solution methodology

4.1 Greedy Buffer Heuristic

The idea of the greedy buffer heuristic (GBH) is to start without any buffer time and step by step allocate units of buffer until the maximum amount of buffer is reached. In each iteration of the greedy buffer heuristic, the cost reduction of increasing the buffer time with one unit in a certain port compared to the current buffer distribution is calculated for each port, where the buffer time for all other ports remains the same as in the current distribution. Since the problem reduced to a Markov decision process when the buffer time distribution is fixed, $|\mathcal{P}|$ Markov decision processes have to be solved in each iteration. The algorithm can be summarized as follows:

<i>Initialization:</i>	Each ship position has 0 units of buffer, $n = 0$.
<i>Iteration n:</i>	
<i>Step 1: Determine savings:</i>	Determine for each ship position the cost reduction when one unit of additional buffer is added to this position.
<i>Step 2: Buffer allocation:</i>	Allocate the $n + 1$ -th unit of buffer to the ship position that correspond to the highest cost reduction.
<i>Stopping criterion:</i>	Repeat until $n = M$.

Notice that it follows from Theorems 1.2 and 2 that the greedy buffer heuristic would return the optimal solution if the cost reduction of adding one unit of buffer time to a port is independent on the allocated buffer times to other ports. In this case, the cost reduction of adding one additional unit of buffer would only change for the port to which the last unit of buffer is allocated. From Theorem 2 it then follows that this new cost reduction will be at most as high as the previous one. Thus, the heuristic select the M best positions to allocated the buffer time to. However, since the cost reduction of adding one unit of buffer is in reality dependent on the allocated

buffer times to other ports, the greedy buffer heuristic will not be guaranteed optimal in reality.

4.2 Buffer Exchange Heuristic

In the buffer exchange heuristic (BEH), we start with a feasible buffer distribution and exchange in each iteration units of buffer between two ports in order to obtain a better buffer distribution. The heuristic starts with exchanging as much buffer as possible between the two ports for which the change is most profitable. In this way, the solution space is quickly explored. When no profitable changes can be made anymore the amount of buffer to be exchanged in each iteration is decreased to find the best solution in the direct neighborhood of the current solution. The following algorithm gives a description of the buffer exchange heuristic:

<i>Initialization:</i>	Start with initial feasible buffer allocation and set b equal to 2^k for $k \in \mathbb{N}$ such that $2^k \leq b_{max}$ and $2^{k+1} > b_{max}$, with b_{max} the largest amount of buffer in one of the ports.
<i>Iteration n:</i>	
<i>Step 1: Determine savings:</i>	For each combination of two port exchange b units of buffer and compute new total costs.
<i>Step 2: Buffer exchange:</i>	If there is a profitable exchange, make the most profitable one. Otherwise set $b = b/2$.
<i>Stopping criterion:</i>	Repeat until $b < 1$.

5 Computational Experiments

5.1 Data

To test our methods, we use the ME1 route in September 2012 of the Maersk Line network (Maersk Line (2012)), which we have slightly adapted in order to obtain valid input for our model. First of all, we have to discretize the time in order to obtain a finite number of states in our model. We decided to discretize time in time units of 4 hours with a maximum delay of 25 time units. Table 1 shows the order in which the ports are visited in the route, the distances and sailing times between ports and the time needed in the port. Furthermore, the current buffer time distribution (in multiples of 4 hours) of the route is given, which we obtained by assuming that the sailing speed between each two ports is equal to 17 knots (1 knot = 1.85 km/hr). In the second column of table 1 the distances between the ports in nautical miles are presented. This

column has to be read in the following way. The distance shown for each port is the distance that the ship has to cover to reach that port. For example, the distance between Algeciras and Felixstowe can be found in the row of Felixstowe and is equal to 1274 nautical miles. The third column shows the expected sailing time in hours needed to reach that port using the 17 knot speed. The expected sailing time for Antwerp is 7 hours. This means that a ship needs 7 hours to sail from Felixstowe to Antwerp, when it does not encounter a delay during their trip. The expected sailing times can be computed by dividing the distance between two ports by the sailing speed. The fourth column denotes the total time needed in the port to load and unload the ship. The last column denotes the buffer times for the ports. These buffer times are subtracted from the encountered delay to determine the delay with respect to the time schedule when the ship arrives in the port. The time needed to make one full round tour is 1176 hours (7 weeks). The total bunker cost of one round tour are \$2.6 million for a ship that follows the published schedule.

