Bachelor's thesis Econometrics & Operations Research

Determining the optimal recovery policy in liner shipping networks using a Markov decision model

Author:
Judith Mulder
303469

Supervisor:
Prof. dr. Ir. R. Dekker

Co-reader:
Dr. A.F. Gabor

Erasmus School of Economics, Erasmus University Rotterdam
15th July 2009

Table of contents

1 Introduction

In liner shipping networks ships follow a fixed route within a fixed time schedule (Ong (2006)). Shipping companies buy a certain time slot in terminals in ports for a specified number of berths (Notteboom (2006)). Because ports 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.

If a ship is delayed before it arrives in the port, the result will be extra costs. Three different costs can be distinguished. First, the terminal can encounter extra costs, for example because the berth schedule has to be changed. The containers, which 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, do 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 (`http://www.haugenconsulting.com/whatisdemurrage.asp`).

The problem arises, when a ship first encounters a delay. The 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, it will encounter additional costs, as is explained earlier. Next to this, the possibility that the ship will also have a delay in the next ports to visit is increasing. Furthermore, the delay can increase, for example caused by the waiting time. A few actions can be performed to avoid this problem: a company can for example decide to hire an extra crane in the port to fasten the loading and unloading process or it can choose to increase the ship speed when sailing to the next port. These actions will give additional costs, but they decrease the costs caused by the delay.

To formulate the problem, two questions are needed: 'Which recovery actions should be performed to prevent the delays to accumulate?' and 'How much buffer should be scheduled to cover the delays?'

This thesis will discuss how to decide which action should be made in case of a known position and delay of a ship. To decide this, a stochastic model is needed. A Markov decision model can be used as a stochastic model, but simulation can also be used. The purpose is to find the best actions, so the model should also be able to optimize a certain objective function. It is hard to optimize using simulation. A Markov decision model, on the other hand, can be transformed in a linear programming problem, which can well be used to optimize. Furthermore, some extensions of the model are investigated. The research question in this project can be formulated in the following way:

- What is the use of a Markov decision model in this problem?

To answer this research question, the following subquestions will be answered:

- Is it possible to model the above problem as a Markov decision process?

- Can the Markov decision model be solved efficiently?

- How much time is needed to solve the Markov decision model?
- What can be concluded from the results of the Markov decision model?
- Can the Markov decision model be improved?

To answer these questions, a linear programming problem based on a Markov decision model will be used. The linear programming problem will be implemented in the computer program Aimms®.

In the following section a review of the literature available on liner shipping networks is presented. The next section defines the problem. After defining the problem, a model is developed to solve the problem. First this model is performed with a small data set to check whether the model is valid. Thereafter, a larger example is implemented. Next, two extensions of the model are discussed. Finally, the research questions are answered and a conclusion is formulated.

2 Literature review

This chapter reviews an overview of literature on ship routing and ship scheduling. Next, articles on sources of delay are viewed. Research, which is already performed, will also be discussed and finally some research on robustness in train and airplane networks is discussed.

This thesis is one of the first articles, which contains a model to reduce the delays in liner shipping networks. Therefore, little research is available on delays in shipping networks. Notteboom (2006) and Gkanatsas (2005) provide a useful background on the issues related to delays, but they do not provide a model, which can be used to optimize delay costs. The research on train networks (Clausen et al. (2006) and Dekker et al. (2007)) is useful to get some insight, but in this thesis a Markov desicion model is assumed.

A very detailed description of research performed on ship scheduling and related problems in the period from 1983-1993 is addressed by Ronen (1993). The article describes many other articles written on ship scheduling. Furthermore, in this article the improvement in computers is discussed, which made it possible to get better solutions and which reduced the time needed to compute a solution. Further, it discusses the development in the realization of solutions.

Such a description is also available for the period from 1994-2004 by Christiansen et al. (2004). This article describes the major developments in the ship routing and ship scheduling problems in the mentioned time period. The writers conclude that little research has been performed on liner shipping, which is a bit strange, because liner shipping becomes more and more important.

Research is also performed on sources of delay. Notteboom (2006) gives an overview of sources of delay and the ratio in which they occur. The delays are divided in four different groups: terminal operations, port access, maritime passage and chance. The group terminal operations contains all delays, which arises in port, such as port/terminal congestions (unexpected waiting times before berthing or before loading/unloading) and port/terminal productivity below expectations. The group port access contains for example unexpected waiting times due to irregularities in pilotage or towage services. The group maritime passage contains for example delays caused by missing the Suez convoy. Finally, the group chance contains unexpected waiting times due to weather conditions and on route mechanical problems. The first group, terminal operations, causes most of the delay.

Gkanatsas (2005) uses simulation to analyze the robustness of liner shipping networks. Furthermore, the simulation is used to investigate the effect of recovery actions on the robustness. The writer considers four different scenarios. These scenarios are increasing of nominal speed, omitting a port, omitting a port under a decision rule and increasing the nominal speed under a decision rule. The difference between the scenarios with and without a decision rule can be explained in the following way. In the scenarios without a decision rule, the action chosen for a certain harbour is equal in every simulation run. In the scenarios with a decision rule, the action performed in a certain harbour may vary between simulation runs. The costs are much lower for the scenarios with decision rules than for the scenarios without, but the average delay is higher for the scenarios with decision rule. So a trade off between expected costs and average delay has to be defined.

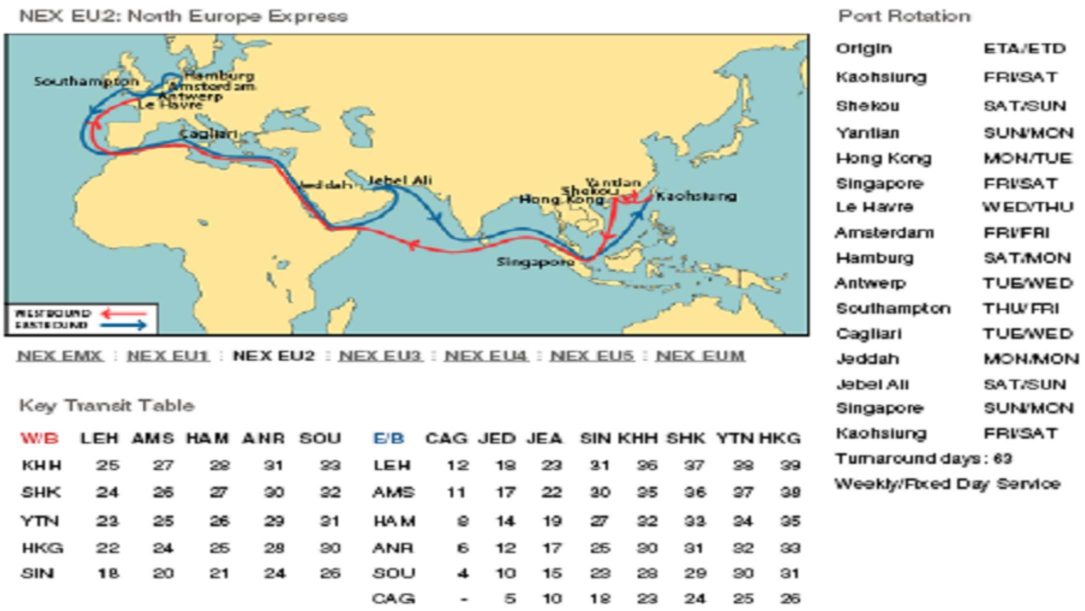
Similar research as above described is also performed in train and airplane networks. Clausen et al. (2006) use simulation to test the robustness of railway timetables. Furthermore simulation is used to investigate the effects of performing recovery actions on the robustness. The writers conclude that the best robustness is obtained with the recovery action that leads to the largest increase in space between trains. They also investigate the effect of buffer times in railway stations and conclude that an upper limit can be determined for which buffer time still leads to a positive effect.

Dekker et al. (2007) discuss the allocation of buffer time between trains and supplements in running times and dwell times. The writers try to minimize the total delay and try to implement their method for a cyclic timetable. They find that the optimal allocation of the buffer time looks like a reversed U, meaning that little buffer time is implemented at the first stations, where after the implemented buffer time increases to a maximum around half the tour and then decreases, so little buffer time is implemented at the end of the tour.

Goldsman et al. (2000) simulate the daily operations of an airport to create for example crew schedules or define recovery actions. This article also describes some sources of delay, which the plane can encounter. Furthermore some possible performance measures are discussed.

3 The ship delay recovery problem definition

In this thesis a liner container-shipping network is considered. An example of a ship string can be found in figure 1. This string is obtained from the website of the NYK Line: `http://www2.nykline.com/liner/service_network/eu2.html`. In a port, a schedule of the allocation of ships to berths is available. An example of a schedule for the berths is shown in figure 2. This schedule is obtained from the presentation of Dekker. Because companies know at which berth a ship is arriving, they can place the containers for the ship near that berth. When a ship is delayed, it is possible that it has to arrive at another berth than planned. The costs of transporting containers to the ship are then increased. Furthermore, the port is not the destination of the containers that are unloaded in a port. These containers have to be transported to their real destination. When a ship is delayed, it is possible that they cannot make it in time to the connecting transportation mode. In that case, the containers have to be placed in another transportation mode. Finally, a ship can encounter demurrage costs.


NEX EU2: North Europe Express

NEX EMX : NEX EU1 : NEX EU2 : NEX EU3 : NEX EU4 : NEX EU5 : NEX EUM

Port Rotation

Origin	ETA/ETD
Kaohsiung	FRI/SAT
Shekou	SAT/SUN
Yantian	SUN/MON
Hong Kong	MON/TUE
Singapore	FRI/SAT
Le Havre	WED/THU
Amsterdam	FRI/FRI
Hamburg	SAT/MON
Antwerp	TUE/WED
Southampton	THU/FRI
Cagliari	TUE/WED
Jeddah	MON/MON
Jebel Ali	SAT/SUN
Singapore	SUN/MON
Kaohsiung	FRI/SAT

Turnaround days : 63
Weekly/Fixed Day Service

Key Transit Table

W/B	LEH	AMS	HAM	ANR	SOU
KHH	25	27	28	31	33
SHK	24	26	27	30	32
YTN	23	25	26	29	31
HKG	22	24	25	28	30
SIN	18	20	21	24	26

E/B	CAG	JED	JEA	SIN	KHH	SHK	YTN	HKG
LEH	12	18	23	31	36	37	38	39
AMS	11	17	22	30	35	36	37	38
HAM	8	14	19	27	32	33	34	35
ANR	6	12	17	25	30	31	32	33
SOU	4	10	15	23	28	29	30	31
CAG	-	5	10	18	23	24	25	26

Figure 1: An example of a ship string, *source: website of NYK Line*

Costs are thus associated to delays. This can be used to find a model, which has the purpose to reduce the amount of delay. A problem with implementing costs in the model is that it is hard to estimate the amount of the costs, which are described above. The costs need to be increasing, because a reason is needed to perform a recovery action. When the costs are not increasing anymore for some delay, it is possible that ships do not perform recovery actions anymore, because it will cost more than it can reduce the costs.

It is necessarily for ships to perform recovery actions, because it prevents the delay to accumulate. This thesis discusses the determination of which recovery action should be made

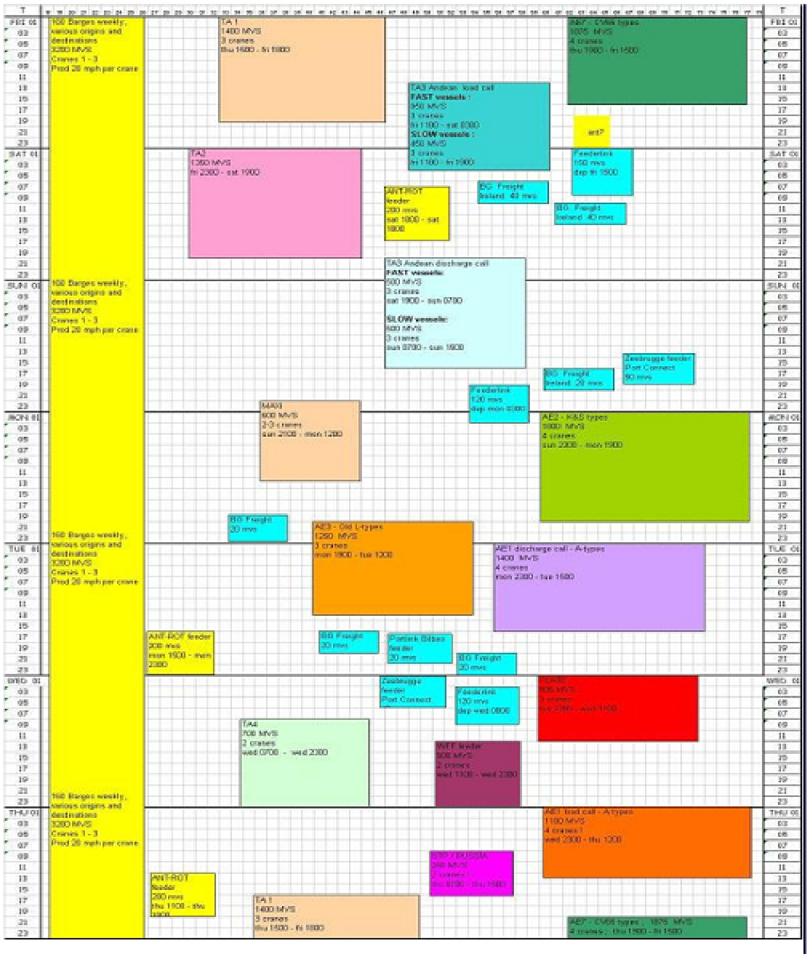


Figure 2: An example of a berth schedule, *source: presentation Dekker*

to minimize the total costs. To perform this, it is assumed that ships follow a certain time schedule (what they indeed do). This time schedule is the starting point. Next, probabilities of delay and costs of delay are determined. With these costs and probabilities, the total expected costs can be computed. These costs are only valid for the case in which no recovery action is performed. However, it is more interesting to try to decrease these costs by taking recovery actions at some place. By formulating this problem a Markov process is used. The objective is to find a policy, which describes for every possible state of the Markov process what action has to be performed, if the process is in that state.

3.1 Data

The following data are available to evaluate the problem. Most of the data used are the same as the data used in Gkanatsas (2005).

- The order in which ships are visiting the ports.
- The scheduled arrival and depart time for each port.

- The distributions of delays during the process of loading and unloading or during the trip from one port to the following one.
- The buffer time to capture (a part of) this delay.
- The set of possible actions, which can be performed to reduce the delay.
- The costs of delay dependent on the amount of delay.
- The costs of each recovery action.

Note that in each port the buffer time has to be at least equal to the expected delay corresponding to that port. When this is not the case, a delay over the round tour will be expected. Using the data of the buffer time and the distributions of the delay, the transition probabilities for each action can be calculated.

3.2 Decisions

The objective in this thesis is to obtain a policy, which contains the best action to perform for every possible state in which a ship can be. It is important to give a clear definition of a possible state. A state has to contain information about the position of the ship as well as the amount of delay of the ship. Because a finite number of possible states is needed, to make it possible to solve the problem for each possible state, the amount of delay is expressed as an integer number. The position of the ship can be expressed in two ways. First, only the harbour in which the ship is, can be denoted. Second, besides the harbour in which the ship is, also whether the ship is arriving or leaving the harbour can be denoted. When only actions in port or only actions out of port can be performed, the first way to express the position of a ship is satisfying. However, when actions can be performed both in port and out of port, the second way is the most convenient way to express the position. When only the harbour is expressed, two decisions should be made for each state. This is more difficult to implement than one decision for each state with twice as many states. Because in this thesis both actions in port and actions out of port can be performed, states define the harbour in which the ship is, whether the ship is arriving or leaving and the amount of delay a ship has.

Another important reason to define the Markov states as just explained, is because it allows making use of the Markov property. The Markov property states that the transition probabilities from one state to another are independent on the past states of the process. This is a very useful property, because it is assumed in this thesis that the probability of a certain delay in a port is independent on the amount of delay in previous ports. Similarly, the probability of a certain delay out of port is independent on previous delays out of port and also the probability of delay in port is independent on the delays out of port and vice versa.

In this thesis ships are considered, which are sailing round tours without a resting point. In train networks it is common that trains do have such a resting point. This station can then be used to capture the delay obtained during the trip. In these shipping networks, such a point does not exist. This means that the ship will make round tours without resting in a certain harbour to cover the obtained delay. When it is preferred to have such a resting point, this can easily be implemented by increasing the corresponding buffer time.

3.3 Performance

The objective in this thesis is to minimize total costs. Therefore, costs are used as performance measure. Some costs, which are incurred, are hard to express in monetary units, like reputation. In this thesis it is assumed that all costs are included in the monetary costs. Thus, it is assumed that a method exists to express such costs in monetary units. Furthermore, the fraction of time that a ship has a delay of less than 3 hours is viewed for each model. This fraction is called the punctuality.

4 Model

In this chapter, the problem described will be modelled using a Markov decision process. First, a list of sets and variables, which are used throughout this document, is provided. Next, it is explained how the problem can be implemented in a model and which methods can be used to solve the model.

4.1 List of sets and variables

Throughout this document, the following notation is used, which is mostly as used by Hillier and Lieberman (2001). The meaning of this list is to give an overview of the used formulation. The meaning of the sets and variables are explained when they are introduced in the document.

