Chapter 16
A Robust Optimisation Approach to Synchromodal Container Transportation

I. Chiscop

Abstract This chapter addresses synchromodal planning at operational level from the perspective of a logistics service provider and studies an optimisation problem with simultaneous vehicle routing and container-to-mode assignment, having uncertain data. It is assumed that the release times of the containers belong to an uncertainty interval, and no further statistical information is available. This problem belongs to the fourth quadrant of Fig. 1.4. The container routing problem is modelled as a mixed integer program with explicit time variables and lateness penalties. A robust formulation is then proposed to eliminate the uncertain parameters from the objective function and constraints. By solving the new model exactly, with the aid of an optimisation solver, robust solutions are obtained corresponding to transportation plans which remain feasible for any realisation of the release times within the pre-specified uncertainty interval. In order to introduce some flexibility in the transportation plan, the continuous time variables are modelled as affine functions of the uncertain parameters. The resulting two-stage decision model is tested for a small-sized instance in both situations, with high and low lateness penalties.

Introduction

Synchromodal transportation can be studied from multiple perspectives. There are several agents acting in the transportation network, each with their own modes and terminals/warehouses but sharing the existing infrastructure. Although synchromodality entails collaboration between all these parties, this is not always the case. Therefore, it is necessary to understand how much information is actually available and shared, and what kind of optimisation objectives are desired. The information within the network is available globally or locally. If the information

I. Chiscop (✉)
TNO, The Hague, The Netherlands
e-mail: irina.chiscop@tno.nl

F. Phillipson (ed.), *Optimisation in Synchromodal Logistics*, Lecture Notes in Operations Research, https://doi.org/10.1007/978-3-031-15655-7_16

is locally available, it means that only the agents themselves know, for example, where they are or what their status is at a certain time. If the information is global, this information is also known to the network operator, to all other agents or both. Furthermore, if all agents need to be individually optimised, the optimisation objective is local. If the optimisation objective is global, the best option for the entire network is the desired outcome.

The logistics service provider (LSP) whose activity is serving as a case study in this chapter is interested in reducing its own overall costs but has certain knowledge of the other agents in the network. This corresponds to a selfish approach to synchromodality as described in Chap. 1 and illustrated in Fig. 1.2. Given these facts, the following question arises: how can the logistics service provider optimally plan his transportation activities in order to minimise the associated costs? By investigating the characteristics of the problem further, we can develop this question into a proper research inquiry. In the following subsection we give a description of the practical setting behind the activity of the LSP and identify the optimisation problem in their planning process based on the information that was made available for us.

Use Case

Practical Setting

A logistics service provider is a company that uses its resources to offer and perform transportation services of goods from origin to destination. The company usually manages the goods being transported along the entire way and is responsible for storage and handling. In our case, the LSP has a few inland terminals and one warehouse. Moreover, it has a fleet of trucks and several kinds of chartered barges for transporting containers between the deep-sea or inland terminals and different customer warehouses in the hinterland. These barges may have different capacities. For instance, the larger ones may transport up to 156 TEU in three layers.

The LSP receives transportation requests from customers on a daily basis. These requests consist of one or more standardised containers to be picked up at a terminal and then transported either to another terminal or to the customer's warehouse. The transport between terminals is usually carried out by barge and, when this is not possible, by truck. The way in which these orders are handled within the LSP administration can be visualised in Fig. 16.1.

When a transportation order is received by the LSP the amount of information accompanying it may vary. In general, the destination and due date, namely the latest time at which the containers should arrive at their destination, are always specified. Moreover, the terminal from where these containers should be picked up, the time at which they are available for pickup and the shipping company may be indicated. However, this is not always the case. If the pickup location is known,

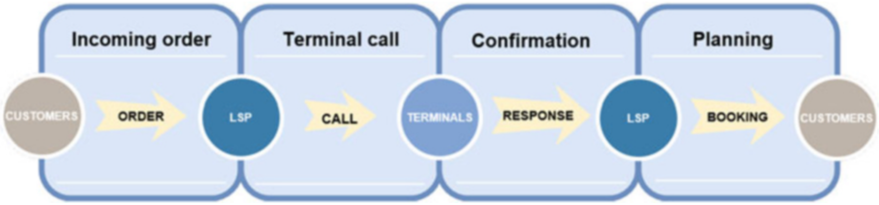

Fig. 16.1 Administration of a transportation request at the LSP

then the planners of the LSP will make a call towards that particular terminal in order to request a date and time-slot for the pickup. Depending on the working volume and the number of vessels to handle, the terminal may either confirm the proposed appointment, confirm the appointment on a different date, or not confirm an appointment at all. It is worth mentioning that the last two scenarios occur quite often in practice. Depending on the particular terminal, the time difference between the requested time and the confirmed time, otherwise known as the planning delay, can reach up to 10 days. After a response has been received from the terminal, then the LSP planners need to evaluate the current positions and loads of the available barges and decide which one will execute the pickup and when, and inform the customer about this. This process is difficult and the resulting plan is often subjected to change due to the uncertain elements in the network. The planner aims to schedule the available barges in such a way that all containers are picked up on time, then timely delivered to their destinations with a minimum amount of costs. These costs emerge from the usage of transportation modes, stationing at the terminals before the actual handling of containers and the eventual failure of meeting the due dates at the customers.

Our goal is to make use of all the practical information available in order to formulate an optimisation problem. Therefore, we need to further elaborate on what kind of elements are influencing the planning and what information is available to a planner at the moment that a decision must be made. To achieve this, we employ the framework for synchromodal problems developed in Chap. 2, where is distinguished between *resources* and *demand elements*. Intuitively, the resource elements refer to the available transportation modes, namely barges or trucks, whilst the demand elements consist of freight containers.[1] The features of these elements may be:

- *controllable*: Since we are discussing a decision problem, at least one element of the system must be in control. This can be, for instance, the allocation of demands to resources.
- *fixed*: A fixed element does not change within the scope of the problem.

[1] In this chapter the demand elements will always correspond to one container.