Port	Distance (nm)	Sailing Time (hr)	Port Time (hr)	Buffer Time (hr)
Jebel Ali	2186	109	31	16
Jawaharlal Nehru	1132	71	33	12
Mundra	371	21	16	4
Salalah	958	49	14	8
Jeddah	1278	64	11	4
Suez Canal	636	34	16	0
Algeciras	2002	99	18	0
Felixstowe	1274	65	24	24
Antwerp	136	7	16	8
Bremerhaven	315	16	24	16
Rotterdam	246	13	20	12
Suez Canal	3361	168	22	24
Aqaba	300	16	20	4
Jeddah	560	28	19	12

Table 1: Characteristics of the route

In the experiments we only consider the possibility of taking recovery actions when sailing between ports. However, it is also possible to include recovery actions in ports in the model. We include both the possibility to increase speed and to decrease speed. For each sea leg we include an action to decrease the sailing speed in such a way that the needed time to sail the leg will be increased by one and two time units. Furthermore, we include actions to reduce the sailing time on a sea leg by one to five time units. For some sea legs it is not possible to include all actions,

because the sailing speed should be unrealistic high to obtain these gains. In these cases, we only consider the actions for which it is still possible to sail the leg in the corresponding time without exceeding the maximum speed of 30 knots. The costs of performing the recovery actions are given in Table 2. The costs are in dollars and are computed using the formula obtained from Alderton (2005):

$$C_f(v) = f_d^s \cdot p_{bunker} \left(\frac{v}{v_d^s} \right)^3 \quad (13)$$

in which v denotes the speed in knots, v_d^s the design speed of ship s , f_d^s the fuel consumption in ton per day of ship s , p_{bunker} the bunker price in US dollars per ton at design speed and $C_f(v)$ the fuel consumption in US dollars per day at a speed of v knots. We use a bunker price per ton of \$650, a design speed of 20.75 knots and a fuel consumption of 149.3 ton per day. These values are based on data provided by a liner shipping company.

Using Formula 13 and the scheduled speed of each sea leg, which can be obtained from Table 1, we can determine the fuel cost for each sea leg when the ship sails according to schedule:

$$C_f^i(a) = f_d^s \cdot p_{bunker} \left(\frac{d_i}{(t_i - g(a)) \cdot v_d^s} \right)^3, \quad (14)$$

where $C_f^i(a)$ is the fuel cost in US dollars on sea leg i under recovery action a , d_i the distance to be covered on sea leg i , t_i the scheduled sailing time for leg i , $g(a)$ is the gain of recovery action a in hours and f_d^s , p_{bunker} and v_d^s as defined above. Now,

$$C_a = C_f^i(a) - C_f^i(0) \quad (15)$$

gives the cost of the recovery action a .

In section 5.2, the performance of the route schedule with optimized buffer times is determined using a uniform distribution of additional delay for both the sea legs and the port stays. We assume that both in ports and on sea legs a maximum amount of 3 time units of additional delay can be incurred. The results of this optimized schedule are compared to the current schedule. In section 6, the performance of the greedy buffer and buffer exchange algorithms are compared to the performance of the mixed-integer linear programming model. In this section, 50 test cases

Port	Gain of recovery action (in thousands of \$ per sea leg)								
	-2	-1	0	1	2	3	4	5	
Jebel Ali	-52.6	-27.7	0	30.9	65.6	104.5	148.7	198.9	
Jawaharlal Nehru	-43.7	-23.9	0	29.3	65.9	112.1	171.8	250.8	
Mundra	-24.9	-15.4	0	27.6	84.3	-	-	-	
Salalah	-43.3	-24.1	0	30.8	71.0	124.9	199.7	-	
Jeddah	-48.4	-26.3	0	31.8	70.6	118.7	179.4	257.3	
Suez Canal	-34.7	-20.1	0	28.7	71.5	139.8	-	-	
Algeciras	-53.3	-28.2	0	31.9	68.0	109.3	156.6	211.4	
Felixstowe	-45.9	-24.9	0	30.0	66.5	111.7	168.3	240.7	
Antwerp	-18.2	-13.8	0	-	-	-	-	-	
Bremerhaven	-30.7	-19.9	0	43.0	-	-	-	-	
Rotterdam	-24.6	-16.6	0	43.3	-	-	-	-	
Suez Canal	-54.1	-28.0	0	30.1	62.4	97.3	134.9	175.7	
Aqaba	-26.5	-17.2	0	37.1	-	-	-	-	
Jeddah	-40.1	-23.8	0	36.6	97.3	-	-	-	

Table 2: Cost of performing recovery actions

are generated. For each port stay and sea leg, a unimodal probability distribution of incurring additional delay is generated at random. The maximum amount of additional delay has to be between a lower and an upper bound for each port stay and sea leg.