- $i \in I$: set of states of the Markov chain, for example $i = (\text{Rotterdam in}, 3)$ means that the ship has entered the port of Rotterdam with a delay of 3 hours.
- $j \in J$: dummy set of I used in the Linear Programming formulation, meaning that J consist of the same elements as I.
- $k \in K$: set of possible recovery actions, for example $k = 0$ means that nothing special will be performed.
- $h \in H$: set of harbours.
- $a \in A$: set of possibilities for a harbour, so $A = \{\text{in}, \text{out}\}$.
- $d \in D$: set of possible hours of delay.
- $l \in L$: set of possible port combinations, for example $l = (\text{Rotterdam in})$ means that the ship enters the port of Rotterdam.
- $C_i(k)$: costs of performing recovery action k when the Markov chain is in state i.
- $p_{ij}(k)$: probability that the Markov chain will be the next period in state j, given that it is in state i at the moment and action k is performed.
- π_i: steady-state probability of state i.
- D_{ik}: probability that action k is chosen in state i of the Markov chain.
- y_{ik}: probability that the Markov chain is in state i and action k is chosen.
- R_n: policy obtained after n iterations of policy improvement algorithm.
- $g(R_n)$: (long-run) expected average costs per unit time following policy R_n.
- $v_i(R_n)$: effect on the total expected costs due to starting in state i, when following policy R_n.
- $g_i(R_n, b)$: estimation of $g(R_n)$, with steady state probabilities obtained from policy R_n and the buffer time in state i is set to b hours.
- $E[S_{ld}(b)]$: expected savings in state ld when the buffer in port combination l is b hours.
- $E[C_{ld}(b)]$: expected costs in state ld when the buffer in port combination l is b hours.

4.2 Markov Decision Process

Before a definition of a Markov decision process can be given, three sets have to be defined. These sets are:

- I: set of possible states of the Markov process.
- J: dummy set of I. This means that I and J contains the same elements.
- 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, where after each transition an action $k \in K$ has to be chosen from a set of available actions (Hillier and Lieberman (2001)). Each possible action is associated with a cost $C_i(k)$, where $i \in I$ is the state in which the process is at that moment. Further, the action k performed in state i determines the transition probabilities for the next transition from state i. These transition probabilities are denoted by $p_{ij}(k)$, where $j \in J$ is the state the Markov process is moving to with the given probability. The objective of the process is to find an optimal policy according to the given costs. For each possible state of the Markov chain, a policy has to be defined which action should be performed when the process is in this state. After the model is defined, a method has to be chosen to solve the model.

The objective function (performance measure) can be computed in two ways (Hillier and Lieberman (2001)). The first way is to use the (long-run) expected average cost per unit time, which can be computed as $E(C) = \sum_i C_i(k)\pi_i$, where $i \in I$ denote the different Markov states. The second way to compute the performance measure is to use the expected total discounted cost. This measures uses a discount factor α, with $0 < \alpha < 1$. α denotes the present value of one unit of cost for one period in the future. The formula for α is then $\alpha = \frac{1}{1+i}$, where i is the interest rate per period. The value of costs for more than one period in the future can also be computed. This can be performed by multiplying the costs with α^n, where n is the number of periods in the future. For the problem discussed in this document, the first performance measure is used, because the interest rate is unknown. Furthermore, the advantage of the expected total discounted cost criterion on the (long-run) expected average cost criterion arises when the time that expires is long, so the time value of money cannot be neglected anymore (Hillier and Lieberman (2001)). In this problem the time that a ship needs to sail from one port to another is expressed in hours, and corresponds with days, not with years. Therefore, it is not necessary to discount the costs.

Next Hillier and Lieberman (2001) give three methods to solve the model. These three methods are exhausted enumeration, linear programming and policy improvement algorithms. The first method, exhausted enumeration, makes a list of all relevant policies (a policy defines an action for each possible state of the Markov chain) and computes the costs for each of these policies. The policy with the lowest cost is seen as the optimal policy. This method is very useful for very small problems, because it can be solved fast and it gives a good understanding of the model. The method is not useful for larger methods, because the list of relevant policies increases much harder than the number of states, when the number of states becomes larger. The second method, linear programming, translates the model to a linear programming formulation. This formulation can be implemented in a computer program, which is made to solve linear programming models, for example Aimms® [1]. Because the linear programming problem can be implemented in a computer program, it is possible to solve much larger problems than with the exhausted enumeration method. Furthermore, the model is not very hard to implement. The last method, policy improvement algorithms, starts with an initial policy and uses an algorithm to improve this policy. The advantage of this method in comparison with the linear programming problem is that is tends to be very efficient (Hillier and Lieberman (2001)). This means that an optimal policy is found after a few iterations of the algorithm, where the linear programming model needs more iterations. Since computer programs are available, which can solve linear programming models and because the model can be implemented in such a program, the linear programming method is used to solve the problem.

For this problem a state is defined as (port name or number in / out, amount of delay). Therefore, a state consists of three elements: the port where the ship is or where it is going to; whether the ship is going to the port (in) or is loading or unloading to leave the port (out) and the number of hours the ship has been delayed (the delay is made discrete in units of hours).

In figure 3 a small example with only two ports to visit is visualized. H1 and H2 are abbreviations for Harbour 1 and Harbour 2 respectively. d_1, d_2, d_3 and d_4 denote the delay for respectively Harbour 1 in, Harbour 1 out, Harbour 2 in and Harbour 2 out. This delay can be all integer values between 0 and a specific value d_{max}. In reality the delay does not have to be an integer number. This cannot be implemented, because the objective is to formulate a policy, which describes for every possible states of the Markov process which action should be performed if the process is in that state. The number of states has to be finite to be able to define this for every state and to use a linear programming formulation. Therefore, a finite number of possible delays has to be defined. A finite number of possible delays can be realized by making the delay discrete. This is performed by rounding all delays to the nearest integer that is greater than or equal to the delay. Therefore, in other words, it is assumed that a ship can only leave or arrive at their scheduled time or exactly $1, 2, 3, \ldots$ hours later. Finally $p_{ij}(k)$ is the probability that the ship go from state i to state j while it takes action k. In this formula i and j can be (Harbour 1 in, d_1), (Harbour 1 out, d_2), (Harbour 2 in, d_3) and (Harbour 2 out, d_4), with d_1,d_2,d_3 and d_4 as specified above.

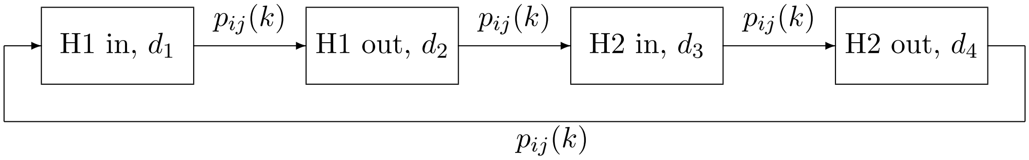


Figure 3: Example of a Markov decision process

[1] Aimms is a mathematical programming language, which can be used to solve models, for example linear models. More information on Aimms® can be found on the website: `www.aimms.com/`

4.3 Linear Programming formulation

Now the states of the Markov chain are defined, it is possible to make a variable D_{ik} which indicate the probability that action k is chosen when the Markov chain is in state i, so $0 \leq D_{ik} \leq 1$. The problem is to find the optimal values of D_{ik}. The decision variables of the linear programming model are $y_{ik} = P(\text{state} = i \text{ and action} = k)$. In words y_{ik} is defined as the probability that the Markov chain is in state i and the action performed is k. Note that y_{ik} and D_{ik} are closely related to each other. It follows that

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

Or

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

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

$$\pi_i = \sum_k y_{ik}.$$

From the previous formulas it can be concluded that

$$D_{ik} = \frac{y_{ik}}{\pi_i} = \frac{y_{ik}}{\sum_k y_{ik}}.$$

The constraints on y_{ik} are:

- $\sum_i \pi_i = 1$, or in terms of y_{ik}: $\sum_i \sum_k y_{ik} = 1$.
- $\pi_j = \sum_i \pi_i p_{ij}$, or in terms of y_{ik}: $\sum_k y_{jk} = \sum_i \sum_k y_{ik} p_{ij}(k)$ for $j \in J$.
- $y_{ik} \geq 0$ for $i \in I, k \in K$.

The expected costs are equal to

$$E[C] = \sum_i \sum_k \pi_i C_i(k) D_{ik} = \sum_i \sum_k C_i(k) y_{ik}.$$

Hence, the linear programming model, which has to be solved, is:

$$\min \sum_i \sum_k C_i(k) y_{ik}$$

subject to the following constraints:

$$\sum_i \sum_k y_{ik} = 1.$$

$$\sum_k y_{jk} = \sum_i \sum_k y_{ik} p_{ij}(k) \qquad j \in J.$$

$$y_{ik} \geq 0 \qquad i \in I, k \in K.$$

5 Small example

It is useful to start with a small example to test whether the model works as expected. Furthermore, the consequences of changing the costs in one state on the decisions to be made can be investigated easier with a small example.

5.1 Data

In table 1 the expected times the ship needs in and between ports are given. The first two rows of the table provide the expected time in port and the last two rows of the table gives the expected sailing time between two ports. The last column of both tables denotes the buffer time for the given situation. For example, the expected time needed in port 1 is 23 hours with an additional hour buffer time and the expected sailing time from harbour 1 to harbour 2 is 46 hours with an additional buffer time of 2 hours.

Table 1: Expected time in and between ports

Harbour	Sailing Time (hr)	Buffer Time (hr)
1	23	1
2	23	1
1-2	46	2
2-1	41	3

This data can be used to make a time schedule for the ship. Assume that the ship arrives in harbour 1 at time 0. The ship will then leaving port 1 at time $23 + 1 = 24$. Because it take 46 hour to sail from port 1 to port 2 and a buffer time of 2 is included, the ship will arrive at time $24 + 46 + 2 = 72$ in harbour 2. The other arrival and departure times can be calculated in the same way. These outcomes can be found in table 2. The first column denotes the number of the harbour that the ship has to visit. The second column gives the arrival time of the ship in that harbour, while the third column denotes the departure time from that harbour. Because the ship is assumed to sail round tours, the arrival in harbour 1 means the start of a new round tour. The expected times in port and sailing times are the same for each round tour, as well as the buffer times. From table 2 can be concluded that the time for one round tour is 140 hours.

Table 2: Time schedule of a ship

Harbour	Arrival Time (hr)	Departure Time (hr)
1	0	24
2	72	96
1	140	168

Therefore, the expected time that a ship needs in harbour 2 is 23 hours. Assume that a ship arrives without delay in harbour 2. When the ship indeed is ready to leave after 23 hours, it has to wait for one hour, because it can only leave after 24 hours (because a buffer time of one hour is included and the ship cannot depart before the planned departure time). When the ship has a delay of 1 hour or less, it still can departure on time. However, when the delay of the ship exceeds 1 hour, it also has a delay on their time schedule. In this case, the ship encounters extra costs. In the other cases the ship may have a delay, but because the delay is smaller than the buffer time, it does not encounter extra costs, because the delay cause no violence to the time schedule.

The possible actions, which can be performed by the ship, are given in table 3. Action 1 can only be performed when the ship is in a port and action 2 can only be performed when the ship is between two ports. In the last column the gain of performing the action is given. Increasing the speed of a ship will lead to a 5% gain. Because the sailing times from port 1 to port 2 and vice versa are respectively 46 and 41 hours, the gain of action 2 will be approximately two hours for both sailing routes.

Table 3: Possible actions

Action	Description	Gain
0	Do nothing	0 hour
1	Increase port-handling capacity	1 hour
2	Increase speed	5%

In table 4 the assumed probabilities of an additional delay of 0, 1, 2, 3 and 4 hours are shown. This probabilities are only valid when the ship chooses action 0, so when nothing is performed. When the ship decides to take another action, the additional hours have to be reduced by the number of hours gained by taking this action. Therefore, when a ship decides to take action 1, the probabilities in the table are not valid for 0, 1, 2, 3 and 4 hours additional delay respectively, but they are valid for 1 hour gain, 0, 1 and 2 hours additional delay respectively. As in table 1 the first two rows are the probabilities of additional delay in a port and the last two rows are the probabilities of an additional delay between two ports. The probabilities stated in the table can be used to compute the transition probabilities. For example, when a ship arrives in port 2 with a delay of 4 hours, it has to choose whether it will take action 0 or action 1. When it chooses for action 0, the ship will leave harbour 2 with a delay of 3, 4 or 5 hours with probability 0.1, 0.4 and 0.5 respectively. Note that the probabilities of an additional delay of 2 and 3 hours are added together, because a maximum delay of 5 hours is assumed. Furthermore, note that the additional delay is reduced with one hour, because there is a buffer time of 1 hour in port 2. However, when the ship decides to perform action 1, it will leave harbour 2 with a delay of 2, 3, 4 or 5 hours with probabilities 0.1, 0.4, 0.35 and 0.15 respectively. In the same way the transition probabilities for all other possible states can be computed. Note that when a ship has no delay, choosing for action 1 or 2 cannot reduce the delay, but increases the probability that the ship will also be without delay in the next state.

Finally, costs of delay and costs of performing a recovery action are needed. These costs

Table 4: Probabilities of a delay for given hours in and between ports

	Delay				
Harbour	0	1	2	3	4
1	0.2	0.3	0.3	0.2	0
2	0.1	0.4	0.35	0.15	0
1-2	0.1	0.15	0.25	0.35	0.15
2-1	0.05	0.2	0.2	0.3	0.25

can be found in table 5 and 6. Table 5 shows the costs of 1, 2, 3, 4 and 5 hours delay. The assumption in this example is that the maximum amount of delay will never exceed the 5 hours. Furthermore, when a ship encounters a delay, this delay will be rounded to the next integer, which is greater than or equal to the delay. Costs of delay can differ between ports and they can also be different for arrivals and departures. An additional unit of delay is in general more expensive than the previous unit of delay. Table 6 presents the costs of performing a recovery action. A - sign means that the action is not allowed to perform when the ship is in the associated state. Action 1 is only allowed when the ship is in the port and action 2 is only allowed when the ship is between two ports. The costs of performing a recovery action are additional to the costs of the delay. For example, when a ship arrives with a delay of 4 hours in port 1 and decides to perform action 1, the total costs are $10 + 5 = 15$, where the costs of the delay are equal to 10 and the costs of performing action 1 are equal to 5.

Table 5: Costs of delay

	Delay					
Harbour	0	1	2	3	4	5
1	0	2	3	5	10	20
2	0	2	3	5	10	20
1-2	0	3	5	9	15	25
2-1	0	5	10	15	25	35

Table 6: Costs of a recovery action

	Action		
Harbour	0	1	2
1	0	5	-
2	0	5	-
1-2	0	-	7
2-1	0	-	10

Now it is possible to run the linear programming problem for this example in Aimms®. Next, the outcomes of the model can be evaluated by changing some of the costs and try to find a explanation for the difference in results.

5.2 Implementation in Aimms®

Some difficulties occur when implementing the above model in Aimms®. Aimms® is a very useful program to solve linear programming problems. Aimms® works with sets, parameters, variables and constraints. Furthermore, a mathematical program has to be defined. The meaning of a set is convenient.

In this problem two sets have to be defined: the set with all possible states of the Markov process and the set with all possible actions. The dummy set is only needed for notational purposes and does not have to be defined in Aimms®. Because it is very time consuming to insert all possible states of the Markov process, it is useful to make use of a little trick.

First sets of all harbours, a set of all possible delay and a set with as elements *in* and *out*, which indicate whether the ship is arriving in the harbour or leaving the harbour in Aimms®, are made. Now, it is possible to define the set of states of the Markov process. This can be performed by selecting the three above sets in the 'subset of' row. The indices of this set are i and j. This set has two indices, because a dummy set is needed. The index of the set of actions is k.

Parameters are all data, which are known in advance. In this problem the parameters are $p_{ij}(k)$ and $C_i(k)$. These parameters can be made in Aimms® and their values can be inserted.

Variables are unknown in advance. The program has to compute the values of the variables. In this problem the variables are y_{ik} and the total costs. The range of y_{ik} has to be set to nonnegative. It is not allowed to set a range for the total costs, because this is the objective variable.

The two constraints of the problem (the last constraint is already implemented by setting the range of y_{ik} to nonnegative) can be implemented.

To run the program, a mathematical program has to be defined. The objective has to be the total costs and the direction minimization. The type used for this problem is LP. Finally, the MainExecution needs to solve the mathematical program. The solver used is the XA14 solver.

5.3 Results

The values of y_{ik} received with the linear programming model, can be easily transformed to values of D_{ik}, which give the probability that action k is chosen in state i with the formula $D_{ik} = \frac{y_{ik}}{\sum_k y_{ik}}$. The solution y_{ik} will only contain one value which deviates from zero for every value of i. This means that in each state i of the Markov chain only one action is optimal. However, the nonzero values of y_{ik} themselves also contain some useful information. Because the D_{ik} value will be 1 for only one value of k for each value of i, the nonzero values of y_{ik} can be interpreted as the steady state probabilities for state i.

In table 7 the steady state probabilities are shown for each state of the Markov chain. In each table the probabilities for one part of the shipping route is shown. A value of 0.000 in the table denotes a steady state probability, which is between 0 and 0.0005. A value of 0 in the table means that the Markov chain will never be in the denoted state. From the steady state probabilities, it can be concluded that the ship will have no delay in about 50% of the time. This value can be found by adding the steady state probabilities of all the states without a delay. In table 8 the fraction of the time that the ship has 1,2,3,4 and 5 hours delay are also presented. Furthermore, the cumulative fractions are shown. The cumulative fractions denote the fraction of the time that the maximum delay is the given number of hours. The value of 1.000 in the table denotes a value between 0.9995 and 1. It can be concluded from the table that in 99% of time a ship has a maximum delay of 2 hours. This seems a reasonable percentage.