- *dynamic*: A dynamic element might change over time or due to a change in the state of the system (e.g., the amount of containers changes the travel time of a barge), but this change is known or computable beforehand.
- *stochastic*: A stochastic element is not necessarily known beforehand. For instance, it is not known when transportation orders will arrive, but the arrivals occur according to a Poisson process.
- *irrelevant*: It might occur that for certain problems not all elements are taken into consideration to model the system. Then these elements are irrelevant.

We will closely follow the classification in of Chap. 2 to describe all the elements occurring in the planning process of the LSP. However, not all elements encountered in the practical setting are encompassed by this framework.

Resource Elements

- *Resource type*: In this study, the LSP owns a fleet of barges of different known capacities and a uniform fleet of trucks. One may distinguish here between owned and subcontracted resources.
- *Resource features*: The resource capacities are fixed. The schedules of the barges and trucks are not fixed. Therefore the resource origin and resource destination are controllable elements. However, the resource departure time, resource travel time and resource arrival time are not controllable. This is a consequence of the delays which may occur either when receiving a confirmation from the terminal, or at the terminal itself, when the handling time takes longer than expected (this can happen due to a crane malfunction, for instance). We will classify these elements outside the framework as *uncertain*, since there is no information available concerning their distribution. Finally, we also have a resource price. Here we can distinguish between the price for employing a certain resource which is a fixed amount (per day, for instance) and the price for handling services provided at the terminals. The latter depends on the load to be handled, which is an uncertain element at the beginning of the planning period.
- *Terminal Handling time*: This refers to time required to handle different types of modes at the terminal. It includes both the waiting time and the time allocated for loading/unloading containers. This is also an *uncertain* element since there exist incoming orders which do not specify the pickup time or locations. For instance, it may be the case that a barge is waiting at a terminal to pick up some containers which have not arrived there yet.

Demand Elements

- *Demand type*: The LSP under study can transport containers of different sizes, of either 1 TEU or 2 TEU in load. Therefore, this element is fixed.
- *Demand-to-Resource allocation*: The assignment of containers to barges is essentially a decision that a planners have to make. Therefore, it is a controlled element.
- *Demand features*: The destination of a container, as well as its volume (in TEU) and due date at the customer to whom it belongs, are fixed elements. The demand

origin (pickup terminal of a container) and its release date (moment in time at which it can be loaded on a barge) are uncertain elements. This uncertainty emerges from the missing data in the transportation order, as customers simply do not specify it.

- *Demand Penalty*: This term refers to costs that are incurred when the due date at the destination for a container is not met. Since these costs are in general customer-dependent, we can classify this element as dynamic.

The resource and demand elements described are the main input for creating a schedule for the barges and trucks. However, the planning process does not only rely on the information that is available but also rely on the moment at which this information becomes available. At the beginning of the planning period, the planner knows the exact locations of all the barges and trucks in the fleet, their capacity and has a list of orders with specified destinations and due dates to be picked up sometime in the next 9 days. Moreover, at every moment in time, a planner has an estimation of the maximum and average delay of the deep-sea terminals (based on historic data in the last 30 days). This is the initial amount of knowledge. As time progresses, more information becomes available. That is, pickup locations along with release times of containers are revealed, and terminals send confirmation for appointment times. Moreover, new transportation orders may come in, which are also required to be executed within the next 9 days. This information can become available at any time so the planner must create a schedule that can handle real-time switches.

Given this practical setting, one may formulate the decision making of the LSP planners as an optimisation problem in which a routing of transportation modes and an assignment of the containers to modes must be provided under uncertain data in such a way that the total delay and costs are minimised.

Base Instance

In order to be able to develop a mathematical model and later on explore solution methods, we consider the following simplified instance obtained by reducing the size of the real-life problem and introducing some assumptions. The network comprises the following elements:

- *2 customers denoted C_1, C_2*: Their physical location is known and it is accessible only by truck.
- *2 deep-sea terminals denoted T_1, T_2*: Deep-sea vessel arrives here and unloads the containers that belong to the two customers.
- *1 container terminal operated by the LSP denoted T*: Barges leave from here and go to the deep-sea terminals to pick up containers.
- *1 hinterland terminal operated by the LSP denoted D*: It is the central terminal of the LSP, closest in distance to any customer (Fig. 16.2).

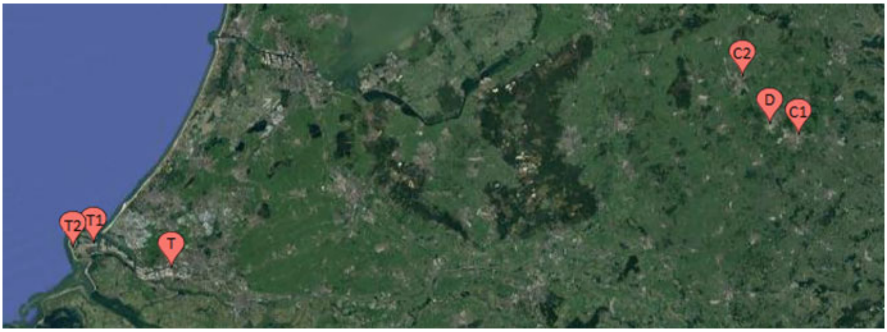

Fig. 16.2 Geographical display of the network

We notice here that there is one main difference between the container terminal and the hinterland terminal of the LSP. The container terminal is the located in the port, nearby deep-sea terminals. On the other hand, the hinterland terminal is situated further away on the continent, in the proximity of customers. This is illustrated in Fig. 16.2.

The LSP has the following resources:

- *3 barges*: All with capacity of 20 units. Two of the barges at the terminal T whilst the other one is situated at the central terminal D. There is a fixed cost per kilometre[2] travelled by a barge.
- *unlimited trucks*: All with capacity 1. There is a fixed cost per kilometre travelled by a truck.

Suppose we are given two transportation orders with the following specifications:

1. Customer C_1 asked the LSP to pick up 30 containers from T_1. The terminal has confirmed a time window for the pickup: [10, 11].[3] These containers have an uncertain release time. They will be simultaneously released sometime in the interval [10, 11]. This order needs to arrive at the customer by time unit 20.
2. Customer C_2 has 10 containers to be picked up from terminal T_2. This terminal has also confirmed a time window for the pickup: [15, 16]. All 10 containers are already available. This order needs to arrive at the customer warehouse by time unit 20.