Finally, the cost per time unit of delay is assumed to be \$40,000 per time unit (\$10,000 per hour). Since it is also possible to decrease the sailing speed and thereby decrease the costs, the cost of one time unit of delay should be at least as high as the gain of decreasing the sailing time by one time unit. Otherwise, it would always be profitable to increase the delay by reducing the sailing speed, which is not desirable.

The (mixed integer) linear programming models are solved by CPLEX 12.4 running on a 3.0 GHz Dual Core PC with 4 GB of RAM. The running time of the mixed integer programming model is limited to 900 seconds.

5.2 Results optimized buffer time distribution

Table 3 shows the average costs in dollar per port and the computational time in seconds for the linear programming model to solve the model for the current buffer time distribution and the

mixed integer programming model and the two heuristics for the optimized buffer distribution. For this instance, all three methods used to improve the buffer times return the same (optimal) buffer distribution and recovery policy. The greedy buffer heuristic needs the lowest computational time of these methods, while the buffer exchange heuristic needs even more computational time than solving the mixed integer programming model for this instance. The average costs per port are decreased by 14.8% compared to the current buffer time distribution. The reduction in average delay costs (16.8%) and in average recovery costs (13.9%) contribute almost equally to the total reduction.

Method	Average costs (in thousands of \$)			Time (s)
	Recovery	Delay	Total	
Current	43.9	20.4	64.3	0.4
MIP	37.8	17.0	54.7	137
GBH	37.8	17.0	54.7	62
BEH	37.8	17.0	54.7	202

Table 3: Average cost per port and computational times for the different methods

The average delay per ship position and buffer time distribution for both the current and the optimized schedules are given in Table 4. Since the gain of both the buffer time and the recovery actions are incurred before the arrival in a port, the average delay is lower for arriving states than for departure states. Furthermore, it can be seen from the table that the average delay is more evenly distributed for the optimized schedule than for the current schedule. In the current schedule a lot of buffer time is allocated to the port of Felixstowe and to the Suez Canal passage in the eastbound direction. This results in very low average arrival delay in these states (0.88 hours and 0.32 hours respectively). This is an indication that a part of the buffer time is unused in many round tours of the route. Therefore, it is more efficient to reduce the allocated buffer times in these ports and allocate the remaining hours to other ports. In the optimized schedule, we see indeed that the amount of buffer is reduced in these ports. Finally, the punctuality, which is defined as the fraction of time that a ship arrives in a port with a delay less than one time unit (4 hours), is 65.8% for the current schedule and 68.1% for the optimized schedule.

Table 5 shows the optimal recovery policy for the schedule with optimized buffer times. The

Port	Current			Optimized		
	Average delay Arrival (hr)	Departure (hr)	Buffer time (hr)	Average delay Arrival (hr)	Departure (hr)	Buffer time (hr)
Jebel Ali	1.24	7.24	16	2.04	8.04	12
Jawaharlal Nehru	1.00	7.00	12	1.24	7.24	8
Mundra	1.64	7.72	4	1.64	7.64	8
Salalah	2.60	8.76	8	1.60	7.60	12
Jeddah	2.52	8.68	4	2.28	8.28	8
Suez Canal	2.48	8.64	0	1.28	7.28	12
Algeciras	6.76	13.80	0	2.40	8.40	8
Felixstowe	0.88	6.88	24	1.80	7.80	8
Antwerp	1.60	7.68	8	1.72	7.72	8
Bremerhaven	1.68	7.68	16	1.72	7.72	16
Rotterdam	2.68	8.88	12	1.00	7.00	16
Suez Canal	0.32	6.32	24	2.16	8.16	12
Aqaba	1.00	7.00	4	1.12	7.12	4
Jeddah	2.12	8.12	12	1.68	7.68	12