The results of the small example discussed in the previous paragraph, can be found in table 9. The numbers in the table denote which action should be performed when the ship is in that state (combination of harbour and delay). The - sign in the table means that the ship will never come in the associated state, so no action has to be performed for that state.

Table 7: Steady state probabilities

Harbour 1	
Delay	Prob
0	0.160
1	0.067
2	0.023
3	0.000
4	0
5	0
Total	0.250

Harbour 1-2	
Delay	Prob
0	0.118
1	0.075
2	0.052
3	0.005
4	0.000
5	0
Total	0.250

Harbour 2	
Delay	Prob
0	0.150
1	0.072
2	0.027
3	0.001
4	0.000
5	0
Total	0.250

Harbour 2-1	
Delay	Prob
0	0.114
1	0.089
2	0.043
3	0.004
4	0.000
5	0.000
Total	0.250

Table 8: Fraction of time with a certain delay

Delay	Fraction	Cum Fraction
0	0.541	0.541
1	0.303	0.844
2	0.146	0.990
3	0.010	1.000
4	0.000	1.000
5	0.000	1

From table 9, it can be concluded that the size of the buffer time plays a role in determining the best action to choose. The buffer time is highest for arriving in port 1. Recovery actions are indeed performed later when a ship is sailing from port 2 to port 1 than when a ship is in port or sailing from port 1 to port 2. This can partly be explained because of the higher buffer time for arriving in a port 1. Though, the costs of a delay and recovery action and the transition probabilities will also have influence on the actions.

Table 9: Best actions for the example

Harbour	Delay					
	0	1	2	3	4	5
1	0	1	1	1	-	-
2	0	1	1	1	1	-
1-2	0	2	2	2	2	-
2-1	0	0	2	2	2	2

The average costs of one round tour are 4.26. This seems very low, because the costs of the cheapest recovery action are 5, but remember that the ship does not encounter a delay more than half the time. The time used by Aimms$^{®}$ to solve this problem is less than 0.1 second.

5.4 Validation

Previously, the results of the model are discussed. However, do these results indeed lead to an optimal policy? This question will be answered in this section. To answer the question, several methods are used. First, the steady state probabilities are investigated. Next, it is tried to improve the result of the model using policy improvement. Finally, a few changes are made in the data of the model and the results are compared to expectations.

5.4.1 Steady state probabilities

In table 7 the steady state probabilities are shown for each state of the Markov process. Furthermore, for part of the round tour the steady state probability of being in that harbour or between those harbours is presented. Because the round tour of the ship can be divided in four different parts (harbour 1, 1-2, 2 and 2-1) and the ship will go from one part to the following with probability 1, the sum of the steady state probabilities of each of the four parts has to equal 0.25. This is indeed the case, as can be concluded from table 7.

However, this is not the only condition on the steady state probabilities. When the optimal policy, which defines the actions to take in each possible state, is known, the corresponding transition probabilities can be calculated. With this transition probabilities, a system of equalities can be made. If these equalities indeed hold, the steady state probabilities are valid. These equalities are $\pi_i = \sum_j p_{ji}(k) * \pi_j$. First an example will be shown. A ship can arrive in harbour 1 with 2 hours delay in the following ways:

- It can leave harbour 2 with a delay of 5 hours and thus it is known from table 9 that the ship will perform action 2 and gain 2 hours. Furthermore, the ship has a buffer time of 3 hours, so the total delay is now $5 - 2 - 3 = 0$ hours. The ship then has to obtain 2 hours delay while sailing from port 2 to port 1. From table 4, it is known that the probability of this situation is 0.2.

- It can leave harbour 2 with a delay of 4 hours and thus it is known from table 9 that the ship will perform action 2 and gain 2 hours. Furthermore, the ship has a buffer time of 3 hours, so the total delay is now $4 - 2 - 3 = -1$ hours. The ship then has to obtain 3 hours delay while sailing from port 2 to port 1. From table 4, it is known that the probability of this situation is 0.3.

- It can leave harbour 2 with a delay of 3 hours and thus it is known from table 9 that the ship will perform action 2 and gain 2 hours. Furthermore, the ship has a buffer time of 3 hours, so the total delay is now $3 - 2 - 3 = -2$ hours. The ship then has to obtain 4 hours delay while sailing from port 2 to port 1. From table 4, it is known that the probability of this situation is 0.25.

- It can leave harbour 2 with a delay of 1 hour and thus it is known from table 9 that the ship will perform action 0 and gain 0 hours. Furthermore, the ship has a buffer time of 3 hours, so the total delay is now $1 - 0 - 3 = -2$ hours. The ship then has to obtain 4 hours delay while sailing from port 2 to port 1. From table 4, it is known that the probability of this situation is 0.25.

These are the only four possibilities to reach the specific Markov state. It is not possible to reach the specific state with a delay of 0 or 2 hours when the ship leaves port 2, because an additional delay of 5 hours while sailing from port 2 to port 1 is needed and this has a probability of 0. The steady state probabilities of the above three states can be found in table 7 and are respectively 0.000, 0.000, 0.004 and 0.089. The steady state probability of arriving in harbour 1 with a delay of 2 hours should then be $0.2 * 0.000 + 0.3 * 0.000 + 0.25 * 0.004 + 0.25 * 0.089 = 0.023$. This is indeed the steady state probability as presented in table 7.

It is possible to write a program in MATLAB®[2] to verify these equalities. First the transition probabilities have to be made by selecting the probabilities corresponding with the actions made for every state. When the matrix of transition probabilities, T, is known, the steady state probabilities s has to equal $s = T's$. Because MATLAB® uses numeric methods to compute these equations, it is checked whether $T's - s < 0.0001$. These inequalities appear to hold, so the steady state probabilities are valid.

5.4.2 Policy improvement

Policy improvement is a tool, which can be used to improve a certain policy. When a policy is available, it will use this to compute costs and will return the policy with lowest costs. Since costs are dependent on the policy, this step has to be repeated as long as the returned policy differs from the initial policy. When the initial policy is exactly the same as the returned policy, the policy is optimal. This method can be used to check the validity of the model. The policy returned by Aimms® is used as initial policy. If the policy improvement method returns the same policy, this policy is optimal. The policy improvement algorithm used is the one explained in Hillier and Lieberman (2001). They provide the following summary of the algorithm.

> *Initialization:* Choose an arbitrary initial trial policy R_1. Set $n = 1$.
>
> *Iteration n:*
>
> *Step 1: Value determination:* For policy R_n, use $p_{ij}(k)$, $C_i(k)$, and $v_M(R_n) = 0$ to solve the system of $M + 1$ equations
>
> $$g(R_n) = C_i(k) + \sum_{j=0}^{M} p_{ij}(k) v_j(R_n) - v_i(R_n), \quad \text{for } i = 0, 1, \ldots, M,$$
>
> for all $M + 1$ unknown values of $g(R_n)$, $v_0(R_n)$, $v_1(R_n)$, …, $v_{M-1}(R_n)$.
>
> *Step 2: Policy improvement:* Using the current values of $v_i(R_n)$ computed for policy R_n, find the alternative policy R_{n+1} such that, for each state i, $d_i(R_{n+1}) = k$ is the decision that minimizes
>
> $$C_i(k) + \sum_{j=0}^{M} p_{ij}(k) v_j(R_n) - v_i(R_n)$$
>
> i.e., for *each* state i,
>
> $$\min_{k \in K} \, [C_i(k) + \sum_{j=0}^{M} p_{ij}(k) v_j(R_n) - v_i(R_n)],$$

[2] MATLAB® is a numerical computer program. More information can be found on the website: www.mathworks.com/

and then set $d_i(R_{n+1})$ equal to the minimizing value of k. This procedure defines a new policy R_{n+1}.

Optimality test: The current policy R_{n+1} is optimal if this policy is identical to policy R_n. If it is, stop. Otherwise, reset $n = n+1$ and perform another iteration.

One iteration of the policy improvement policy is performed to check the validity of the model. The initial policy is the policy returned by Aimms®. The computations are performed in MATLAB®. The new policy returned by MATLAB® is the same as the initial policy. Therefore, it can be concluded that the policy returned by Aimms® indeed is optimal.

5.4.3 Increasing the costs of delay

Next, it is checked whether the results of a scenario that is slightly different, are as can be expected. To perform this, the costs of a delay when leaving port 1 are increased. When the costs of a delay when leaving port 1 are higher, it can be expected that ships try to leave port 1 with a delay as low as possible. Therefore, it is expected that ships will earlier perform recovery actions while sailing from port 2 to port 1 than in the original scenario. These expectations will now be checked by solving the new model in Aimms®. The results, after increasing the costs of a delay in port 1, are as shown in table 10. Therefore, if the costs of a delay in port 1 are higher, action 2 will be performed sooner while sailing from port 2 to port 1, than when the costs of a delay of port 1 are as stated in the previous chapter. This corresponds exactly with the expectations. Recovery action 1 is also earlier performed in harbour 2. This was not expected, and it may seem strange first, because it leads to performing action 1 in port 2 when the ship does not have any delay. However, why should it be necessary to perform a recovery action when the ship does not have encountered any delay? To answer this question, the costs of a delay while leaving harbour 2 and the costs of performing a recovery action while sailing from port 2 back to port 1 has to be taken into account. Furthermore, the transition probabilities play a role in determining the best action. The costs of a delay while leaving harbour 2 are much higher than the costs of a equal delay while arriving in harbour 2. Furthermore, the costs of a recovery action while sailing from harbour 2 to 1 are twice as high as while loading/unloading in harbour 2. By performing a recovery action in harbour 2, even when the ship has no delay, the probability of leaving harbour 2 on time increases. When the ship can leave harbour 2 on time, it does not have to pay for a delay, so the costs are lower. Furthermore, when the delay is less while leaving harbour 2, it will also in expectation be less when arriving in harbour 1, because the transition probabilities are independent on the past. Therefore, taking a recovery action in port 2 will decrease the delay when the ships leave port 2, but also when the ship arrives in port 1, because the transition probabilities are independent on the past. Therefore, this also leads to less delay in port 1, so less extra costs.

Furthermore, the steady state probability of arriving in harbour 1 without a delay is increased from 0.160 to 0.201. Since the round tour of the ship can be divided in four different parts (harbour 1, 1-2, 2 and 2-1) and the ship will go from one part to the following with probability 1, the sum of the steady state probabilities of each of the four parts has to equal 0.25. It can thus be concluded that the ship performs recovery action before arriving

Table 10: Best actions for the example with higher costs of a delay in port 1

	Delay					
Harbour	0	1	2	3	4	5
1	0	1	1	1	-	-
2	1	1	1	1	-	-
1-2	0	2	2	2	2	-
2-1	0	2	2	2	2	-

in harbour 1 at such a level that it will arrive in port 1 without a delay in four out of five cases. It does not matter how much the costs are increased, so it is not useful to compare the average costs of one round tour.

5.4.4 Increasing the costs of a recovery action

In this section the results of increasing the costs of recovery action 1 in harbour 1 are discussed. When it becomes more expensive to perform action 1 in harbour 1, it is expected that ships will choose to perform action 1 in port 1 in a later stadium than in the original model. It is even possible that they do not perform action 1 in port 1 anymore. Because this can lead to higher delays, it is expected that ships will take more actions before arriving in port 1. These expectations will now be checked by solving the new model in Aimms®. The results, after increasing the costs of recovery action 1 in port 1, are shown in table 11. Therefore, if the costs of a recovery action of type 1 in port 1 are higher, action 0 will be performed in port 1. Instead of taking action 1 in port 1, it is preferred to choose action 1 in port 2 even if the ship does not encounter a delay at the moment. The results of the model correspond indeed to the expectations.

Table 11: Best actions for the example with higher costs of a recovery action of type 1 in port 1

	Delay					
Harbour	0	1	2	3	4	5
1	0	0	0	0	0	-
2	1	1	1	1	1	1
1-2	0	2	2	2	2	2
2-1	0	0	2	2	2	2

The steady state probabilities of this policy do not differ much from the steady state probabilities of the original model. In this model the probabilities of being in a state with a small delay are a slightly higher than in the original model. The average costs of one round tour are slightly increased, from 4.26 to 4.42. The increase in costs is due to the higher costs for taking a recovery action in port 1.

5.5 Conclusion

The purpose of this chapter is to investigate the use of a Markov decision model for this problem. Some of the research questions can now be answered. When the states of a Markov decision model are properly defined, the problem can be modelled using a Markov decision model. The model can be solved using the mathematical language Aimms®. Furthermore, little time is needed to solve the model. Now, the validity of the model is guaranteed, it is possible to implement larger examples in the model.

6 Real example

Now the validity of the model is checked, the model is used to determine the optimal policy for a real shipping network. First the data of this network will be shown. Next the results of the model are discussed.

6.1 Data

The data used in this section are obtained from Gkanatsas (2005). In the first column of table 12 the harbours, which will be visited by the ships, are shown. The next column presents the distances between the harbours. This column has to be read in the following way. The distance shown for each harbour is the distance that the ship has to cover to reach that harbour. For example, the distance between Lisbon and Charleston can be found in the row of Charleston and is equal to 3385 Nautical miles. The third column presents the speed at which the ship is sailing on their way to the harbour. The fourth column shows the number of crane moves needed to load and unload the cargo for each port. In the last column the speed at which the cargo is loaded and unloaded is shown. Note that an equal speed of loading and unloading is assumed.

Table 12: Some characteristics about the harbours

Harbour	Distance	Sailing Speed	Load/Unload	(Un)Loading Speed
	(NM)	(NM/hr)	(moves)	(moves/hr)
Charleston	3385	19	468	60
Miami	435	19	421	60
Houston	977	19	702	60
New Orleans	433	19	468	60
Antwerp	4859	19	655	65
Felixtowe	70	19	515	55
Bremerhaven	270	19	655	65
Rotterdam	215	19	468	60
Lisbon	1086	19	328	65

In table 13 the first column denotes several harbours that the ships visit. The second table shows the expected sailing time needed to reach that harbour. The expected sailing time for Antwerp is 270 hour. This means that a ship needs 270 hours to sail from New Orleans to Antwerp, when it does not encounter a delay during their trip. The expected sailing times can be computed with table 12 by dividing the distance between two ports by the sailing speed. The third column denotes the time needed in the harbour to load and unload the ship. These times can also be computed with table 12. Now, the amount of cargo to be loaded and unloaded has to be divided by the loading/unloading speed. The last two columns denote the buffer times for the ports. The fourth column shows the buffer time for arriving in a port. Therefore, when a ship encounters a delay of 3 hours while sailing from Miami to Houston with respect to the expected sailing time, the real delay of the ship is $3-2 = 1$ hour. The fifth column denotes the buffer time for leaving a port. These buffer times are subtracted from the delay encountered in the port, to determine the delay with respect to the time schedule when the ship leaves the harbour.

Table 13: Sailing, port and buffer times for the real example

Harbour	Sailing Time (hr)	Port Time (hr)	Buffer in (hr)	Buffer out (hr)
Charleston	-	8	-	3
Miami	23	7	2	3
Houston	51	12	3	1
New Orleans	23	8	2	2
Antwerp	256	10	2	1
Felixtowe	7	9	1	1
Bremerhaven	16	10	1	1
Rotterdam	13	8	1	1
Lisbon	57	5	1	2
Charleston	178	-	4	-

With table 13 a time schedule for ships can be made. The time schedule can be found in table 14. In this case the assumption is that a ship arrives in Charleston at time 0. The computations of the arrival and departure times are the same as in the small example discussed earlier. The time needed to make one full round tour is 733 hours.

Table 14: Timeschedule for the real example

Harbour	Arrival Time (hr)	Departure Time (hr)
Charleston	0	11
Miami	36	46
Houston	100	113
New Orleans	138	148
Antwerp	406	417
Felixtowe	425	435
Bremerhaven	452	463
Rotterdam	477	486
Lisbon	544	551
Charleston	733	744

The possible actions, which can be performed, to decrease the delay are the same as in the small example. Therefore, these actions are:

- Do nothing.
- Increase port-handling capacity.
- Increase sailing speed.

For both recovery actions, two levels are considered, a low and a high level. The low level is as presented before in this section. In the case of a high level, the sailing speed is assumed to be 23 miles/hour. If the port-handling capacity is increased, the (un)loading speed becomes higher. In the high level case, this speed is assumed to be 80 crane moves/hour. Table 15 presents the gain of increasing the port-handling capacity. The second column shows the new port times. These can be computed by dividing the number of crane moves needed to load and unload the cargo by 80. The last column shows the gain of performing recovery action 1. These gains can be computed by subtracting the new port times from the port times in table 13.

Table 15: Gain of recovery action 1: Increasing the port-handling capacity

Harbour	New Port Time (hr)	Gain (hr)
Charleston	6	2
Miami	5	2
Houston	9	3
New Orleans	6	2
Antwerp	8	2
Felixtowe	6	3
Bremerhaven	8	2
Rotterdam	6	2
Lisbon	4	1

Table 16 shows the gain of performing recovery action 2 while sailing between two ports. In the second column the new sailing times are shown. These sailing times can be computed by dividing the distances from table 12 by 23. The last column shows the gain of performing action 2. These gains can be computed by subtracting the new sailing times from the sailing times in table 13.