When developing this base model we have made several assumptions. We discuss them and their relation with the real practical setting below.

- The planning period starts at midnight or otherwise interpreted, at time step 0 and covers one full day, until time step 24, respectively.

[2] We will elaborate on transportation costs of barges and trucks later in the chapter.

[3] We will take a time unit as being 1 h. Therefore, regard this interval as the time between 10:00 and 11:00.

- We assume fixed time windows at the deep-sea terminals. In practice we saw that a terminal can either answer an appointment call or not. In this scenario, we assume that we have confirmed appointment calls at the beginning of the planning period.
- If a barge arrives either too early or too late at a deep-sea terminal, it can be handled right away. So we assume that there is no waiting time involved.
- We assume that there is no handling time.
- Once it has been loaded, a barge may leave the deep-sea terminal right away.
- At any point in time, there are trucks available at every terminal, which can transport the released containers to other locations.
- There is a waterway connecting the terminals. The customers' warehouses can only be reached by truck.
- The travel times in between any two locations of the barges and trucks are known.

Given this simple instance, we are interested in minimising the overall costs and the total delay at the customers. In order to maintain a uniform objective, we can associate costs with the delay in such a way that the final objective will represent the costs overall. This simple instance will serve as a starting point in developing a mathematical model that determines an assignment of containers to transport modes, and also a specific routing of the containers. Whilst this base model is not of any practical relevance, it will serve as a basic tool to understand, and later on, to incorporate more complex features of the transport network.

After analysing the base instance, we understand that our choice for modelling approaches is somewhat restricted by the lack of probabilistic knowledge. In this case, we will study the container routing problem from a robust perspective. In other words, since we cannot employ stochastic models, we will look at robust optimisation techniques.

Deterministic Problem Formulation

In this chapter we present a mathematical model for the freight routing problem described as our base instance. We will use Sharypova's model [9] as the basis of our research and further develop it to incorporate all aspects which are of interest in the context of uncertain parameters in the transportation network. We describe all the modifications brought to the original mixed integer linear program and elaborate on a further extension of the model that can be used to incorporate multiple trips of a vehicle to a certain location.

Deterministic Model

Sharypova's model [9] serves as a starting point in our problem formulation. This model provides a transportation schedule of minimum cost which meets the strict delivery deadlines at the destinations of commodities. Since our goal is to investigate the impact of uncertainty on the transportation plan, it is reasonable to allow for more flexibility in the network, namely replace the strict deadlines of commodities by the so-called *soft due times*. This implies that a commodity may arrive later than its due date at its destination, in which case a penalty cost is incurred. To model this aspect, we introduce *lateness* decision variables L_i^m, describing how late vehicle m is at location i. Clearly, these variables are only defined when location i is a destination node for some commodity. Moreover, in our base instance we assumed that a barge arriving at a deep-sea terminal must wait a certain amount of time before it starts being loaded or unloaded. Thus, we will incorporate a terminal specific parameter w_i, representing the amount of time that a barge has to wait at terminal i. Finally, the components of the objective function must be addressed. For optimisation, the focus will be on time-related components. Generally speaking, we are interested in minimising the utilisation of trucks. However, in practice it is often the case that trucks and trails are rented by the hour and motivated by this, we will aim for minimising the trucking hours. A trucking hour is 1 h in which a truck has been utilised for transportation purposes. It is important to remark here that the amount of trucks used overall or the time travelled by a truck without being loaded are not important quantities in this setting. Moreover, we complete our multi-objective function by incorporating the total lateness recorded in the planning. This term quantifies by how much time container arrivals differ from their specified due date. Weights are associated with each of the two objectives for scaling purposes. These weights allow us to prioritise one objective over the other one, as in practice one might often find that arriving an hour late at a location might be preferable to renting a truck for another hour. A thorough discussion on these weights will follow later. We will define the following here:

Sets:

$$
\begin{aligned}
V &= \text{set of locations} \\
A &= \text{set of travelling arcs between locations} \\
M &= \text{set of vehicles} \\
K &= \text{set of commodities} \\
V_m &= \text{set of locations that can be accessed with vehicle } m
\end{aligned}
$$

Decision variables:

$x_{i,j}^{k,m}$ = number of containers of commodity $k \in K$ transported from location $i \in V$ to location $j \in V$ by vehicle $m \in M$

$$y_{i,j}^{m} = \begin{cases} 1 & \text{if vehicle } m \in M \text{ travels from location } i \in V \text{ to location } j \in V \\ 0 & \text{otherwise} \end{cases}$$

$$z^{m} = \begin{cases} 1 & \text{if vehicle } m \in M \text{ is used in the transportation plan} \\ 0 & \text{otherwise} \end{cases}$$

$$\theta_i^{m,l} = \begin{cases} 1 & \text{if a transshipment occurs between vehicle } m \in M \text{ and vehicle } l \in M \text{ at location} \\ & i \in V_m \cap V_l \\ 0 & \text{otherwise} \end{cases}$$

$$\tau_i^{k,m} = \begin{cases} 1 & \text{if any container of commodity } k \in K \text{ is loaded on vehicle } m \in M \text{ at node} \\ & i \in \{o(k), d(k)\} \\ 0 & \text{otherwise} \end{cases}$$

A_i^m = arrival time of vehicle $m \in M$ at location $i \in V$

D_i^m = departure time of vehicle $m \in M$ from location $i \in V$

L_i^m = lateness/arrival delay of vehicle $m \in M$ at location $i \in \{d(k) | k \in K\}$

$q_i^{k,m,l}$ = amount of containers of commodity $k \in K$ moved from vehicle $m \in M$ to vehicle $l \in M$ at location $i \in V_m \cap V_l$