Table 4: Average delay and buffer time distribution for the current and optimized route schedule

number in the table corresponds to the gain of the selected recovery action. Thus, when a ship for example departs from Mundra with a delay of 3 time units, it will increase its speed in such a way that 1 time unit of gain is obtained while sailing to Salalah. From the table, we see that it is sometimes possible to reduce the speed of the ship when the departure is on time. The ports in which the ship arrives after a sea leg for which the speed can be decreased all have large buffer times (12 hours or more). Indeed, we see a relation between the level of the recovery action and the amount of buffer time in the next port. For example, on the sea leg from Suez Canal to Aqaba ships always have to increase their speed compared to the scheduled speed (even when they leave Suez Canal without any delay). This is probably caused by the small buffer scheduled when arriving in Aqaba (only 4 hours). Furthermore, the recovery policy shows for each sea leg (multiple) delays for which the level of the recovery action should be increased.

6 Comparison of the methods

Table 6 shows some characteristics of the solutions of 50 instances of the route network. First of all, the number of proven optimal solutions is shown in the table. Because the time limit of the mixed integer programming model is set to 900 seconds, the MIP model will not always

Port	Delay (time units)									
	0	1	2	3	4	5	6	7	8	9
Jebel Ali	0	1	2	3	3	4	5	5	5	-
Jawaharlal Nehru	0	1	2	2	3	4	4	-	-	-
Mundra	0	0	1	1	2	2	-	-	-	-
Salalah	0	1	1	2	3	3	4	-	-	-
Jeddah	-1	0	1	2	2	3	4	-	-	-
Suez Canal	0	1	1	2	2	3	-	-	-	-
Algeciras	0	1	2	2	3	4	5	-	-	-
Felixtowe	0	1	2	2	3	4	-	-	-	-
Antwerp	-1	0	0	0	0	0	-	-	-	-
Bremerhaven	-1	0	0	1	1	1	1	1	-	-
Rotterdam	0	0	0	1	1	1	1	1	1	-
Suez Canal	1	2	3	4	5	5	5	5	5	5
Aqaba	0	0	1	1	1	1	1	1	-	-
Jeddah	-1	0	1	1	1	2	2	2	2	-

Table 5: Optimal recovery policy under the optimized schedule

return the optimal solution. Furthermore, if the MIP model does not return the guaranteed optimal solution, we do not have another method to obtain the optimal solution. Therefore, for all three methods, the number of proven optimal solutions correspond to instances in which the mixed integer programming model was solved to optimality within 900 seconds. This happened for 13 instances and for all these 13 instances both the greedy buffer and the buffer exchange heuristic also returned the optimal solution. Further, we determined both the average and the maximum difference with the best lower bound returned from the MIP model. These differences are about the same for the buffer exchange heuristic and the MIP model, which is confirmed by the average, maximum and minimum difference between the buffer exchange heuristic and the MIP solution (all three very close to 0%). The greedy buffer heuristic performs a bit less, with a maximum difference of 16.1% and 12.9% with the best lower bound and MIP solution respectively. However, on average the performance of the greedy buffer heuristic is less than 1% worse than the MIP solution.

In 11 out of the 50 instances the greedy buffer heuristic returned a solution that was worse than the MIP solution, while this only happened once for the buffer exchange heuristic. Furthermore, the buffer exchange heuristic performed 7 times better than the mixed integer programming

model against twice for the greedy buffer heuristic. However, the average computational time of the greedy buffer heuristic is about 3 times lower than that of the buffer exchange heuristic. The mixed integer programming model always needs the largest amount of computational time, while the greedy buffer heuristic is always the fastest method.

	MIP	GBH	BEH
Number of optimal solutions	13	13	13
Average difference with lower bound (%)	2.18	3.11	2.17
Maximum difference with lower bound (%)	8.84	16.11	8.75
Average difference with MIP solution (%)	-	0.88	-0.01
Maximum difference with MIP solution (%)	-	12.90	0.03
Minimum difference with MIP solution (%)	-	-0.17	-0.17
Number of times better than MIP solution	-	2	7
Number of times worse than MIP solution	-	11	1
Average computational time (s)	843	91	270
Number of times fastest	0	50	0

Table 6: Characteristics of the solutions of the different methods

7 Conclusion

In liner shipping networks ships encounter problems when they are delayed. Since companies buy time slots and space in ports, delays will result in additional costs. Some actions, such as increasing the sailing speed of the ship or increasing the port handling capacity, can decrease the future costs of delay.