Table 16: Gain of recovery action 2: Increasing the sailing speed

Harbour	New Sailing Time (hr)	Gain (hr)
Charleston	19	4
Miami	42	9
Houston	19	4
New Orleans	211	45
Antwerp	6	1
Felixtowe	14	2
Bremerhaven	11	2
Rotterdam	47	10
Lisbon	147	31

Gkanatsas (2005) assumes that probabilities of an additional delay are constant between ports. Two distributions of delay are considered: weather condition delays and port time condition delays. Weather condition delays are responsible for the delay ships encounter while sailing from one port to another and port time condition delays are responsible for the delay ships encounter while in port. The probabilities do not depend on the time a ship is sailing of or in port. Table 17 presents the probability of an additional delay of a given number of hours when a ship is in port or out of port. The second column shows the weather condition probabilities, while the last column shows the port time condition probabilities.

Table 17: Probabilities of an additional delay

Delay	Prob Out of Port	Prob In Port
0	0.02	0.05
1	0.05	0.30
2	0.24	0.10
3	0.20	0.20
4	0.27	0.35
5	0.14	-
10	0.08	-

It seems strange that the probabilities of delay are independent on the time needed in or out of port, because it is expected that more delay can be obtained when a ship for example has to sail a very long distance than when it has to sail only a little distance. In first instance however, this assumption will be made. Later in this document a situation with virtual ports is introduced.

Finally, the costs are needed to implement in the model. Because no data on costs of delay are available, the costs of delay is in first instance assumed to be linear with a costs of $100 per hour delay. Furthermore, it is assumed that these costs are only incurred when arriving with a delay in a harbour. When a ship leaves a harbour with a delay, it will not encounter any costs. The costs of an additional crane move per hour are assumed to be $10. In the second column of table 18 the total costs can be found. The costs of increasing the sailing speed are available from Gkanatsas (2005). These costs are shown in the last column of table 18.

Now, it is possible to solve the linear programming problem using Aimms®. In the next section the results will be presented. However, first some expectations of the results will be formulated when observing the data. The costs and gains of performing action 2 while sailing from New Orleans to Antwerp and from Lisbon to Charleston are relatively high. Therefore, it can be expected that this action will not be taken with small amounts of delay. Furthermore, when the amount of delay is larger, it is possible that the action will be performed in the optimal strategy, but in that case less actions will probably taken in the port combinations just before New Orleans and Lisbon. Next, the gain per unit cost ratio of action 1 is lower than that of action 2. It can thus be expected that action 1 will be performed with less hours of delay than action 2.

Table 18: Costs of recovery actions

Harbour	Action 1 ($)	Action 2 ($)
Charleston	200	1252.2
Miami	200	2792.4
Houston	200	1246.5
New Orleans	200	13987.7
Antwerp	150	405.9
Felixtowe	250	895.3
Bremerhaven	150	734.1
Rotterdam	200	3126.3
Lisbon	150	9744.5

6.2 Results

Figure 4 shows the fraction of time that a ship has a certain delay for both all states and arriving states. The figure shows that a ship is most of time in a state with low delays. Furthermore, ships are almost never in states with a delay of more than 18 hours. The punctuality, which is defined as the fraction of time that a ship is in a state with a delay of 3 hours or less, is 63.9% when all states are considered and 60.0% when only arriving states are considered. These values can be found in table 26, which can be found in the appendix, section 10.1. This table shows an overview of the fraction of time that a ship has a certain delay and the fraction of time that a ship has a delay, which is smaller than or equal to the associated delay. The punctuality is higher when all states are considered than when only arriving states are considered. This is not as expected, because costs are associated to delays while arriving. Therefore, it is expected that more weight is given to delays in arriving states than to delays in departing states. The overall fraction should then be lower then the fraction when only arriving states are considered. The punctuality can be improved by increasing the costs of higher amounts of delay. This can be performed in two ways: by increasing the costs of delay per hour or by assuming that the function of the costs of delay is convex. An overview of all steady probabilities can be found in table 27 in the appendix, section 10.1.

In table 19 the best actions for the real example can be found. In the previous section some expectations were formulated about the optimal policy. The first expectation was that action of increasing the speed will not be performed while sailing from New Orleans to Antwerp and from Lisbon to Charleston, when the amount of delay is small. From table 19 it follows indeed that the action of increasing speed is never performed when sailing from New Orleans to Antwerp. Furthermore, it is only performed while sailing from Lisbon to Charleston when the delay is at least 22 hours. In the port combinations just before Lisbon, fewer actions are performed when the delay is high. This is indeed what was expected in the previous section. It can be explained in the following way. When a ship chooses to perform an action in a port combination just before leaving harbour Lisbon, it will reduce the delay, but this will lead to extra costs. The delay can be reduced to less than 22 hours when the ship leaves harbour Lisbon. In that case, action 2 will not be performed while sailing from Lisbon to Charleston. When no recovery actions are taken when the delay is high before leaving Lisbon, action 2 will

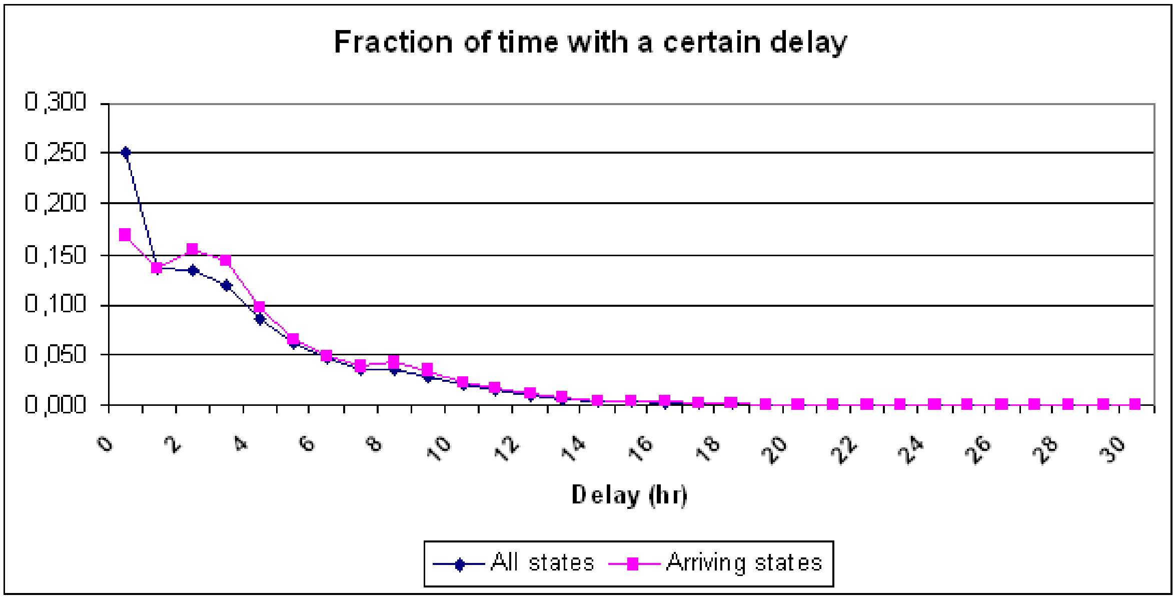


Figure 4: Fraction of time with a certain delay for the real example

performed while sailing from Lisbon to Charleston. This will lead to an enormous reduction of the delay. Of course, some costs will be associated with this action, but it will significantly reduce the delay costs. At the end, it will be more profitable to perform action 0 in the port combinations before Lisbon when the delay is high and then perform action 2 while sailing from Lisbon to Charleston. This will significantly reduce the costs.

In the small example, it was concluded that the size of the buffer time influences the amount of delay for which a recovery action is performed in that port combination. In the real example, this influence is also present. This is best shown by the recovery action, which increases port handling capacity, because the costs are about the same for this recovery action is the different ports. In harbours Charleston, Miami and Lisbon recovery action 1 is not performed when the ship has not encountered any delay yet. For all other ports, recovery action 1 is performed even when the delay equals 0. By comparing these observations with the buffer times in the ports, it can be seen that Charleston, Miami and Lisbon have a buffer time of 3,3 and 2 hours respectively, where all other ports (except New Orleans, which also has 2 hours buffer time) have a buffer time of 1 hour. Thus, a higher buffer time leads indeed to performing recovery actions later than a low delay.

The total costs for one round tour are $467. Aimms® needs less than 1 second to solve the model. The model considers 558 different states. This results in 1675 variables in this model. Furthermore, 560 constraints are available to solve the model. Now, the average delay in a state can be calculated, which turns out to be about 3.3 hours. The average delay in the arriving states is 3.7 hours. This was not as could be expected, because costs are associated to delays when ships arrive in a port, while delays when ships leave a port are costless. Therefore, it would have been expected that the average delay in arriving states should be lower than the average delay in all states. In figure 5 the average delay of each port is showed. The average delay is showed for both arriving and departing states. From figure 5, it can be seen that the average delay is highest in the ports Rotterdam, Bremerhaven, Felixtowe and Lisbon. Furthermore, the average delay is lowest in New Orleans, Houston and Miami. Note, that the ports combination for which the average delay is higher than average, have a low buffer time. The port combinations where the average delay is low, have higher buffer times. However, Charleston has the highest buffer times and this does not result in a low average delay. A relation between buffer times and average delays seems present, but some other factors also influence the average delay.

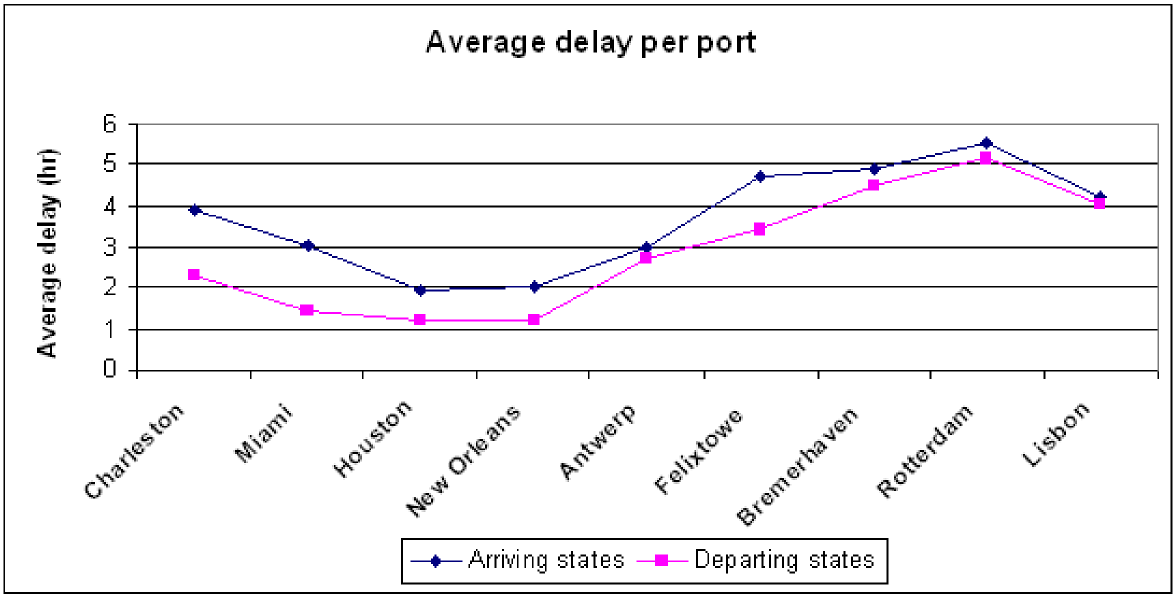


Figure 5: Average delay per harbour for the real example

Table 19: Best actions for the real example

	Charleston		Miami		Houston		New Orleans		Antwerp		Felixtowe		Bremerhaven		Rotterdam		Lisbon	
Delay	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
1	1	0	1	0	1	0	1	0	1	2	1	0	1	2	1	0	0	0
2	1	0	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
3	1	0	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
4	1	2	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
5	1	2	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
6	1	2	1	0	1	2	1	0	1	2	1	2	1	2	1	2	1	0
7	1	2	1	2	1	2	1	0	1	2	1	2	1	0	1	2	1	0
8	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
9	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
10	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
11	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
12	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
13	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
14	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
15	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
16	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
17	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
18	1	2	1	2	1	-	-	-	1	2	1	2	1	2	1	2	1	0
19	1	2	1	2	1	-	-	-	1	2	1	2	1	2	1	2	1	0
20	1	2	1	2	-	-	-	-	1	2	1	2	1	2	1	2	1	0
21	1	2	1	2	-	-	-	-	1	2	1	2	1	2	1	2	1	0
22	1	2	1	2	-	-	-	-	1	2	1	2	1	2	1	2	1	2
23	1	2	1	2	-	-	-	-	1	2	1	2	1	2	1	2	0	2
24	1	2	1	2	-	-	-	-	1	2	1	2	1	2	1	2	0	2
25	1	2	1	-	-	-	-	-	-	-	1	0	1	2	1	2	0	2
26	1	-	1	-	-	-	-	-	-	-	1	0	1	0	1	0	0	2
27	1	-	-	-	-	-	-	-	-	-	1	0	1	0	1	0	0	2
28	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	2
29	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	2
30	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	2

7 Extensions

In this section two extensions on the problem will be discussed. The first extension is the addition of a waypoint between two ports. A waypoint is a fixed point, where ships can compare their time with the scheduled time and decide on basis of the delay whether it will increase their speed or not. This will give the ship more opportunities to cover (parts of) their delay. The other extension is the reallocation of the buffer times. The objective is to find the best allocation of the buffer times over the ports, where the total buffer time has to be the same as before the reallocation.

7.1 Waypoints

First we will consider the addition of so-called waypoints. Until now it was only possible to perform a recovery action when arriving in a harbour or when leaving a harbour. When a ship sails between two ports, a possibility exists that the ship will encounter a delay in the beginning of the trip. At the end of the trip the delay can be increased or can be remained approximately the same, but the ship cannot have performed a recovery action to capture a part of the delay. This is a disadvantage, especially when the distance between two ports is large. This disadvantage can easily be removed by adding points between harbours. When a ship arrives at such a point, it can check their amount of delay. By knowing the delay, it can decide whether to perform a recovery action or not. In the situation, which is described above, the ship can decide at some points to increase their speed. The delay encountered in the beginning of the trip can in this way (partly) be covered when the ship arrives in the following harbour.

7.1.1 Implementation

Adding extra ports could perform the implementation of waypoints in the model. These ports are the waypoints. The difference between normal ports and waypoints is that a ship can go in and out a normal port, whereas it can only leave from a waypoint. Here, it can only perform a recovery action, which is associated with leaving a port. When a ship has a delay at a waypoint, no costs are imposed. Therefore, the only costs that can be incurred at a waypoint are costs associated with recovery actions. The gain has to be divided by two for each trip, because the distance whereover the speed is increased is decreased by half, so the gain will also decrease by half. This causes difficulties for some trips, because the delay has to be an integer number and te gain is an uneven number. This is solved by adding 30 minutes more gain to the first trip and giving 30 minutes less gain to the second trip.

The implementation causes a problem, because the probabilities of delay for the whole trip have to be converted to separate probabilities of delay for the trip to the waypoint and for the trip from the waypoint to the following port. Two distributions X_1 and X_2 have to be found in such a way that $X_1 + X_2 = X$, where X is the known discrete distribution. Because these two distributions are very hard to find (see the appendix in section 10.2), it is assumed in this thesis that the delay distribution of the trip to the waypoint and the delay distribution of the trip from the waypoint to the next port are known. The distribution of the delay for the whole trip between two ports can then be calculated from these known distributions.

7.1.2 Delay

In table 20 the probabilities of delay between a port and a waypoint are shown. It is also possible two add more waypoints for a trip. In that case, the probabilities in the table are also the probabilities of delay between two waypoints. It is also possible to use different distributions for different trips, however, since not enough information is available on the distributions of delay for given trips, it will not add extra value to the model to implement this.

Table 20: Probabilities of a delay for given hours between ports and waypoints

Delay	Prob
0	0.20
1	0.35
2	0.25
3	0.20

For every trip one waypoint will be added exactly between the two ports. By adding the waypoints exactly between two ports, the costs of performing a recovery action can be divided by two for each trip. The results of the model with the waypoints are compared with the results of the model without waypoints. For the last model, the probabilities of delay have to be calculated. This can be performed in the following way. First, all combinations of delay in the two separate trips between two harbours have to be made. The sum of the separate delays is the delay over the whole trip. The probability on this delay is the product of the probabilities of the two separate delays. Finally, the probabilities of all combinations that lead to the same total delay have to be summed to obtain the probability of that delay over the whole trip. A delay of 2 hours can for example be obtained by a first delay of 0 hours and a second delay of 2 hours. The probability of this situation is $0.2 * 0.25 = 0.05$. Other possible combinations are first 1 hour, second 1 hour and first 2 hours, second 0 hours. The probabilities of these delays are respectively $0.35 * 0.35 = 0.1225$ and $0.25 * 0.2 = 0.05$. The probability that a ship will encounter a delay of 2 hours while sailing from one port to the following is thus $0.05 + 0.1225 + 0.05 = 0.2225$. In table 21 these probabilities are shown for all possible hours of delay.