Parameters:

s_i = service time at node $i \in V$

d_i^k = demand of commodity $k \in K$ at node $i \in V$

w_i = waiting time at terminal location $i \in V$

r_k = release time of commodity $k \in K$ at its origin

due_k = due time of commodity $k \in K$ at its final destination

$t_{i,j}^m$ = travelling time of vehicle $m \in M$ from node $i \in V$ to node $j \in V$

c^m = maximum capacity of vehicle $m \in M$

$(o(k), d(k))$ = origin-destination node pair of commodity $k \in K$

$\omega_{1,2}$ = weights of the objective functions

Given the previous decision variables and parameters, the model for container assignment and vehicle routing is [4]:

$$\min \omega_1 \sum_{k\in K} \sum_{m\in M} \sum_{(i,j)\in A} t^m_{i,j} x^{k,m}_{i,j} + \omega_2 \sum_{m\in M} \sum_{i\in V} L^m_i$$

$$s.t. \ \sum_{(j)\in V^+(i)} y^m_{i,j} - \sum_{j\in V^+(i)} y^m_{j,i} = 0 \ \forall m, \forall i \in V_m \tag{16.1}$$

$$\sum_{m\in M} \sum_{j\in V^+(i)} x^{k,m}_{i,j} - \sum_{m\in M} \sum_{j\in V^-(i)} x^{k,m}_{j,i} = d^k_i \qquad \forall i \in V, \ \forall k \tag{16.2}$$

$$\sum_{k\in K} x^{k,m}_{i,j} \leq c^m y^m_{i,j} \qquad \forall m, \forall (i,j) \in A \tag{16.3}$$

$$\sum_{k\in K} q^{k,m,l}_i > 0 \iff \theta^{m,l}_i = 1 \quad \forall m, l, \forall i \in V_m \cap V_l \tag{16.4}$$

$$\sum_{\text{æ}\in V^-(i)} x^{k,m}_{j,i} = \sum_{l\in K} q^{k,m,l}_i \qquad \forall m, k, \forall i \in V_m \setminus \{d(k)\} \tag{16.5}$$

$$\sum_{\text{æ}\in V^+(i)} x^{k,m}_{i,j} = \sum_{l\in K} q^{k,l,m}_i \qquad \forall m, k, \forall i \in V_m \setminus \{o(k)\} \tag{16.6}$$

$$\sum_{l\in K} q^{k,m,l}_i = 0 \qquad \forall m, k, \forall i \in \{o(k), d(k)\} \tag{16.7}$$

$$\sum_{j\in V^+(i)} x^{k,m}_{i,j} > 0 \iff \tau^{k,m}_i = 1 \ \forall m, k \forall i \in \{o(k)\} \tag{16.8}$$

$$\sum_{j\in V^-(i)} x^{k,m}_{j,i} > 0 \iff \tau^{k,m}_i = 1 \ \forall m, k, \forall i \in \{d(k)\} \tag{16.9}$$

$$\theta^{m,l}_i = 1 \Rightarrow D^l_i - A^m_i - s_i \geq 0 \qquad \forall m, l, \forall i \in V_m \cap V_l \tag{16.10}$$

$$y^m_{i,j} = 1 \Rightarrow D^m_i + t^m_{i,j} - A^m_j \leq 0 \quad \forall m, \forall (i,j) \in A \tag{16.11}$$

$$D^m_i \geq A^m_i + s_i \qquad \forall m, \forall i \in V_m \tag{16.12}$$

$$D^m_i \geq r_k \tau^{k,m}_i \qquad \forall m, k, \forall i \in \{o(k)\} \tag{16.13}$$

$$\tau^{k,m}_i = 1 \Rightarrow L^m_i \geq A^m_i - due_k \qquad \forall m, k, \forall i \in \{d(k)\} \tag{16.14}$$

$$\sum_{j\in V^+(i)} y^m_{i,j} \leq z^m \qquad \forall m, \forall i \in V_m \tag{16.15}$$

$$x^{k,m}_{i,j} \in \mathbb{N}_0 \qquad \forall m, k, \forall (i,j) \in A \tag{16.16}$$

$$q^{k,m,l}_i \in \mathbb{N}_0 \qquad \forall m, k, l, \forall i \in V_m \cap V_l \tag{16.17}$$

$$A^m_i, D^m_i, L^m_i \geq 0 \qquad \forall m, \forall i \in V_m \tag{16.18}$$

$$y^m_{i,j} \in \{0,1\} \qquad \forall m, \forall (i,j) \in A \tag{16.19}$$

$$\theta^{m,l}_i \in \{0,1\} \qquad \forall m, l, \forall i \in V_m \cap V_l \tag{16.20}$$

[4] In all cases where we say $\forall m, \forall l$ or $\forall k$ we mean $\forall m \in M, \forall\, l \in M$ and $\forall\, k \in K$.

$$\tau_i^{k,m} \in \{0, 1\} \qquad \forall m, k, \forall i \in \{o(k), d(k)\} \quad (16.21)$$

$$z^m \in \{0, 1\} \qquad \forall m \quad (16.22)$$

We recall that the objective is the weighted sum of trucking hours and total lateness. Constraints (16.1) ensure flow conservation at a location, while constraints (16.2) account for the demand requirement at the origin and destination of every commodity. Constraints 16.3 impose the capacity restriction of each vehicle. Constraints (16.4)–(16.7) regulate the occurrence of transshipment of containers from one vehicle to another depending on their current location. The inequalities in (16.8)–(16.9) assure that every commodity leaves its origin and arrives at its destination by means of some vehicle. The following four sets of inequalities (16.10)–(16.13) validate the time-related variables. Constraints (16.14) are of particular importance, as they establish the definition of lateness variables. Inequalities (16.15) provide the relation between used travel routes and the number of vehicles. Finally, the remaining constraints define the range of each decision variable.

The mixed integer program presented above describes a transportation problem which can be viewed as a complex extension of the capacitated vehicle routing problem with time windows (abbreviated as CVRP-TW). Since VRP is known to be NP-hard, we understand that there is no polynomial-time algorithm to solve the freight routing problem. Therefore, we expect that solving this problem even for small data instances with state-of-the-art optimisation solvers might require a considerable computational effort.

Additional Remarks

The mixed integer program described in the previous section has many binary and integer variables which makes it difficult to solve. Therefore, it is important to ensure that the solution space is as restricted as possible. In order to do so, we include the following *strong forcing* constraints:

$$x_{i,j}^{k,m} \leq \min\{D_k, C^m\} y_{i,j}^m \quad \forall (i, j) \in A, \forall k \in K, \forall m \in M,$$

where D^k is the demand of containers of commodity k, to be transported from their origin location to their destination. These constraints can be derived as flow cover inequalities and have been shown to be effective in improving the LP-relaxation of multi-commodity network design problems [6]. Therefore, they are added to the mixed integer program presented in the previous section.