We presented a Markov decision process that can be used to determine an optimal recovery policy for a given liner shipping route. The recovery policy specifies for each combination of ship position and delay which recovery action should be performed when a ship is in that specific combination. Since Markov decision processes can be solved using a linear programming model, optimal recovery policies for a given buffer time distribution can easily be determined.

Furthermore, we presented a mixed integer linear programming model to formulate the ship delay recovery problem with variable buffer times. To the best of our knowledge, such a formulation does not exist yet in the literature. For small problems, the mixed integer programming model

can be solved quite fast using CPLEX 12.4. However, for larger instances, the computational time increases exponentially. Therefore, we also presented two heuristical methods to solve the problem.

First, we constructed a data set based on a liner shipping route as used by Maersk Line in September 2012 with uniform probabilities of additional delay between 0 and 3 time units for each ship position. We included recovery actions that increase or decrease the sailing speed on a sea leg and used the fuel consumption and bunker oil price to determine the cost these actions. The route with optimal buffer time distribution and recovery policy resulted in a cost decrease of 14.8% compared to the current route with optimal recovery policy.

Thereafter, we generated 50 test instances to compare the results of the heuristics. Both heuristical methods perform good for the ship delay recovery model with variable buffer times. The buffer exchange heuristic performs a bit better than the greedy buffer heuristic with respect to total cost, while the greedy buffer heuristics has lower computational times. Since, the computational time of the buffer exchange heuristic depends on the initial buffer distribution, it might be a good idea to combine both heuristics. Then, we could first apply the greedy buffer heuristic to obtain fast a good initial solution, whereafter the buffer exchange heuristic can be applied to improve the solution. Furthermore, at the moment we have the opportunity to implement some of our methods at a liner shipping company.

Acknowledgements

The authors want to thank Erasmus Smart Port BV for providing financial support for their research.

References

P. Alderton. *Reeds Sea Transport Operations and ECONOMICS*. (Adlard Coles Nautical, London), 5th edition, 2005.

- B.D. Brouer, J. Dirksen, D. Pisinger, C.E.M. Plum, and B. Vaaben. The Vessel Schedule Recovery Problem (VSRP) - a MIP model for handling disruptions in liner shipping. *European Journal of Operational Research*, 2012. doi: <http://dx.doi.org/10.1016/j.ejor.2012.08.016>.
- P. Cariou. Is slow steaming a sustainable means of reducing CO₂ emissions from container shipping? *Transportation Research Part D*, 16(3):260–264, 2011.
- J. Corbet, H. Wang, and J. Winebrake. The effectiveness and costs of speed reductions on emissions from international shipping. *Transportation Research Part D*, 14(8):539–598, 2009.
- K. Fagerholt, G. Laporte, and I. Norstad. The effectiveness and costs of speed reductions on emissions from international shipping. *Journal of Operational Research Society*, 61:523–529, 2010.
- F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research*, pages 1053–1078. (McGraw-Hill Education, New York), 7th edition, 2001. ISBN 0-07-246121-7.
- L.G. Kroon, R. Dekker, and M.C.J.M. Vromans. Cyclic railway timetabling: a stochastic optimization approach. In *Algorithmic Methods for Railway Optimization*, pages 41–66. (Springer Berlin/Heidelberg), 2007. ISBN 978-3-540-74245-6.
- L.G. Kroon, G. Maróti, M. Retel Helmrich, M.J.C.M. Vromans, and R. Dekker. Stochastic improvement of cyclic railway timetables. *Transportation Research Part B*, 42(6):553–570, 2008.
- Maersk Line. Maersk line schedule by vessel: Maersk kimi, 2012. <https://www.maerskline.com/frameSet.jsp>, Retrieved September 2012.
- T.E. Notteboom. The time factor in liner shipping services. *Maritime Economics & Logistics*, 8(1):19–39, 2006.
- T.E. Notteboom and B. Vernimmen. The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography*, 17(5):325–337, 2009.
- X. Qi and D.P. Song. Minimizing fuel emissions by optimizing vessel schedules in liner shipping with uncertain port times. *Transportation Research Part E*, 48(4):863–880, 2012.
- S. Wang and Q. Meng. Robust schedule design for liner shipping services. *Transportation Research Part E*, 48(6):1093–1106, 2012a.
- S. Wang and Q. Meng. Liner ship route schedule design with sea contingency time and port time uncertainty. *Transportation Research Part B*, 46(5):615–633, 2012b.