Table 21: Probabilities of a delay for given hours between two ports

Delay	Prob
0	0.0400
1	0.1400
2	0.2225
3	0.2550
4	0.2025
5	0.1000
6	0.0400

7.1.3 Results

The results of the model with additional waypoints consist of two parts. The first part contains the results of the model where the waypoints are included and the second part contains the results of the model in which the waypoints are not included, but where the probabilities of delay of the model with waypoints are used. The last results are necessary to compare the model with waypoints to the model without waypoints.

Figure 6 shows the fraction of time with a certain delay in both all states and arriving states for the model with waypoints. Figure 7 shows the fraction of time for the model without waypoints, but where the probabilities of dealy are as shown in table 21. Again, ships are most of time in states with low delay. From the figures, it can be seen that the fraction of time that a ship is in a state with 0 delay is higher for the model without waypoints added than for the model with waypoints added when all states are considered. When only arriving states are considered, the fractions are about the same. The fraction of time that a ship is in states with delay 1-5 hours, is higher for the model with waypoints added. This can be explained in the following way. Delay costs are only associated to states when a ship is arriving in a port. Therefore, the amount of delay when a ship arrives in a port is more important than the amount of delay when a ship leaves a port or passes a waypoint. In the model with waypoints only one third of the states are arriving states. In the model without waypoints, half the states are arriving states. Therefore, it is better to compare the fraction of time with a certain delay when only the arriving states are considered. The fraction of time that a ship has no delay, when only arriving states are considered, is indeed higher for the model with waypoints added than for the model without waypoints. Because the addition of waypoints also reduces the total delay of the round tour, the fraction of time with 1-4 hours delay is higher for the model with waypoints added than for the model without waypoints. The last result holds both when all states are considered and when only arriving states are considered. An overview of all fractions and the steady state probabilities of all states are presented in tables 28, 29, 30, 31 and 32 in the appendix (section 10.1).

From the results shown, it is possible to compute the average delay in a state. The average delay is for both models about the same, namely almost 2.25 hours. Again the average delay when only arriving states are considered, is for both models higher than the average delay in a state when all states are considered. For the model with waypoints added, the difference is very small. The average delay in arriving states is also almost 2.25 hours. For the model without waypoints, the difference is a bit larger. In this model, the average delay in arriving states is almost 2.6 hours. Note that the delay costs of the model with waypoints are a lot lower than the costs of the model without waypoints, because in the model with waypoints the arriving states are only one third of the total states, where the arriving states in the model without waypoints are half the number of total states. Therefore, to calculate the delay costs, the average delay in arriving states of the model with waypoints has to be divided by three and multiplied by 100 (the costs of 1 hour delay). In the model without waypoints the average delay in arriving states has to be divided by two before multiplied by 100. The delay costs are reduced by more than 40% when adding waypoints to the model. Figures 8 and 9 show the average delay in each port for both the model with waypoints added and the model without waypoints added, but with the probabilities associated to the model with waypoints. In figure 8, the average delay is showed for arriving states, departing states and

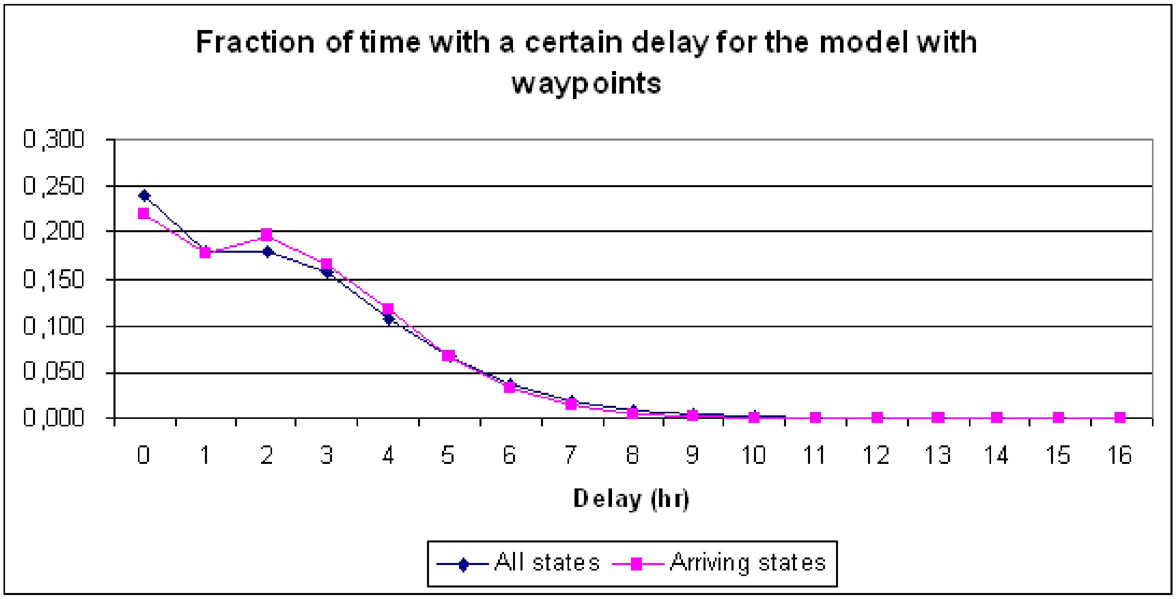


Figure 6: Fraction of time with a certain delay for the model with waypoints

waypoints. The average delay in waypoints is, in general, higher than the average delay in arriving and departing states. Since delay in waypoints does not lead to additional costs, this is as expected. Though, average delay in arriving states is again higher than average delay in departing states. The ports with high and low average delay are about the same as in the model for the real example.

In table 22 actions chosen in each state, when following the optimal policy in the model with waypoints, are shown. The I in the table means In; these columns show the actions chosen when a ship arrives in the associated port with the associated delay. Further, the O means Out; these columns denote the actions chosen when a ship leaves the associated port with the associated delay. Finally, the W means Waypoint; these columns denote the actions chosen when a ship passes the waypoint with the associated delay. The waypoint is located between the associated port and the next port. Table 23 shows the action chosen in each state, when following the optimal policy, if no waypoints are added. The total gain when sailing from Antwerp to Felixtowe is only one hour, so the gain when sailing from Antwerp to the waypoint is one hour and the gain when sailing from the waypoint to Felixtowe is 0 hours. This explains why action 2 is for most delays chosen while sailing from Antwerp to the waypoint, altough it is not chosen while sailing from the waypoint to Antwerp. Further, the difference in the action chosen when leaving the harbour does not differ very much from the action chosen when passing the waypoint. The largest difference in delay for which action 2 is first chosen is 2 hours. This occurs for the waypoint between Rotterdam and Lisbon. Before waypoints are added, it is optimal for ships to chose action 2, while sailing from Bremerhaven to Rotterdam when they have a delay of 2-7 and 11-16 hours when leaving Bremerhaven. When they have a delay of 0-1 or 8-10 hours, it is optimal to do nothing. After the waypoints are added, a unique delay is the boundary between two actions for every trip. From the data in the tables, it is possible to calculate the average costs of performing recovery action for

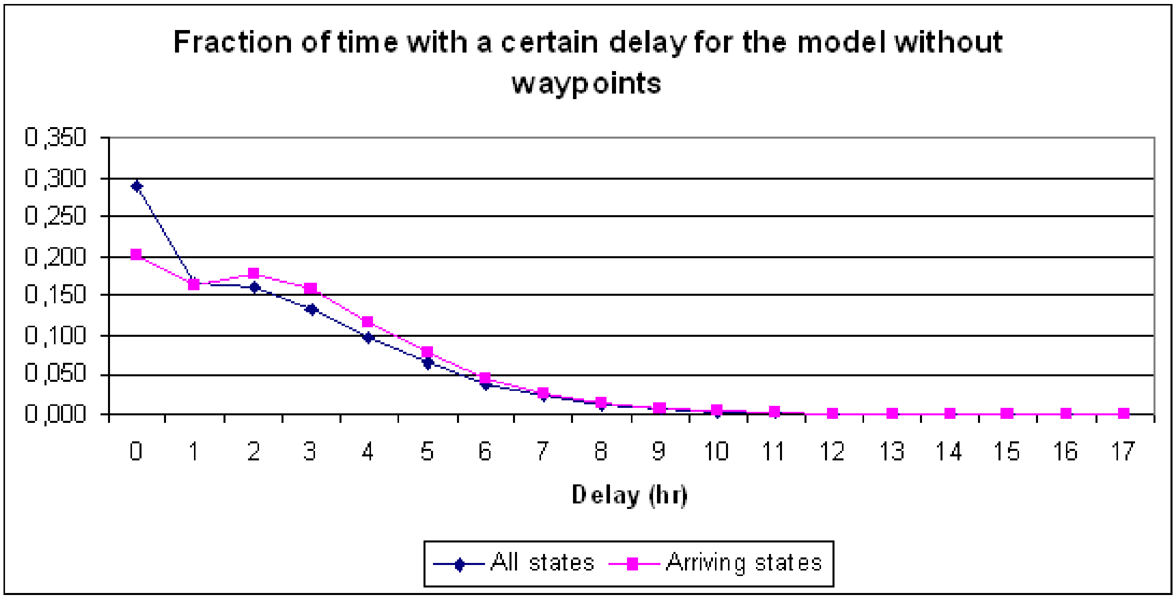


Figure 7: Fraction of time with a certain delay for the model without waypoints

one round tour. These costs are reduced by more than 30% when waypoints are added.

Furthermore, the total costs of one round tour are 295 before the addition of waypoints and 186 after the addition of waypoints. Therefore, the total costs are reduced by almost 37% when waypoints are added. The computation time of both models is less than 1 second. The model without waypoints has the same number of states as the model for the real example. Therefore, the number of variables and constraints, are also the same. The model with waypoints added has for each combination of port and delay 3 instead of 2 states. The number of states is thus multiplied by 1.5 and becomes 837. The number of variables and constraints also increases to 2512 and 839 respectively. From tables 28 and 29 in the appendix in section 10.1, it follows that the punctuality rises from 74.8% to 75.8% when waypoints are added to the model. When only the arriving states are considered, the punctuality rises from 70.2% to 76.0%, an increase of almost 8.3%.

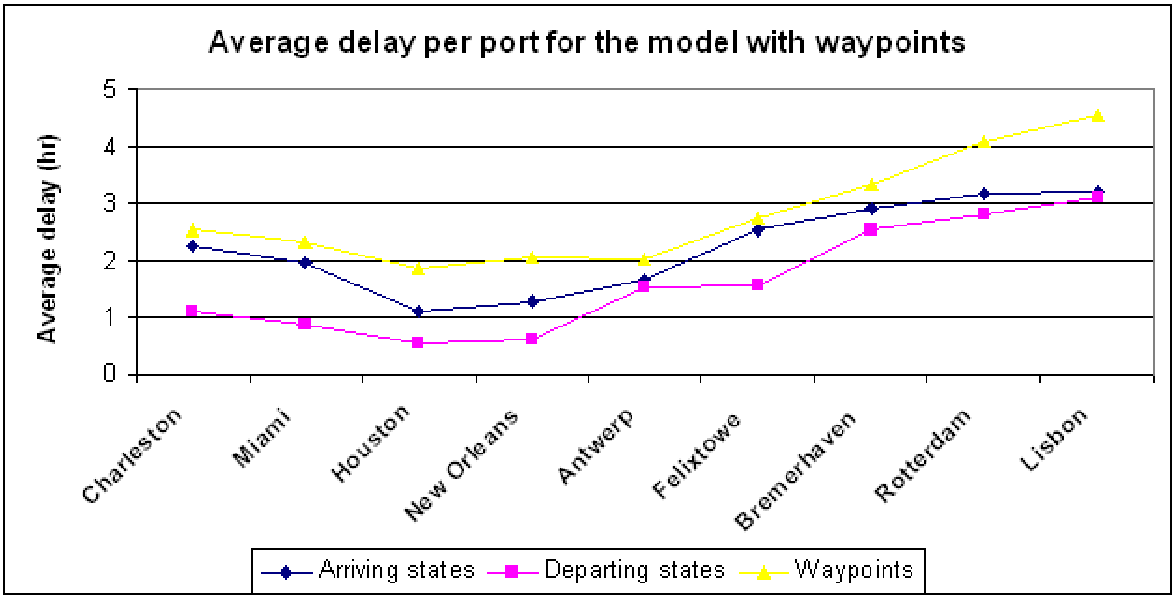


Figure 8: Average delay per harbour for the model with waypoints

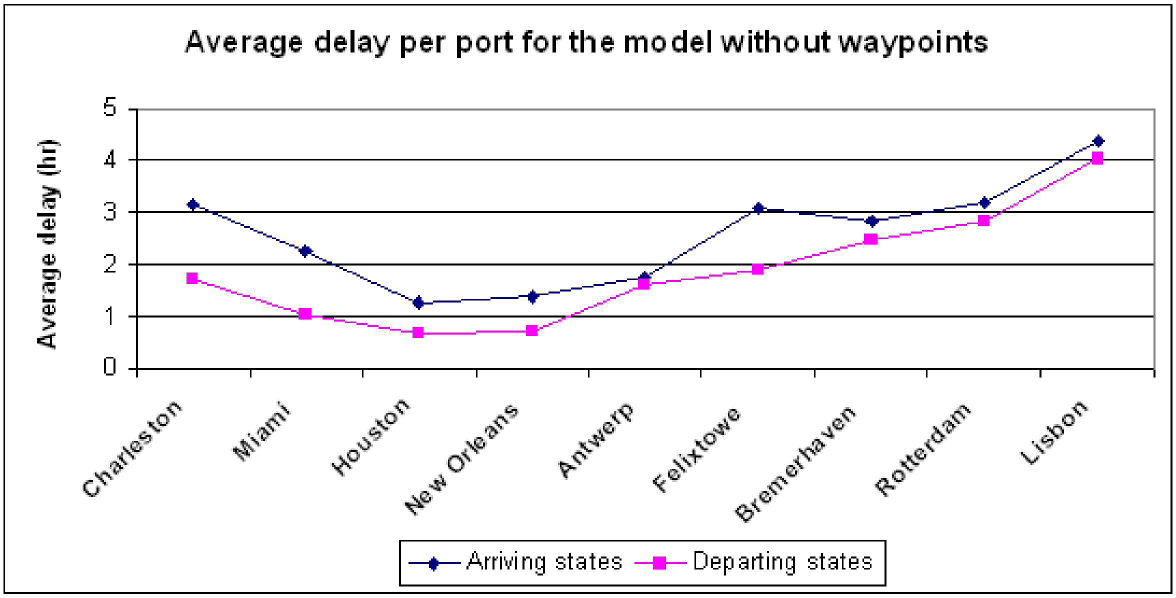


Figure 9: Average delay per harbour for the model without waypoints

Table 22: Best actions for the real example with waypoints added

	Charleston			Miami			Houston			New Orleans			Antwerp			Felixtowe			Bremerhaven			Rotterdam			Lisbon		
Delay	I	O	W	I	O	W	I	O	W	I	O	W	I	O	W	I	O	W	I	O	W	I	O	W	I	O	W
0	0	0	0	0	0	0	1	0	0	1	0	0	1	2	0	1	0	0	1	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	1	0	0	1	2	0	1	0	0	1	0	0	1	0	0	0	0	0
2	1	0	0	1	0	0	1	0	0	1	0	0	1	2	0	1	0	0	1	2	0	1	0	0	0	0	0
3	1	0	0	1	0	0	1	2	0	1	0	0	1	2	0	1	2	0	1	2	2	1	0	0	1	0	0
4	1	0	0	1	0	0	1	2	2	1	0	0	1	2	0	1	2	2	1	2	2	1	0	0	1	0	0
5	1	0	0	1	0	0	1	2	2	1	0	0	1	2	0	1	2	2	1	2	2	1	0	0	1	0	0
6	1	2	2	1	2	0	1	2	2	1	0	0	1	2	0	1	2	2	1	2	2	1	0	2	1	0	0
7	1	2	2	1	2	2	-	-	2	-	-	0	1	2	0	1	2	2	1	2	2	1	0	2	1	0	0
8	1	2	2	1	-	2	-	-	-	-	-	0	1	2	0	1	2	2	1	2	2	1	2	2	1	0	0
9	1	2	2	1	-	-	-	-	-	-	-	0	1	2	0	1	2	2	1	2	2	1	2	2	1	0	0
10	1	-	2	-	-	-	-	-	-	-	-	-	-	2	0	1	2	2	1	2	2	1	2	2	1	0	0
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	1	2	2	1	2	2	1	2	2	-	-	0
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	2	1	2	2	1	2	2	-	-	0
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	2	1	2	2	1	2	2	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1	2	2	1	2	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	2	1	2	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1	2	-	-	-	-

Table 23: Best actions for the real example with different probabilities

	Charleston		Miami		Houston		New Orleans		Antwerp		Felixtowe		Bremerhaven		Rotterdam		Lisbon	
Delay	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
2	1	0	1	0	1	0	1	0	1	2	1	2	1	2	1	0	1	0
3	1	0	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
4	1	0	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
5	1	2	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
6	1	2	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
7	1	2	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
8	1	2	1	2	1	2	1	0	1	2	1	2	1	0	1	2	1	0
9	1	2	1	2	1	2	1	0	1	2	1	2	1	0	1	2	1	0
10	1	2	1	2	1	2	-	-	1	2	1	2	1	0	1	2	1	0
11	1	2	1	2	-	-	-	-	1	2	1	2	1	2	1	2	1	0
12	1	2	1	-	-	-	-	-	-	2	1	2	1	2	1	2	1	0
13	1	2	1	-	-	-	-	-	-	-	1	2	1	2	1	2	-	0
14	1	2	-	-	-	-	-	-	-	-	1	2	1	2	1	2	-	-
15	1	-	-	-	-	-	-	-	-	-	-	-	1	2	1	2	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1	2	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	-	-

7.2 Reallocation of buffer times

The next extension discussed in this thesis is the reallocation of the buffer times in ports. Until now, the buffer time in a port is assumed to be shown. However, can changing the buffer times in ports increase the performance of liner shipping networks? And which allocation will lead to the best performance? A constraint on the new allocation is that the total buffer time in a round tour has to equal the total buffer time in a round tour with the initial allocation. To find the optimal allocation of the buffer time, the policy improvement algorithm, already discussed in section 5.4.2, is used. The algorithm needs to be adjusted for this purpose. This is performed in the following way:

Initialization: Set an initial policy R_1, for example the optimal policy shown the allocation of the buffer time. Set $n = 1$. Make all combinations between the set of harbours and the set of possibilities in a harbour (leaving and arriving). Call these combinations $l = 1, \ldots, l_{max}$, with l_{max} the number of possible port combinations.