A final remark concerns the modelling of trucks. Since a truck in general only has the capacity to transport one or two containers, it was preferable to not model them individually, as the size of the instance would have been too large. Instead, a number of trucks with very large capacity (set to 3000 in our instances) was enabled at every location. This number was set equal to the number of commodities to ensure that there is enough transport capacity for timely deliveries.

Robust Problem Formulation

In Chap. 4 we have presented a deterministic model for the freight routing problem which gives optimal solutions if the input data is assumed to be fully correct. However, in practice, this is almost never the case, as perturbations in data occur due to estimation, prediction or implementation errors. This sort of uncertainty may drastically affect the quality of the solution and it is not considered in deterministic optimisation. Nevertheless, it can be handled by stochastic optimisation (SO) and robust optimisation (RO). Stochastic programming is a commonly used method which optimises the problem by making use of the parameters' expected value. This approach generates a number of scenarios that represent the possible realisations of the stochastic parameters, assigns a probability to each of these scenarios and finally, creates a model optimising over all scenarios. Stochastic programming cannot be used when detailed statistical information is missing or when the number of scenarios becomes too large, making the problem intractable. In Chap. 4, it was shown that already for a small instance the freight routing problem with fixed vehicle routes and stochastic travel times, the scenario tree becomes prohibitively large. Although we will not investigate stochastic programming further, we refer the interested reader to [5]. The robust optimisation framework, on the other hand, is based on obtaining solutions which remain feasible for any realisation of the parameters within a predefined uncertainty set. For this reason, we will explore in this chapter how can the adjustable robust optimisation be used in order to deal with the uncertainty in the release times of the containers. We present the robust optimisation paradigm and explain how to formulate and solve the robust counterpart. The robust mathematical formulation of the freight routing problem is given at the end of this section.

Robust Optimisation Paradigm

Robust optimisation is an increasingly popular methodology to model mathematical optimisation programs with uncertain data. Instead of assuming a known probability distribution, the uncertain data is presumed to reside in a user-specified set of realisations, called the uncertainty set. We consider a general formulation of an uncertain linear optimisation problem:

$$\min_{x}\{c^T x : Ax \leq b\}, \tag{P$_0$}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Suppose that the matrix A is uncertain and it belongs to a bounded uncertainty set $\mathcal{U}_A \subset \mathbb{R}^{m\times n}$. In a similar fashion we assume that right hand side vector b belongs to uncertainty set $\mathcal{U}_b \subset \mathbb{R}^m$, whilst the objective coefficients c reside in the uncertainty set $\mathcal{U}_c \subset \mathbb{R}^n$. The sets $\mathcal{U}_A$, $\mathcal{U}_b$ and $\mathcal{U}_c$ specify all possible realisations of the uncertain data and are collectively referred to as the uncertainty set $\mathcal{U}$. The robust optimisation paradigm as described by Ben-Tal et al. [2] relies on the following assumptions:

A.1 All decision variables $x \in \mathbb{R}^n$ represent here-and-now decisions: they should be assigned specific numerical values as a result of solving the problem before the actual data 'reveals itself'.

A.2 The decision maker is fully responsible for consequences of the decisions to be made when, and only when, the actual data is within the pre-specified uncertainty set $\mathcal{U}$.

A.3 All the constraints of the uncertain problem in case are 'hard': we cannot tolerate violations of constraints when the data is in the uncertainty set $\mathcal{U}$.

These assumptions indicate what are the relevant feasible solutions of the linear uncertain problem P_0. The first assumption **A.1** asserts that the solution vector should have fixed values or otherwise said, it should not contain any components to which there has not been assigned a numerical value. By assumptions **A.2** and **A.3** this solution vector should satisfy all the constraints, regardless of the realisation of the data in the uncertainty set $\mathcal{U}$. Such a solution is called *robust feasible* [2]. Thus we understand that robust optimisation is concerned with finding robust feasible solutions for problems with a predefined uncertainty set.

The Robust Counterpart

We observe that the linear uncertain problem P_0 exhibits uncertainty in all parameters. In fact, one can show that this problem can be re-formulated in such a way that only the matrix A will contain uncertain entries. Firstly, the uncertainty in the objective function can be removed by introducing an additional continuous decision variable $t \in \mathbb{R}$. Problem P_0 is then equivalent to:

$$\min_{x,t}\{t : c^T x - t \leq 0 \quad \forall c \in \mathcal{U}_c, \quad Ax \leq b \quad \forall A \in \mathcal{U}_A, \forall b \in \mathcal{U}_b\}.$$

Secondly, the uncertain components of vector b can be transferred to the matrix A in the following way: vector b is added as a column of A and value $x_{n+1} = -1$ is added as an extra component to the vector x. Then the problem P_0 can be written as:

$$\min_{x,t}\{t : c^T x - t \leq 0 \quad \forall c \in \mathcal{U}_c, \quad Ax \leq 0 \quad \forall A \in \mathcal{U}_A \cup \mathcal{U}_b\}.$$