Iteration n:

Step 1: *Value determination:* Using the allocation of the buffer time and which actions will be chosen, calculate $p_{ij}(k)$. Use $p_{ij}(k), C_i(k)$ and $v_M(R_n) = 0$ to solve the following system of $M + 1$ equalities

$$g(R_n) = C_i(k) + \sum_{j=0}^{M} p_{ij}(k) v_j(R_n) - v_i(R_n), \quad \text{for } i = 0, 1, \ldots, M.$$

This will show values for $g(R_n), v_0(R_n), v_1(R_n), \ldots, v_{M-1}(R_n).g(R_n)$ denotes the expected average cost per unit time of policy R_n. Further, $v_i(R_n)$ denote the effect on the total expected cost due to starting in state i (Hillier and Lieberman (2001)).

Step 2: *Policy improvement:* Using the current values of $v_i(R_n)$ computed for policy R_n, find the alternative policy R_{n+1} such that, for each state i, $d_i(R_{n+1}) = k$ is the decision that minimizes

$$C_i(k) + \sum_{j=0}^{M} p_{ij}(k) v_j(R_n) - v_i(R_n)$$

i.e., for *each* state i,

$$\min_{k \in K} [C_i(k) + \sum_{j=0}^{M} p_{ij}(k) v_j(R_n) - v_i(R_n)],$$

and then set $d_i(R_{n+1})$ equal to the minimizing value of k. This procedure defines a new policy R_{n+1}.

Optimality test: The current policy R_{n+1} is optimal if this policy is identical to policy R_n. If it is, set $l = 1$ and go to step 3. Otherwise, reset $n = n + 1$ and perform another iteration.

Step 3: *Compute steady state probabilities:* Compute the steady state probabilities π_i corresponding with the optimal policy found in step 2. This can be performed by solving the system of equalities

$$\pi_j = \sum_i p_{ij}(k)\pi_i$$

$$\sum_i \pi_i = 1.$$

Because the value of k is known for every i, the values of $p_{ij}(k)$ are also known for every combination of i and j.

Step 4: *Calculate savings:* Increase the buffer time of combination l with one hour and compute the expected average costs per unit time

$$g_{ld}(R_{n+1}, b+1) = C_{ld}(k) + \sum_{j=0}^{M} p_{ldj}(k)v_j(R_n) - v_i(R_n).$$

In this formulas the element $l \in L$ and $d \in D$ together are an element $i \in I$. Further $g_{ld}(R_{n+1}, b+1)$ denotes the expected value of $g(R_{n+1})$, where a buffer time of $b+1$ in stead of b hours in port combination l is used to calculate $p_{ldj}(k)$. Decrease the buffer time for combination l with one hour. Next, compute the expected savings per state by

$$E[S_{ld}(b+1)] = g(R_n) - g_{ld}(R_{n+1})$$

and the expected savings per port combination by

$$E[S_l(b+1)] = \sum_d \pi_{ld} E[S_{ld}(b+1)].$$

If $l < l_{max}$, set $l = l + 1$ and go back to step 4. Else go to step 5.

Step 5: *Calculate costs:* Decrease the buffer time of combination l with one hour and compute the expected average costs per unit time

$$g_{ld}(R_{n+1}, b-1)] = C_{ld}(k) + \sum_{j=0}^{M} p_{ldj}(k)v_j(R_n) - v_i(R_n).$$

The values of $p_{ldj}(k)$ should be calculated with a buffer of $b-1$ in stead of b hours in port combination l. Increase the buffer time for combination l with one hour. Next, compute the expected costs per state by

$$E[C_{ld}(b-1)] = g_{ld}(R_{n+1}, b-1) - g(R_n)$$

and the expected costs per port combination by

$$E[C_l(b-1)] = \sum_d \pi_{ld} E[C_{ld}(b-1)]].$$

If $l < l_{max}$, set $l = l + 1$ and go back to step 5. Else go to step 6.

Step 6: *Reallocation of the buffer time:* Find the port combinations l_{S1} and l_{S2} which generate respectively the highest and second highest savings when the buffer time is increased with one hour and the port combinations l_{C1} and l_{C2} which lead respectively to the lowest and second lowest extra costs when the buffer time is decreased with one hour. Only the port combinations where the buffer time is at least one hour has to be considered when searching for the lowest costs. Now, a few possibilities can arise

- If $l_{S1} \geq l_{C1}$
    - If $l_{S1} = l_{C1}$
        - If $l_{S1} - l_{C2} \geq l_{S2} - l_{C1}$
            - If $l_{S1} - l_{C2} \geq 0$, decrease the buffer time with one hour in port combination l_{C2} and increase the buffer time with one hour in port combination l_{S1} and go back to step 1.
            - Else stop.
        - Else
            - If $l_{S2} - l_{C1} \geq 0$, decrease the buffer time with one hour in port combination l_{C1} and increase the buffer time with one hour in port combination l_{S2} and go back to step 1.
            - Else stop.
    - Else, decrease the buffer time with one hour in port combination l_{C1} and increase the buffer time with one hour in port combination l_{S1} and go back to step 1.
- Else stop.

As initial policy the optimal policy with the presented allocation of the buffer time can be used, but each policy, which is chosen randomly, can be used. Furthermore, it is necessary to make a list of all possible port combinations. In step 1 and 2 the policy improvement is performed. Next, the steady state probabilities of the optimal policy are calculated in step 3. In step 4 and 5 the expected costs and savings for the policy are calculated for each possible combination made in the initialization. Note that the transition probabilities differ for each combination, because the transition probabilities are dependent on the buffer time. Therefore, they have to be calculated for each combination separately. In step 6 one hour of buffer time is switched between two port combinations, but only if the switch is profitable.

7.2.1 Optimality

An important question is whether the method to compute the best buffer times leads to the optimal allocation of the buffer times or not. The first question, which can be considered is: Is it possible that a possible switch of more than one hour buffer time can be made, when no possible switch of one hour buffer time can be made? This does not seem very logical, because switches of 1 hour are made. Therefore, when it is not profitable to make a switch, it does not seem possible to find a combination of 2 or more consecutive switches in such a way that the combination of switches is profitable. This problem would rather occur when switches of more than one hour are made in one step. It could then happen that no profitable switch can be made, but a profitable switch of only one hour could be made. Though, this question cannot be answered with certainty, because the savings of increasing and the cost of decreasing the buffer time with one hour in a port combination are calculated with the data obtained from the previous policy. The real savings and costs should be calculated with the data obtained from the optimal policy, with the new buffer times. This is not implemented in the algorithm, because the optimal policy and the corresponding values of v_i and π_i of all combinations of possible new buffer should be computed in advance. This would take very much time.

A second question is: Is it possible that a switch will be made in one step and the best switch in the next step is the reverse of the previous switch? Therefore, if the switch in step 1 is for example increase 2, decrease 3, is it possible that the switch in step 2 is increase 3, decrease 2? This question can be answered, because an example of this can be found (see appendix C, section 10.3). Therefore, this is indeed possible. This problem could be solved by checking whether this happens or not in the algorithm. If it happens, the switch, which will lead to the highest total savings, is made and the corresponding distribution will be seen as the optimal distribution of the buffer time. Again, it is not sure whether this will lead to the optimal distribution. Therefore, it is hard to say something about the optimality of the algorithm.

7.2.2 Results

The algorithm that is described above is implemented in MATLAB®. In table 24 the best buffer times, which are found by performing the algorithm, can be found. The second column denotes the buffer times of arriving in port and the last column shows the buffer times of leaving the port. The buffer times while leaving are lower than the buffer times while arriving in a port. This is probably because it is costless to depart with a delay, so it is better to have more buffer when arriving, since costs are associated to arriving with a delay.

Table 24: Best buffer times of the real example

Harbour	Buffer in (hr)	Buffer out (hr)
Charleston	2	0
Miami	3	0
Houston	2	0
New Orleans	3	2
Antwerp	4	1
Felixtowe	3	0
Bremerhaven	2	1
Rotterdam	3	1
Lisbon	3	2

Figure 10 shows the fraction of time that a ship is in a state with a certain delay. These results can be compared with the results in section 6.2. Then, it follows that the fraction of time that a ship is in a state without delay is increased in both cases. An overview of all fractions of delay and steady state probabilities can be found in the appendix (section 10.1), in tables 33 and 34. The difference between the fraction of time that a ship is in a state with a certain delay, when all states are considered and when only arriving states are considered, is much smaller for the model with the new distributed buffer times, than it was for the initial model. Comparing the results of the model to the original distribution of the buffer times and the model with the new distribution of the buffer times, leads to the conclusion conclusion that introducing the new buffer times decreases the amount of delay. Figure 11 shows the average delay in each harbour for both arriving states and departing states, after the new distribution of the buffer times is implemented. The average delay of the new model is almost 2.7 hours when all states are considered and over 2.7 hours when only arriving states are considered. In the original model the average delay was almost 3.3 and 3.7 hours respectively. This leads to a decrease of the delay costs of almost $\frac{3.7-2.7}{2} \cdot 100 = 50$ dollars, which is a reduction of about 26%. Furthermore, the average delay in all states of the model is reduced by 18.4%. Furthermore, it can be noted that the difference in average delay between harbours is much smaller when the new buffer time is implemented than when the initial buffer times are used.

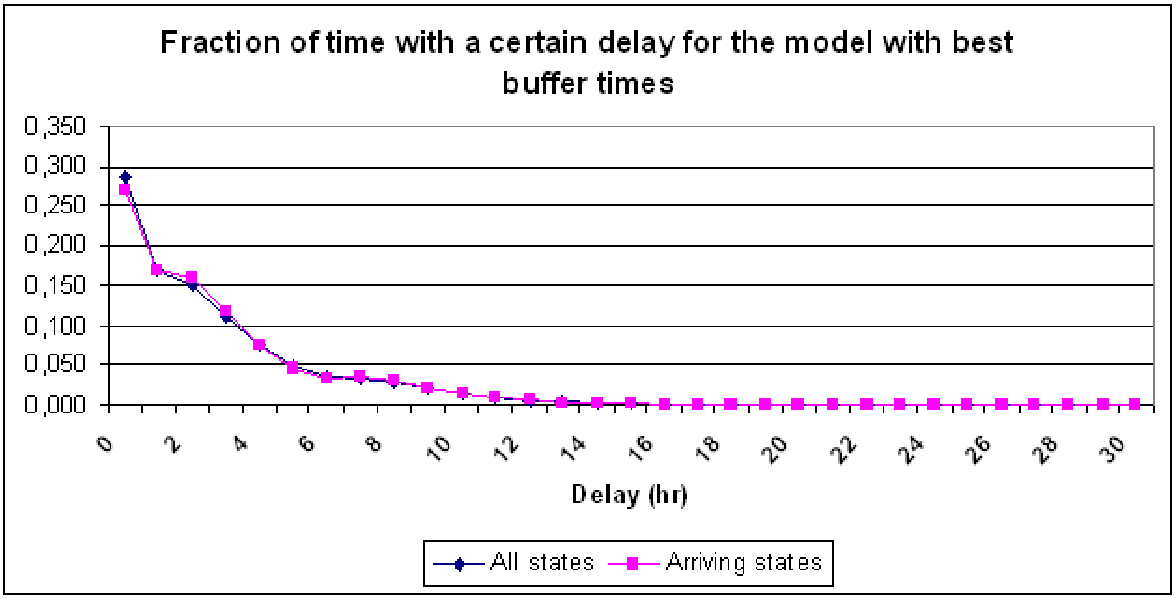


Figure 10: Fraction of time with a certain delay for the model with best buffer times

Table 25 presents the optimal policy found by the algorithm. When comparing this policy to the policy shown in table 19, it can be seen that the structure of both policies looks about the same. Note that in the first and last ports, recovery actions are performed with a little lower delay with the new distribution of the buffer times than with the original distribution, where in the middle ports, recovery actions are taken with a little higher delay with the new distribution. With these policies, the reduction of recovery costs can be computed, which turns out to be a reduction of almost 5.8%.

The total costs of one round tour of this model are $ 402. The costs were $ 467 in the original model, so they are reduced by almost 14%. The computation time of the algorithm is about one minute. The punctuality of the model is increased from 63.9% with the initial

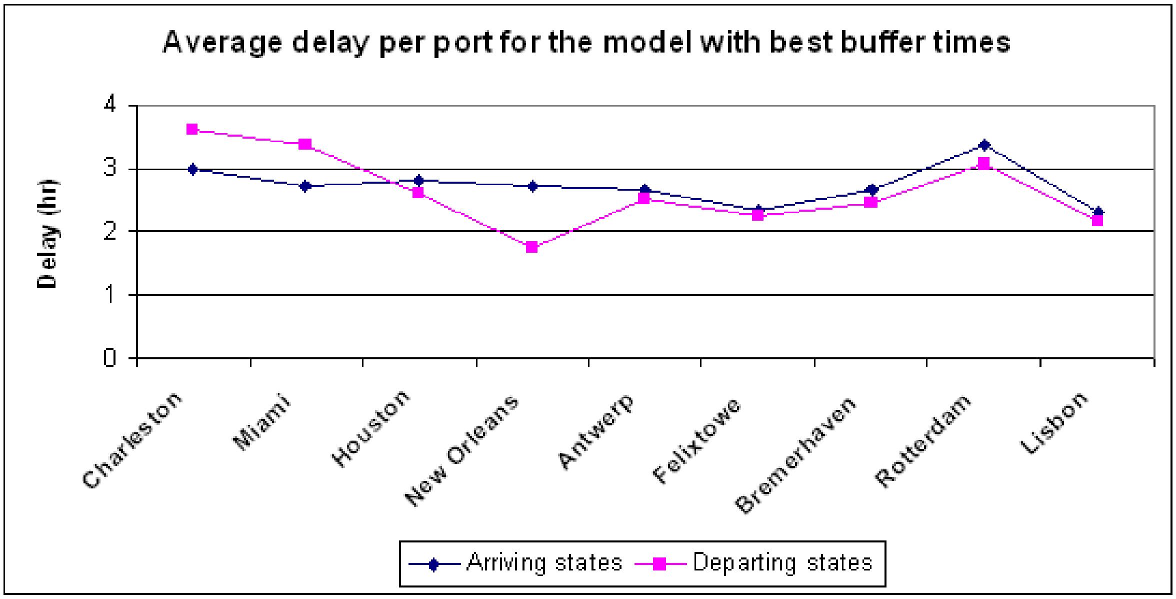


Figure 11: Average delay per harbour for the model with best buffer times

distribution of the buffer times to 71.6% with the new allocation when all states are considered. When only arriving states are considered, the punctuality is increased from 60.0% to 71.7%. This is an increase of 19.5%.

Table 25: Best actions for the real example with best buffer

	Charleston		Miami		Houston		New Orleans		Antwerp		Felixtowe		Bremerhaven		Rotterdam		Lisbon	
Delay	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
2	1	2	1	0	1	0	1	0	1	0	1	0	1	2	1	0	1	0
3	1	2	1	0	1	2	1	0	1	0	1	0	1	2	1	0	1	0
4	1	2	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
5	1	2	1	0	1	2	1	0	1	2	1	2	1	2	1	0	1	0
6	1	2	1	0	1	2	1	0	1	2	1	2	1	2	1	2	1	0
7	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
8	1	2	1	2	1	2	1	0	1	2	1	2	1	0	1	2	1	0
9	1	2	1	2	1	2	1	0	1	2	1	2	1	0	1	2	1	0
10	1	2	1	2	1	2	1	0	1	2	1	2	1	0	1	2	1	0
11	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
12	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
13	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
14	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
15	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
16	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
17	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	0
18	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	1	2
19	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	0	2
20	1	2	1	2	1	2	1	0	1	2	1	2	1	2	1	2	0	2
21	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	0	2
22	1	2	1	2	1	2	1	-	1	2	1	2	1	2	1	2	-	2
23	1	2	1	2	-	-	-	-	1	2	1	2	1	2	1	2	-	2
24	1	2	1	2	-	-	-	-	1	2	1	2	1	2	1	2	0	2
25	-	2	1	2	-	-	-	-	1	2	1	2	1	2	1	0	0	2
26	-	-	1	2	-	-	-	-	-	-	1	2	1	0	1	0	0	2
27	-	-	-	2	-	-	-	-	-	-	1	0	1	0	1	0	0	2
28	-	-	-	-	-	-	-	-	-	-	-	-	1	0	0	0	0	2
29	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	2
30	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	2

8 Conclusion

In liner shipping networks ships encounter problems when they are delayed. Since companies buy time slots and space in harbours, 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.

This thesis investigates the feasibility of Markov decision theory to perform a model that can determine which action should be performed in case of a known position and known delay of a ship.

A Markov decision model is very useful to determine the optimal recovery policy in liner shipping networks. States can be defined in such a way that an action has to be chosen after each transition. When the states are properly defined, it is possible to write the problem as a linear programming problem. This problem can be implemented in Aimms® and the optimal solution can be computed. Furthermore, the computation time of the model is less than one second for all models considered in this thesis.

The optimal policies show that the recovery action of increasing the sailing speed of a ship is not chosen when the distance between two ports is very large. When the distance between ports is a bit smaller, it can be optimal to chose the recovery action that increases the sailing speed, but only if the obtained delay is large. Furthermore, in port combinations just before a port with a large distance to the next port no recovery actions are performed when the delay is high. This is because the delay can only be reduced with a few hours in a short distance, while it can be reduced with many hours while sailing a large distance between the two ports, which come soon after these port combinations. Besides, the amount of delay for which it is optimal to perform a recovery action is dependent on the buffer time in the associated port combination.

The optimal recovery policy found can be improved by performing one of the two (or both) extensions. The first extension discussed is the addition of waypoints between two ports. The average delay in the model is not affected by the addition of waypoints. However, the costs of one round tour are reduced. The delay costs are reduced by more than 40%. Furthermore, the recovery costs are reduced by more than 30%. This leads to a reduction of almost 37% of the total costs. Furthermore, the punctuality increases with about 8.3% to 76%.

The other extension discussed is the reallocation of the buffer times. The optimality of this method is not guaranteed, but the methods leads to a reduction of both delay and costs. The average delay is reduced by 18.4%. Furthermore, the delay costs are reduced by 26% and the recovery costs are reduced by almost 5.8%. This leads to a total reduction of almost 14%. Finally, the punctuality increases with 19.5%, from 60.0% to 71.7% (when only arriving states are considered).

9 Discussion

Because little time is needed to solve the Markov decision model, it is a useful model in practice. When using the model, it is important that reliable data are available. Not all data in this thesis are very reliable. The probabilities of delay, for example, are assumed the same for each trip between two ports. This is not realistic, because it is expected that some factors, including distances, bottlenecks, mechanical problems and climate related factors, exists, which influence the probabilities of an additional delay. Before this model is used in practice, it is recommended to perform further research to the probabilities of delay of the different trips.

In this thesis, only two recovery actions, increasing the sailing speed of the ship and increasing the port handling capacity, are considered. It is possible to add as much recovery actions as desired. Furthermore, it is possible to consider more levels of for example speed increase. When a ship has to sail a large distance to the next port, it is probably more profitable to increase the sailing speed with a much lower amount than when a ship has to sail a small distance.

In this thesis, exactly one waypoint is added between two ports. This is performed to investigate the effect of adding waypoints. The addition of waypoints seems to be an effective method to reduce the costs. It can be considered to add more waypoints between some ports (if the distance is large) and add no waypoints between other ports (when the distance is small). The method described in this thesis remains the same. Before the waypoints can be added, it is recommended to investigate what the probabilities of delay are on the different parts of trips.

Finally, the distribution of the buffer times can be changed to reduce both delay and costs. The algorithm used to reallocate the buffer time indeed leads to a reduction in delay and costs. The optimality of the algorithm, however, cannot be proved. It is recommended to do some further research to determine the optimality of the reallocation of the buffer times.

References

M. Christiansen, K. Fagerholt, and D. Ronen. Ship routing and scheduling: Status and perspectives. *Transportation Science*, 38(1):1–18, 2004.

J. Clausen, J.J. Groth, M. Hofman, J. Larsen, and L. Madsen. Robustness and recovery in train scheduling - a simulation study from dsb s-tog a/s. ATMOS 2006 - 6th Workshop on Algorithmic Methods and Models for Optimization of Railways, 2006.

R. Dekker. Developing robust shipping networks. presentation.

R. Dekker, L.G. Kroon, 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.

E. Gkanatsas. Designing robust shipping schedules. Master's thesis, Erasmus University Rotterdam, 2005.

D. Goldsman, E.L. Johnson, A.J. Kleywegt, G'L' Nemhauser, J.M. Rosenberger, and A.J. Schaefer. Simair: a stochastic model of airline operations. *Simulation Conference Proceedings*, 2:1118–1122, 2000.

F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research*, pages 1053–1078. McGraw-Hill Education, seventh edition, 2001. ISBN ISBN 0-07-246121-7.

T.E. Notteboom. The time factor in liner shipping services. *Maritime Economics & Logistics*, 8:19–39, 2006.

P. Ong. Block exemption for liner shipping agreements, chapter 50b competition act. AG/LEG/SL/50B/2005/7, 2006.

D. Ronen. Ship scheduling: The last decade. *European Journal of Operational Research*, 71: 325–333, 1993.

`http://www2.nykline.com/liner/service_network/eu2.html`. An example of a ship string on the website of the NYK Line, retrieved July 15, 2009.

`http://www.haugenconsulting.com/whatisdemurrage.asp`. Haugen Consulting LLC, retrieved July 15, 2009.

`www.aimms.com/`. Website of the mathmatical language Aimms®, retrieved July 15, 2009.

`www.mathworks.com/`. Website of the numerical computer program MATLAB®, retrieved July 15, 2009.

10 Appendices

10.1 Appendix A: Tables with results

In this appendix the fractions of time that a ship is in a certain state and the steady state probabilities of all states of the different models are shown. The left section of table 26 shows the fraction of time with a certain delay of the reals example. The right section shows the fraction of time that a ship is in a state with a delay of less than or equal to the associated delay. Tables 27 denotes the steady state probabilities of the real example. Next, tables 28, 30 and 31 show the same of the model with added waypoints. Tables 29 and 32 denote the steady state probabilities of the model without waypoints added, but where the probabilities of delay used are as shown in table 21. Finally, tables 33 and 34 show the fractions of time and the steady state probabilities of the real example with the best distribution of the buffer time.

Table 26: Fraction of time with a certain delay for the real example

Delay	Fraction	Cum Fraction
0	0.250	0.250
1	0.135	0.386
2	0.135	0.520
3	0.119	0.639
4	0.086	0.725
5	0.060	0.786
6	0.046	0.831
7	0.037	0.868
8	0.036	0.904
9	0.029	0.934
10	0.021	0.954
11	0.015	0.969
12	0.010	0.979
13	0.007	0.985
14	0.004	0.990
15	0.003	0.993
16	0.002	0.995
17	0.002	0.997
18	0.001	0.998
19	0.001	0.999
20	0.000	0.999
21	0.000	0.999
22	0.000	1.000
23	0.000	1.000
24	0.000	1.000
25	0.000	1.000
26	0.000	1.000
27	0.000	1.000
28	0.000	1.000
29	0.000	1.000
30	0.000	1.000

Delay	Fraction	Cum Fraction
0	0.167	0.167
1	0.135	0.302
2	0.155	0.457
3	0.143	0.600
4	0.098	0.697
5	0.066	0.763
6	0.047	0.810
7	0.038	0.848
8	0.043	0.891
9	0.035	0.925
10	0.023	0.949
11	0.017	0.966
12	0.011	0.976
13	0.008	0.984
14	0.005	0.989
15	0.003	0.992
16	0.003	0.995
17	0.002	0.997
18	0.001	0.998
19	0.001	0.999
20	0.000	0.999
21	0.000	0.999
22	0.000	1.000
23	0.000	1.000
24	0.000	1.000
25	0.000	1.000
26	0.000	1.000
27	0.000	1.000
28	0.000	1.000
29	0.000	1.000
30	0.000	1.000

Table 27: Steady state probabilities of the real example

	Charleston		Miami		Houston		New Orleans		Antwerp		Felixtowe		Bremerhaven		Rotterdam		Lisbon	
Delay	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
0	0.012	0.023	0.009	0.028	0.021	0.034	0.017	0.032	0.011	0.015	0.002	0.013	0.002	0.007	0.003	0.007	0.008	0.008
1	0.006	0.009	0.009	0.011	0.012	0.007	0.011	0.008	0.009	0.009	0.006	0.007	0.006	0.006	0.005	0.005	0.004	0.005
2	0.006	0.005	0.012	0.005	0.009	0.005	0.012	0.006	0.012	0.008	0.007	0.007	0.008	0.007	0.007	0.006	0.005	0.008
3	0.006	0.004	0.010	0.003	0.004	0.003	0.008	0.003	0.009	0.007	0.010	0.007	0.010	0.007	0.008	0.006	0.007	0.008
4	0.005	0.003	0.005	0.002	0.002	0.002	0.002	0.001	0.004	0.005	0.008	0.005	0.009	0.007	0.007	0.006	0.007	0.006
5	0.004	0.003	0.003	0.001	0.002	0.001	0.001	0.001	0.002	0.002	0.005	0.003	0.005	0.005	0.005	0.004	0.006	0.006
6	0.004	0.002	0.002	0.001	0.001	0.001	0.001	0.001	0.001	0.002	0.003	0.003	0.003	0.003	0.003	0.003	0.006	0.005
7	0.003	0.002	0.001	0.001	0.003	0.001	0.000	0.001	0.001	0.001	0.002	0.002	0.002	0.003	0.002	0.003	0.004	0.004
8	0.002	0.001	0.003	0.001	0.001	0.001	0.003	0.001	0.003	0.002	0.002	0.002	0.002	0.002	0.003	0.003	0.003	0.002
9	0.002	0.001	0.001	0.000	0.000	0.000	0.001	0.000	0.001	0.002	0.003	0.002	0.003	0.002	0.003	0.003	0.002	0.002
10	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.002	0.001	0.002	0.002	0.003	0.002	0.001	0.001
11	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.002	0.001	0.001	0.001	0.002	0.002	0.001	0.001
12	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
13	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.001	0.001	0.001	0.001	0.000
14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000
16	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000
17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.000	0.000	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.000	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
21	0.000	0.000	0.000	0.000	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22	0.000	0.000	0.000	0.000	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23	0.000	0.000	0.000	0.000	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
24	0.000	0.000	0.000	0.000	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25	0.000	0.000	0.000	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
26	0.000	-	0.000	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
27	0.000	-	-	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
28	-	-	-	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
29	-	-	-	-	-	-	-	-	-	-	0.000	-	0.000	0.000	0.000	0.000	0.000	0.000
30	-	-	-	-	-	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000

Table 28: Fraction of time with a certain delay for the real example with extra waypoints

Delay	Fraction	Cum Fraction
0	0.240	0.240
1	0.181	0.421
2	0.181	0.601
3	0.157	0.758
4	0.107	0.866
5	0.066	0.932
6	0.036	0.968
7	0.019	0.987
8	0.008	0.995
9	0.003	0.998
10	0.001	1.000
11	0.000	1.000
12	0.000	1.000
13	0.000	1.000
14	0.000	1.000
15	0.000	1.000
16	0.000	1.000

Delay	Fraction	Cum Fraction
0	0.220	0.220
1	0.179	0.399
2	0.196	0.594
3	0.166	0.760
4	0.118	0.878
5	0.067	0.945
6	0.033	0.977
7	0.014	0.992
8	0.005	0.997
9	0.002	0.999
10	0.001	1.000
11	0.000	1.000
12	0.000	1.000
13	0.000	1.000
14	0.000	1.000
15	0.000	1.000
16	0.000	1.000

Table 29: Fraction of time with a certain delay for the real example with probabilities corresponding to the model with extra waypoints

Delay	Fraction	Cum Fraction
0	0.290	0.290
1	0.165	0.454
2	0.160	0.614
3	0.134	0.748
4	0.098	0.846
5	0.065	0.911
6	0.040	0.951
7	0.023	0.974
8	0.013	0.987
9	0.007	0.994
10	0.003	0.997
11	0.002	0.999
12	0.001	1.000
13	0.000	1.000
14	0.000	1.000
15	0.000	1.000
16	0.000	1.000
17	0.000	1.000

Delay	Fraction	Cum Fraction
0	0.203	0.203
1	0.162	0.365
2	0.178	0.544
3	0.158	0.702
4	0.118	0.819
5	0.077	0.896
6	0.047	0.942
7	0.027	0.969
8	0.016	0.985
9	0.008	0.993
10	0.004	0.997
11	0.002	0.999
12	0.001	1.000
13	0.000	1.000
14	0.000	1.000
15	0.000	1.000
16	0.000	1.000
17	0.000	1.000

Table 30: Steady state probabilities of first five ports of the model with waypoints

	Charleston			Miami			Houston			New Orleans			Antwerp		
Delay	In	Out	Way	In	Out	Way	In	Out	Way	In	Out	Way	In	Out	Way
0	0.01	0.017	0.003	0.009	0.018	0.004	0.017	0.026	0.005	0.012	0.024	0.005	0.011	0.014	0.009
1	0.006	0.009	0.008	0.007	0.010	0.008	0.008	0.005	0.011	0.010	0.006	0.010	0.008	0.008	0.007
2	0.006	0.006	0.009	0.009	0.006	0.009	0.006	0.003	0.010	0.008	0.004	0.009	0.008	0.006	0.008
3	0.005	0.002	0.008	0.006	0.002	0.009	0.004	0.002	0.008	0.005	0.002	0.008	0.005	0.004	0.005
4	0.004	0.001	0.004	0.004	0.001	0.004	0.002	0.001	0.002	0.002	0.001	0.003	0.003	0.003	0.004
5	0.003	0.001	0.003	0.002	0.000	0.002	0.001	0.000	0.001	0.000	0.000	0.002	0.001	0.001	0.002
6	0.002	0.000	0.001	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.001
7	0.001	0.000	0.001	0.000	0.000	0.000	-	-	0.000	-	-	0.000	0.000	0.000	0.001
8	0.000	0.000	0.000	0.000	-	0.000	-	-	-	-	-	0.000	0.000	0.000	0.000
9	0.000	0.000	0.000	0.000	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000
10	0.000	-	0.000	-	-	-	-	-	-	-	-	-	-	0.000	0.000
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.000
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 31: Steady state probabilities of the last four ports of the model with waypoints

	Felixtowe			Bremerhaven			Rotterdam			Lisbon		
Delay	In	Out	Way	In	Out	Way	In	Out	Way	In	Out	Way
0	0.007	0.016	0.003	0.003	0.007	0.001	0.002	0.007	0.001	0.002	0.003	0.001
1	0.006	0.006	0.007	0.005	0.006	0.005	0.005	0.005	0.003	0.005	0.005	0.002
2	0.007	0.005	0.008	0.008	0.006	0.007	0.008	0.006	0.005	0.006	0.005	0.004
3	0.006	0.004	0.008	0.009	0.006	0.008	0.008	0.006	0.006	0.008	0.007	0.005
4	0.004	0.003	0.005	0.006	0.005	0.006	0.007	0.005	0.006	0.007	0.008	0.006
5	0.003	0.002	0.003	0.004	0.003	0.004	0.003	0.004	0.006	0.005	0.004	0.006
6	0.002	0.001	0.001	0.001	0.002	0.003	0.002	0.002	0.004	0.003	0.003	0.005
7	0.001	0.000	0.001	0.001	0.001	0.001	0.001	0.001	0.003	0.001	0.001	0.004
8	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.002	0.000	0.000	0.002
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.001
10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	0.000
12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	0.000
13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	-
14	-	-	0.000	0.000	0.000	0.000	0.000	0.000	-	-	-	-
15	-	-	-	-	0.000	0.000	0.000	0.000	-	-	-	-
16	-	-	-	-	-	0.000	0.000	0.000	-	-	-	-

Table 32: Steady state probabilities of the model without waypoints added

	Charleston		Miami		Houston		New Orleans		Antwerp		Felixtowe		Bremerhaven		Rotterdam		Lisbon	
Delay	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
0	0.012	0.021	0.011	0.026	0.024	0.038	0.019	0.036	0.016	0.020	0.004	0.019	0.006	0.012	0.005	0.011	0.003	0.005
1	0.006	0.012	0.010	0.014	0.011	0.007	0.013	0.009	0.012	0.012	0.008	0.009	0.009	0.009	0.008	0.008	0.004	0.004
2	0.007	0.008	0.011	0.009	0.009	0.005	0.011	0.006	0.011	0.009	0.011	0.009	0.012	0.009	0.011	0.009	0.006	0.007
3	0.007	0.005	0.009	0.004	0.006	0.003	0.007	0.003	0.008	0.007	0.011	0.007	0.011	0.009	0.011	0.009	0.008	0.009
4	0.006	0.004	0.007	0.002	0.003	0.001	0.003	0.001	0.005	0.004	0.009	0.005	0.008	0.007	0.009	0.007	0.008	0.007
5	0.005	0.003	0.004	0.001	0.001	0.001	0.001	0.000	0.002	0.002	0.006	0.003	0.005	0.005	0.006	0.005	0.008	0.007
6	0.004	0.002	0.002	0.000	0.001	0.000	0.000	0.000	0.001	0.001	0.003	0.002	0.002	0.003	0.003	0.003	0.007	0.006
7	0.003	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.001	0.001	0.001	0.002	0.002	0.005	0.004
8	0.002	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.001	0.001	0.001	0.003	0.003
9	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.002
10	0.001	0.000	0.000	0.000	0.000	0.000	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001
11	0.000	0.000	0.000	0.000	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12	0.000	0.000	0.000	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
13	0.000	0.000	0.000	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	-	0.000
14	0.000	0.000	-	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	-	-
15	0.000	-	-	-	-	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	0.000	0.000	0.000	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.000	0.000	-	-