Given these two reformulations, we conclude that it is always safe to assume that uncertain quantities occur only in the matrix of coefficients. This being said, we can finally give a most general form of the uncertain linear problem as:

$$\min_{x}\{c^T x : Ax \leq b \quad \forall A \in \mathcal{U}\}. \tag{P}$$

The robust reformulation of problem P is referred to as the *robust counterpart* (RC) problem [2] and we will present it as given in [10]. We assume that the coefficient matrix $A(\zeta)$ is an affine[5] function of the uncertain parameter ζ:

$$\min_x \{c^T x : A(\zeta)x \leq b \quad \forall \zeta \in \mathcal{Z}\}, \tag{RC}$$

where $\mathcal{Z} \subset \mathbb{R}^p$ denotes the user defined uncertainty set. Recall that a solution x is robust feasible if the constraints $A(\zeta)x \leq b$ are satisfied for every value of $\zeta \in \mathcal{Z}$. As discussed in [2], the robust counterpart of an uncertain linear optimisation problem with a certain objective is a 'constraint-wise' construction. In other words, the original ith row constraint $(Ax)_i \leq b_i \Leftrightarrow a_i^T x \leq b_i$, (with a_i being the ith row in A) from the nominal problem is replaced by $a_i^T x \leq b_i \quad \forall [a_i; b_i] \in \mathcal{U}_i$, where $\mathcal{U}_i$ is the projection of $\mathcal{U}$ on the space of data of ith constraint: $\mathcal{U}_i = \{[a_i; b_i] : [A, b] \in \mathcal{U}\}$. Therefore, we can address the uncertainty by a single constraint. For instance, we extract one constraint from the robust counterpart problem RC modelled as an affine expression in terms of ζ:

$$(a_i + P\zeta)^T x \leq b_i \quad \forall \zeta \in \mathcal{Z}, \tag{16.23}$$

where $a_i \in \mathbb{R}^n$ is interpreted as the nominal value of the data, $P \in \mathbb{R}^{n \times p}$ and $b_i \in \mathbb{R}$. The idea behind this process is to reformulate the robust counterpart constraint-wise in such a way that it becomes computationally tractable. The expression in (16.23) has infinitely many constraints due to the *for all* (∀) quantifier and it is thus intractable in general. In [10] the authors provide a compact overview of the steps to be followed in order to remove this quantifier. We will closely follow their approach. Consider a polyhedral uncertainty set defined as:

$$\mathcal{Z} = \{\zeta : D\zeta + q \geq 0\}, \tag{16.24}$$

where $D \in \mathbb{R}^{m \times p}$, $\zeta \in \mathbb{R}^p$ and $q \in \mathbb{R}^n$.

In a worst-case reformulation, when the realisation of the uncertain data yields the largest objective value, one can re-write the nominal problem P as:

$$a_i^T x + \max_\zeta \{(P^T x)^T \zeta : \quad D\zeta + q \geq 0\} \leq b_i. \tag{16.25}$$

By strong duality, the inner maximisation problem in the expression above can be replaced by its dual. Therefore, expression (16.25) is equivalent to:

$$a_i^T x + \min_w \{q^T w : \quad D^T w = -P^T x, \quad w \geq 0\} \leq b_i. \tag{16.26}$$

[5] A function $f : A \to B$ is affine if and only if the mapping $x \mapsto f(x) - f(0)$ is linear.

We see that in order to satisfy inequality (16.26), it suffices to find at least one w. Hence, the final formulation of the RC is given by:

$$\exists w : \quad a_i^T x + q^T x \leq b_i, \quad D^T w = -P^T x, \quad w \geq 0, \tag{16.27}$$

which is an LP feasibility problem.

From everything that we have done so far, we conclude that solving the robust counterpart of a general linear optimisation problem with continuous variables and a polyhedral uncertainty set reduces to finding a feasible solution to the linear problem described in Eq. (16.27). Therefore, the robust counterpart of an uncertain linear program (LP) with a polyhedral uncertainty set is in fact a computationally tractable LP. Moreover, this property also holds for the so-called box uncertainty set of the form: $\mathcal{Z} = \{\zeta : \|\zeta\|_\infty \leq 1\}$, since the robust counterpart in this case is simply given by $a_i^T x + \|P^T x\|_1 \leq b_i$ [10]. For a thorough mathematical discussion on tractability properties of the robust counterpart for various uncertainty sets, the reader is referred to the book of Ben-Tal et al. [2].

Adjustable Robust Optimisation

The robust optimisation formulation given earlier is static in the sense that the numerical values of all decision variables must be determined before the uncertain quantities reveal their true value. For this reason, the solutions obtained by solving the robust counterpart are indeed robust feasible but sometimes very conservative: they are only optimal for the worst-case realisations of the uncertain data. With this static approach it may often be the case that the objective function of the solution becomes unnecessarily high given the actual data realisations attained in practice. This concept is also known as the *price of robustness*, described by Bertsimas and Sim [4] as the trade-off between the optimal solution and robustness. In order to achieve a reasonable price of robustness, the adjustable robust optimisation framework has been proposed [3]. In this framework, assumption **A.1** from the robust optimisation paradigm is relaxed, meaning that we allow for some *wait-and-see* decision variables. In other words, some decision variables can be adjusted at a later point in time according to the realisation of the data. Most commonly, these adjustable decisions are modelled as functions of the uncertain data. In view of this, the adjustable robust counterpart (ARC) can be formulated as:

$$\min_{x, y(\cdot)} \{c^T x : A(\zeta)x + By(\zeta) \leq b\} \quad \forall \zeta \in \mathcal{Z}, \tag{ARC}$$

where $x \in \mathbb{R}^n$ represents the first stage here-and-now decision vector that is made before $\zeta \in \mathbb{R}^p$ is realised, $y \in \mathbb{R}^k$ denotes the second-stage wait-and-see decision vector that can be computed according to the realisation of ζ, and $B \in \mathbb{R}^{n \times k}$ is a given coefficient matrix. For the scope of this chapter it is sufficient to assume that the matrix B does not contain any uncertain elements. In general, it is difficult to optimise over functions, so a commonly used approach is to express the adjustable decision variables as affine functions of the uncertain data, namely:

$$y(\zeta) = y_0 + Q\zeta. \tag{16.28}$$

In the expression above, $y_0 \in \mathbb{R}^k$ and $Q \in \mathbb{R}^{k\times p}$ are here-and-now decisions to be optimised by the model in the first stage. Substituting the expression for y given in Eq. (16.28) into the ARC we obtain the affinely adjustable robust counterpart (AARC):

$$\min_{x,y_0,Q} \{c^T x : A(\zeta)x + By_0 + BQ\zeta \le b \quad \forall\zeta \in \mathcal{Z}\}. \tag{AARC}$$

Since the AARC is linear in both the decision variables and the uncertain parameter, it can be solved by following the same reformulation steps as in the previous section. Therefore, the AARC has the same tractability as the original robust counterpart, regardless of the uncertainty set chosen. Two important remarks are required here. First of all, the AARC might contain many more decision variables than the RC due to the size of matrix Q. Secondly, although the AARC will likely require more computational effort, the solution thus obtained will be at least as good as the one given by solving the RC.

Up to this point, we have presented both the static and the affinely adjustable robust counterpart problems and showed that in the case of linear programs with polyhedral or box uncertainty, both formulations are tractable. The fact that we can provide adjustable robust feasible solutions, makes the robust optimisation approach extremely appealing for further applying it to our freight routing problem. Nevertheless, the model developed includes many binary and integer variables for which the mathematical treatment is not directly applicable. Therefore, we will further discuss how robust optimisation techniques can be used in the context of mixed integer programs.

Robust Optimisation for Mixed Integer Programs

A mixed integer program is a mathematical program which contains both real valued decision variables and variables restricted to take integer values. It is well known that determining whether a feasible solution of a given mixed integer program with rational coefficients exists is in the class of NP-complete problems [7]. As such, we expect that a robust counterpart of a mixed integer linear program is also intractable. Consider the general form of a mixed integer program:

$$\min_{x,y}\{c^T x + d^T y : \quad Ax + Gy \le p, \quad x \in \mathbb{Z}^n, y \in \mathbb{R}^k\}, \tag{MIP}$$

where $c \in \mathbb{Q}^n$ and $d \in \mathbb{Q}^k$ are given cost vectors, $A \in \mathbb{Q}^{l\times n}$ and $G \in \mathbb{Q}^{l\times k}$ are coefficients matrices and $p \in \mathbb{Q}^l$. We assume that the matrix A is the only element affected by uncertainty. This assumption is motivated by our deterministic model in Chap. 4, in which the uncertain release time is multiplied with a binary variable in constraints (16.14). Thus we consider a model of the form:

$$\min_{x \in \mathbb{Z}^n, y \in \mathbb{R}^k} \{c^T x + d^T y : \quad A(\zeta)x + Gy \leq p, \quad \forall \zeta \in \mathcal{Z}\}. \tag{RC-MIP}$$

Since uncertainty was showed to appear constraint-wise in a general linear program, we can once again model uncertainty affected constraints by an affine transformation of the uncertainty term $\zeta \in \mathcal{Z}$, namely every element of A can be written as a summation between a linear combination of the components of vector ζ and a constant:

$$A(\zeta)^T = \left[a_1(\zeta)\ a_2(\zeta)\ \ldots\ a_n(\zeta)\right] = \left[a_1\ a_2\ \ldots\ a_n\right] + \left[P_1\zeta\ \ P_2\zeta\ \ldots\ P_n\zeta\right]. \tag{16.29}$$

The robust counterpart then contains constraints of the form:

$$\min_{x \in \mathbb{Z}^n, y \in \mathbb{R}^k} \{c^T x + d^T y : \quad (a_i + P\zeta)^T x + g_i^T y \leq p_i, \quad \forall i \in 1, \ldots, l \quad \forall \zeta \in \mathcal{Z}\}, \tag{16.30}$$

where $a_i \in \mathbb{Q}^n$ is the nominal value, $P \in \mathbb{R}^{n \times p}$ is the matrix with vectors $P_1, P_2, \ldots, P_n \in \mathbb{R}^p$ as columns, g_i^T is a vector corresponding to the ith row of matrix G and p_i is the ith entry of vector p. Just as in the case of a general linear program, we now wish to bring the RC-MIP problem into a reasonable form, removing the 'for all' ($\forall$) operator. The uncertainty set to be considered is the simple box uncertainty:

$$\mathcal{Z} = \{\|\zeta\|_\infty \leq 1\}. \tag{16.31}$$

This kind of uncertainty set is the most intuitive for the freight routing problem, since the release time of a container is assumed to belong to a certain bounded interval of time. Using the worst-case values of the uncertain parameter ζ, the robust counterpart RC-MIP is re-formulated as:

$$\min_{x \in \mathbb{Z}^n, y \in \mathbb{R}^k} \{c^T x + d^T y : \quad a_i^T x + \|P^T x\|_1 + g_i^T y \leq p_i \quad \forall i \in 1, \ldots, l\}. \tag{16.32}$$

We observe that expression (16.31) is a convex optimisation problem that can be re-written as a linear mixed integer problem by introducing auxiliary decision variables. The next step from here is to adjust the continuous variables which in our freight routing model correspond to time-related decisions. We assume that they can be written as affine functions of the uncertainty as in expression (16.28). By doing so, we obtain the following adjustable robust counterpart:

$$\min_{x\in\mathbb{Z}^n, y_0\in\mathbb{R}^k, Q\in\mathbb{R}^{k\times p}} \{c^T x + d^T(y_0 + Q\zeta) : \quad A(\zeta)x + G(y_0 + Q\zeta) \leq p, \quad \forall\zeta \in \mathcal{Z}\}. \tag{16.33}$$

In the case of box uncertainty, this can be formulated as a convex problem very similar to (16.31):

$$\min_{x\in\mathbb{Z}^n, y_0\in\mathbb{R}^k, Q\in\mathbb{R}^{k\times p}} \{c^T x + d^T(y_0 + Q\zeta) : \quad a_i^T x + \|P^T x + Q^T g_i\|_1 + g_i^T y_0 \leq p_i\}. \tag{16.34}$$

The final form of this mixed integer problem without uncertainty removed from the objective and constraints is:

$$\begin{aligned}\min_{x\in\mathbb{Z}^n, y_0\in\mathbb{R}^k, Q\in\mathbb{R}^{k\times p}, t\in\mathbb{R}} \{t :& c^T x + d^T y_0 + \|Q^T d\|_1 - t \leq 0, \\ & a_i^T x + \|P^T x + Q^T g_i\|_1 + g_i^T y_0 \leq p_i \quad \forall i \in 1, \ldots, l\}.\end{aligned} \tag{ARC-MIP}$$

Finding a solution to the ARC-MIP reduces to solving a mixed integer program bigger in size than the original MIP. Nevertheless, it provides a suitable modelling framework for the freight routing problem and a way to find static and adjustable robust feasible solutions. Since the robust optimisation approach has been discussed for both a general linear program and the mixed integer case, we are now ready to present a robust model for the freight routing problem.