Table 33: Fraction of time with a certain delay for the real example with the best buffer times

Delay	Fraction	Cum Fraction
0	0.287	0.287
1	0.168	0.455
2	0.149	0.605
3	0.111	0.716
4	0.075	0.791
5	0.049	0.840
6	0.035	0.875
7	0.034	0.909
8	0.029	0.938
9	0.022	0.960
10	0.015	0.975
11	0.009	0.984
12	0.006	0.990
13	0.004	0.994
14	0.002	0.996
15	0.001	0.997
16	0.001	0.998
17	0.001	0.999
18	0.000	0.999
19	0.000	1.000
20	0.000	1.000
21	0.000	1.000
22	0.000	1.000
23	0.000	1.000
24	0.000	1.000
25	0.000	1.000
26	0.000	1.000
27	0.000	1.000
28	0.000	1.000
29	0.000	1.000
30	0.000	1.000

Delay	Fraction	Cum Fraction
0	0.269	0.269
1	0.169	0.438
2	0.160	0.598
3	0.118	0.717
4	0.075	0.792
5	0.045	0.837
6	0.033	0.870
7	0.037	0.907
8	0.031	0.938
9	0.023	0.961
10	0.015	0.975
11	0.009	0.985
12	0.006	0.991
13	0.004	0.994
14	0.002	0.996
15	0.001	0.997
16	0.001	0.998
17	0.001	0.999
18	0.000	1.000
19	0.000	1.000
20	0.000	1.000
21	0.000	1.000
22	0.000	1.000
23	0.000	1.000
24	0.000	1.000
25	0.000	1.000
26	0.000	1.000
27	0.000	1.000
28	0.000	1.000
29	0.000	1.000
30	0.000	1.000

Table 34: Steady state probabilities of the real example with best buffer times

	Charleston		Miami		Houston		New Orleans		Antwerp		Felixtowe		Bremerhaven		Rotterdam		Lisbon	
Delay	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
0	0.014	0.010	0.017	0.011	0.013	0.016	0.011	0.026	0.016	0.019	0.022	0.021	0.015	0.018	0.009	0.014	0.018	0.019
1	0.009	0.007	0.009	0.007	0.008	0.010	0.010	0.008	0.011	0.010	0.009	0.012	0.010	0.010	0.009	0.009	0.009	0.011
2	0.009	0.009	0.008	0.010	0.008	0.007	0.012	0.007	0.009	0.007	0.006	0.006	0.009	0.007	0.011	0.008	0.009	0.007
3	0.006	0.007	0.006	0.007	0.007	0.006	0.009	0.005	0.005	0.005	0.004	0.004	0.006	0.006	0.009	0.007	0.007	0.006
4	0.004	0.006	0.003	0.005	0.006	0.005	0.005	0.003	0.003	0.003	0.003	0.003	0.004	0.004	0.004	0.005	0.005	0.004
5	0.003	0.004	0.002	0.004	0.004	0.004	0.002	0.002	0.002	0.003	0.002	0.002	0.002	0.003	0.002	0.002	0.003	0.003
6	0.002	0.003	0.002	0.002	0.003	0.003	0.001	0.001	0.002	0.002	0.003	0.002	0.001	0.002	0.001	0.002	0.002	0.002
7	0.003	0.002	0.002	0.002	0.002	0.002	0.001	0.001	0.003	0.002	0.002	0.002	0.003	0.002	0.001	0.002	0.002	0.001
8	0.002	0.002	0.002	0.002	0.001	0.001	0.002	0.001	0.002	0.002	0.001	0.001	0.002	0.002	0.003	0.001	0.001	0.001
9	0.001	0.002	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.002	0.002	0.001	0.001
10	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.000	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.000
11	0.001	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000
12	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000
13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
16	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
21	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23	0.000	0.000	0.000	0.000	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
24	0.000	0.000	0.000	0.000	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25	-	0.000	0.000	0.000	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
26	-	-	0.000	0.000	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
27	-	-	-	0.000	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
28	-	-	-	-	-	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	0.000	0.000
29	-	-	-	-	-	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	-	0.000
30	-	-	-	-	-	-	-	-	-	-	-	-	0.000	0.000	0.000	0.000	-	0.000

10.2 Appendix B: Splitting discrete distributions

In this appendix, the splitting of discrete distributions is investigated. The central question in this appendix is: Is it possible to split a known discrete distribution in two equal discrete distributions in such a way that the sum of these two distributions equals the known distribution? First, a general approach will be used. Next, some notes will be made using small examples. Finally, the central question will be answered and a conclusion will be drawn.

Suppose a discrete distribution X is given by $P(X = i) = p_i$. Now, the distribution X_1 and X_2, such that $X = X_1 + X_2$, are tried to be found. In this section it is assumed that $X_1 = X_2$. This assumption makes the problem a bit easier, because it reduces the number of unknowns by half. Define $P(X_1 = i) = P(X_2 = i) = q_i$. It is now possible to express the values of p_i in terms of q_i. With these expressions the values of q_i can be calculated when the values of p_i are known. For $i = 0$ this gives

$$p_0 = P(X = 0) = P(X_1 = 0, X_2 = 0) = q_0^2.$$

Thus,

$$q_0 = \sqrt{p_0}.$$

The same can be performed for $i = 1$:

$$p_1 = P(X = 1) = P(X_1 = 1, X_2 = 0) + P(X_1 = 0, X_2 = 1) = q_1 q_0 + q_0 q_1 = 2q_0 q_1.$$

Therefore,

$$q_1 = \frac{p_1}{2q_0}.$$

For $i = 2$ this leads to

$$p_2 = P(X = 2) = P(X_1 = 2, X_2 = 0) + P(X_1 = 1, X_2 = 1) + P(X_1 = 0, X_2 = 2)$$

$$p_2 = q_2 q_0 + q_1^2 + q_0 q_2 = 2q_0 q_2 + q_1^2.$$

Thus,

$$q_2 = \frac{p_2 - q_1^2}{2q_0}.$$

In general,

$$p_n = P(X = n) = q_n q_0 + q_{n-1} q_1 + \ldots + q_0 q_n = 2q_0 q_n + \sum_{j=1}^{n-1} q_j q_{n-j}.$$

Thus,

$$q_n = \frac{1}{2q_0}\left(p_n - \sum_{j=1}^{n-1} q_j q_{n-j}\right).$$

Now, consider the discrete distribution X_1 with $P(X_1 = 0) = q_0 = \frac{1}{2}$ and $P(X_1 = 1) = q_1 = \frac{1}{2}$. Then, the sum distribution $X = X_1 + X_2$, where X_2 follows the same distribution as X_1, is given by $P(X = 0) = p_0 = \left(\frac{1}{2}\right)^2 = \frac{1}{4}$, $P(X = 1) = p_1 = 2 \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}$ and

$P(X = 2) = p_2 = \left(\frac{1}{2}\right)^2 = \frac{1}{4}$. If the solving method, which is described above, is used to compute the distribution of X_1 (and X_2), this gives

$$q_0 = \sqrt{p_0} = \sqrt{\frac{1}{2}} = \frac{1}{4}$$

$$q_1 = \frac{p_1}{2q_0} = \frac{\frac{1}{2}}{2 \cdot \frac{1}{2}} = \frac{1}{2}$$

$$q_2 = \frac{p_2 - q_1^2}{2q_0} = \frac{\frac{1}{4} - \left(\frac{1}{2}\right)^2}{2 \cdot \frac{1}{2}} = 0$$

$$q_3 = \frac{p_3 - 2q_1q_2}{2q_0} = \frac{0 - 2 \cdot \frac{1}{2} \cdot 0}{2 \cdot \frac{1}{2}} = 0.$$

Therefore, the initial distribution of X_1 is indeed found.

Next, consider the discrete distribution X with $P(X = 0) = p_0 = \frac{1}{4}$ and $P(X = 1) = p_1 = \frac{3}{4}$. Using the solving method to compute the distribution of X_1, the following values of q_i are found:

$$q_0 = \sqrt{p_0} = \sqrt{\frac{1}{4}} = \frac{1}{2}$$

$$q_1 = \frac{p_1}{2q_0} = \frac{\frac{3}{4}}{2 \cdot \frac{1}{2}} = \frac{3}{4}$$

$$q_2 = \frac{p_2 - q_1^2}{2q_0} = \frac{0 - \frac{9}{16}}{1} = -\frac{9}{16}.$$

In this case, negative probabilities arise. Another condition on the values of q_i is thus that the sum of every subset of q_i may never exceed 1. It can be checked that the last value of i for which $q_i > 0$ is less than or equal to a half times the last value of i for which $p_i > 0$. Therefore, if, for example, p_6 is the last value of p_i that is greater than 0, then the last value of q_i that is greater than 0, has to less than or equal to $\frac{6}{2} = 3$. This can be seen, because suppose q_4 is the last value of q_i, which is greater than 0, then also $p_7 = P(X = 7) = 2q_0q_7 + 2q_1q_6 + 2q_2q_5 + 2q_3q_4 = 2q_0 \cdot 0 + 2q_1 \cdot 0 + 2q_2 \cdot 0 + 2q_3q_4 > 0$. Therefore, $p_7 > 0$ and this leads to a contradiction, because it was supposed that q_6 was the last value of p_i, which is greater than 0.

Now, consider the discrete distribution X with $P(X = 0) = p_0$ and $P(X = 1) = p_1 = 1 - p_0$. What requirements have to be met for this distribution to be the sum of two equal distributions X_1 and X_2? For the distribution of X_1 it has to hold that

$$q_0 = \sqrt{p_0}$$

and

$$q_1 = \frac{p_1}{2q_0} = \frac{1 - p_0}{2\sqrt{p_0}}.$$

Furthermore, it has to hold that $q_0 + q_1 = 1$, so

$$\sqrt{p_0} + \frac{1 - p_0}{2\sqrt{p_0}} = 1.$$

Now define $y = \sqrt{p_0}$, then it follows

$$y + \frac{1 - y^2}{2y} = 1$$

$$2y^2 + 1 - y^2 = 2y$$
$$y^2 - 2y + 1 = 0$$
$$(y-1)^2 = 0$$

Therefore, $y = 1$ and thus $p_0 = 1$ en $q_0 = 1$ are the only distribution which satisfy these constraints. This could have been expected, because when p_1 are the last value of p_i which is greater than 0, then q_0 has to be the last value of q_i which is greater than 0. But if only $q_0 > 0$ then can p_1 never be greater than 0.

Now, consider a discrete distribution with 3 positive possibilities, $P(X = 0) = p_1, P(X = 1) = p_1$ and $P(X = 2) = p_2 = 1 - p_0 - p_1$. The solving method gives

$$q_0 = \sqrt{p_0}$$

$$q_1 = \frac{p_1}{2q_0} = \frac{p_1}{2\sqrt{p_0}}$$

$$q_2 = \frac{p_2 - q_1^2}{2q_0} = \frac{p_2 - \frac{p_1^2}{4p_0}2\sqrt{p_0}}{=}0.$$

The last equality follows from the fact that if p_2 is the last value of p_i which is greater than 0, then q_1 has to be the last value of q_i which is greater than 0. Thus,

$$\frac{p_1^2}{4p_0} = p_2 = 1 - p_0 - p_1$$

$$p_1^2 = 4p_0 - 4p_0^2 - 4p_0p_1$$
$$p_1^2 + 4p_0^2 = 4p_0 - 4p_0p_1 = 4p_0(1 - p_1).$$

Assume that p_0 is given and it holds that $p_0 = \frac{1}{2}$. Then it follows that

$$p_1^2 + 1 = 2(1 - p_1)$$

$$p_1^2 + 2p_1 + 1 - 2 = 0$$
$$(p_1 + 1)^2 = 2$$
$$p_1 = -1 \pm \sqrt{2}.$$

It follows from $p_1 > 0$ that

$$p_1 = -1 + \sqrt{2} = 0.414213.$$

Thus,

$$p_2 = 1 - p_0 - p_1 = 0.085787.$$

And

$$q_0 = \sqrt{p_0} = 0.7071$$

$$q_1 = \frac{p_1}{2q_0} = 0.2929$$
$$q_2 = 0.$$

From the above calculation, it can be concluded that it is possible to formulate a discrete distribution for X_1 for some known discrete distributions of X. But, most known discrete distributions X are not the sum of two equal distributions X_1 and X_2. When looking for the requirements of a distribution to be the sum of two equal distributions, the equalities became non linear. With an example of only three positive probabilities, the calculations could still be made, but the question is whether this is also the case with larger distribution or not. For now, it can be concluded that it is hard to find a distribution, which can be split in two equal distributions. The central question asked earlier can be answered in the following way. For some distributions it is possible to split the distribution in such a way that it is the sum of two equal distributions. These distributions are not very easy to recognize, but the calculations can be performed. Therefore, by performing a few steps of the calculation of q_i, it can be concluded whether the distribution can be split or not. Most distributions can not be split in the way required in this section.

10.3 Appendix C: Example of computing the best buffer time

In this appendix, an example of computing the best buffer time will be presented, because it shows some difficulties, which can occur during the computation. The example worked out is that of the computation of the best buffer time for the small example. Not all calculations are shown, because it will lead to many big matrices, but the consecutive distributions of the buffer times are presented.

Initialization: The initial buffer times in each port combination, can be found in the second column of table 38.

Iteration 1: In table 35, the costs of decreasing the buffer time in the corresponding port combination with one hour can be found in the second column. In the last column, the savings of increasing the buffer time in the corresponding port combination with one hour are shown. From the table, it follows that both the lowest costs and the highest savings are expected when sailing from port 2 to port 1. The second highest savings are expected in port 1 and the second lowest costs are expected when sailing from port 1 to port 2. Therefore, two possible switches have to be considered. The first one is: increasing the buffer time with one hour when sailing from port 2 to 1 and decreasing the buffer time with one hour when sailing from port 1 to port 2. Because expected savings of harbour 2-1 minus expected costs of harbour 1-2 equals $1.0669 - 1.5408 = -0.5261 \leq 0$, this is not a feasible switch. The other possibility is: increasing the buffer time with one hour in port 1 and decreasing the buffer time with one hour when sailing from port 2 to port 1. The expected savings of this possibility are: expected savings of harbour 1 minus expected costs op harbour 2-1. This equals $0.8192 - 0.7510 = 0.0682 \geq 0$. Because this is the only feasible switch, this switch will be made. The new buffer times can be found in the third column of table 38.

Table 35: Costs and savings of the first iteration

Harbour	Costs ($)	Savings ($)
1	1.5426	0.8192
2	2.9768	0.5913
1-2	1.5408	0.7640
2-1	0.7510	1.0669

Iteration 2: In table 36, the costs of decreasing the buffer time in the corresponding port combination with one hour can be found in the second column. In the last column, the savings of increasing the buffer time in the corresponding port combination with one hour are shown. From the table, it follows that the lowest costs are expected in harbour 1 and the highest savings are expected in harbour 2. Because the expected savings in harbour 2 minus the expected costs in harbour 1 equals $1.3590 - 0.8946 = 0.4544 \geq 0$, the buffer time will be increased with one hour in harbour 2 and decreased with one hour in harbour 1. The new distribution of the buffer times can be found in the fourth column of table 38.

Iteration 3: In table 37, the costs of decreasing the buffer time in the corresponding port combination with one hour can be found in the second column. In the last column, the savings of increasing the buffer time in the corresponding port combination with one hour

Table 36: Costs and savings of the second iteration

Harbour	Costs ($)	Savings ($)
1	0.8946	1.0420
2	2.3088	1.3590
1-2	1.6082	0.8115
2-1	1.5157	0.9522

are shown. From the table, it follows that the lowest costs are expected in harbour 2 and the highest savings are expected in harbour 1. Because the expected savings in harbour 1 minus the expected costs in harbour 2 equals $1.3774 - 1.0771 = 0.3003 \geq 0$, the buffer time will be increased with one hour in harbour 1 and decreased with one hour in harbour 2. The new distribution of the buffer times can be found in the fifth column of table 38. Because this distribution is the same as in the third column, iteration 2 and 3 will be repeated until infinity.

Table 37: Costs and savings of the third iteration

Harbour	Costs ($)	Savings ($)
1	1.2912	1.3774
2	1.0771	0.9142
1-2	1.3107	0.7301
2-1	1.7048	1.0401

Determining best distribution: The best distribution of the buffer time is now obtained by comparing the expected savings of iteration 2 and 3. Because the expected savings of iteration 2 are higher than those of iteration 3, $0.4544 \geq 0.3003$, the distribution of the buffer time after iteration 2 is seen as the best buffer time. This distribution can be found in the last column of table 38. When the costs of the distribution of the buffer times of iteration 2 and 3 are calculated, these seem to be respectively \$3.99 and \$4.22. Indeed, the costs of model with the distribution of the buffer times, which is obtained in iteration 2 are lower than the costs of the model with the distribution of the buffer times, which is obtained in iteration 3.

Table 38: Distributions of the buffer time

Harbour	Buffer				
	Start (hr)	It 1 (hr)	It 2 (hr)	It 3 (hr)	Best (hr)
1	1	2	1	2	1
2	1	1	2	1	2
1-2	2	2	2	2	2
2-1	3	2	2	2	2

List of Tables

List of Figures