Robust Model

In the robust model the release times of the commodities are uncertain. We recall that every container has a predefined earliest and latest pickup time from its terminal of origin, and the moment at which it is actually released from the terminal and available for loading on the vehicle is contained in this time window. In mathematical terms we have:

$$r_k \in [e_k, l_k] \quad \forall k \in K,$$

where e_k and l_k mark the earliest and the latest pickup time, respectively. In practice, these two quantities are made available in advance by the terminal where the pickup should occur. The release is known to take place sometime between these two moments. This can be modelled as follows:

$$r_k = \frac{1}{2}e_k(1-\zeta_k) + \frac{1}{2}l_k(1+\zeta_k) \quad \forall k \in K,$$

where $\zeta_k \in [-1, 1]$ is the actual uncertain parameter based on which the release r_k can be computed. Therefore, just as in Sect. "Robust Optimisation for Mixed Integer Programs", the uncertainty set is the simple boxed uncertainty given by:

$$\mathcal{Z} = \{\zeta \in \mathbb{R}^k : \zeta_k \in [-1, 1]\}.$$

Based on this uncertainty set, we introduce an adjustable robust model which contains two stages of decisions: the first stage variables that must be determined before the value of the uncertain parameter becomes known, and second-stage decision variables which can change their value according to the realisation of the parameters. In our robust model, the first stage variables $x_{i,j}^{k,m}$, $y_{i,j}^{m}$, z^m, $\theta_i^{m,l}$, $\tau_i^{k,m}$ and $q_i^{k,m,l}$ concern the routing, the sequence of terminal visits, the assignment and transshipment of containers. The second-stage decisions are the continuous variable D_i^m, A_i^m and L_i^m which account for the explicit departure and arrival times and are modelled as adjustable variables. The idea of adjusting time variables to the uncertain parameters originates from Agra et al. [1], who give a robust formulation for a maritime inventory routing problem with uncertain vessel sailing times. Therefore, we define $D_i^m(\zeta)$, $A_i^m(\zeta)$ and $L_i^m(\zeta)$ as the arrival time, departure time and lateness, respectively, when scenario ζ (a vector containing release times of all commodities) has been revealed.

The first stage solution must ensure that, for each possible realisation of the release times in the uncertainty set, the containers are transported from their origin to their destination without missing any of their planned transshipment on the way. In other words, these decisions should result in a robust plan that can be carried out regardless of delayed releases of containers. In the original deterministic model, all time-related constraints (16.10)–(16.14) become:

$$\theta_i^{m,l} = 1 \Rightarrow D_i^l(\zeta) - A_i^m(\zeta) - s_i \geq 0 \qquad \forall m, l \in M, \forall i \in V_m \cap V_l, \forall \zeta \in \mathcal{Z} \tag{16.35}$$

$$y_{i,j}^m = 1 \Rightarrow D_i^m(\zeta) + t_{i,j}^m - A_j^m(\zeta) \leq 0 \qquad \forall m \in M, \forall (i,j) \in A, \forall \zeta \in \mathcal{Z} \tag{16.36}$$

$$D_i^m(\zeta) \geq A_i^m(\zeta) + s_i \qquad \forall m \in M, \forall i \in V_m, \forall \zeta \in \mathcal{Z} \tag{16.37}$$

$$D_i^m(\zeta) \geq r_k \tau_i^{k,m} \qquad \forall k \in K, \forall i \in \{o(k)\}, \forall m \in M, \forall \zeta \in \mathcal{Z} \tag{16.38}$$

$$\tau_i^{k,m} = 1 \Rightarrow L_i^m(\zeta) \geq A_i^m(\zeta) - due_k \qquad \forall k \in K, \forall i \in \{d(k)\}, \forall m \in M, \forall \zeta \in \mathcal{Z}. \tag{16.39}$$

As we have already discussed in the previous section, a common approach to handle adjustable variables is to use affine decision rules. In this case, we can write the arrival and departure times as affine functions of the uncertain release times:

$$D_i^m(\zeta) = D_{i,0}^m + \sum_{k\in K} D_{i,k}^m \zeta_k \quad \forall i \in V, \forall m \in M \tag{16.40}$$

$$A_i^m(\zeta) = A_{i,0}^m + \sum_{k\in K} A_{i,k}^m \zeta_k \quad \forall i \in V, \forall m \in M \tag{16.41}$$

$$L_i^m(\zeta) = L_{i,0}^m + \sum_{k\in K} L_{i,k}^m \zeta_k \quad \forall i \in \{d(k) : k \in K\}, \forall m \in M. \tag{16.42}$$

The newly introduced variables $D_{i,0}^m \geq 0$, $D_{i,k}^m \in \mathbb{R}$ and so on must be determined in the first stage, together with the routing, assignment and transshipment decisions. We are interested in robust feasible solutions that satisfy constraints (16.35)–(16.39) for any realisation of the release time vector $\zeta \in \mathcal{Z}$. Such a solution must also satisfy the following re-formulated constraints ($m, l \in M, k \in K, \zeta inZ$):

$$\theta_i^{m,l} = 1 \Rightarrow D_{i,0}^l + \sum_{k\in K} D_{i,k}^l \zeta_k \geq s_i + A_{i,0}^m + \sum_{k\in K} A_{i,k}^m \zeta_k \qquad \forall m, l, \zeta, \forall i \in V_m \cap V_l \tag{16.43}$$

$$y_{i,j}^m = 1 \Rightarrow D_{i,0}^m + \sum_{k\in K} D_{i,k}^m \zeta_k + t_{i,j}^m \leq A_{j,0}^m + \sum_{k\in K} A_{j,k}^m \zeta_k \qquad \forall m, \zeta, \forall (i,j) \in A \tag{16.44}$$

$$D_{i,0}^m + \sum_{k\in K} D_{i,k}^m \zeta_k \geq A_{i,0}^m + \sum_{k\in K} A_{i,k}^m \zeta_k + s_i \qquad \forall m, \zeta, \forall i \in V_m \tag{16.45}$$

$$D_{i,0}^m + \sum_{k\in K} D_{i,k}^m \zeta_k \geq \big(\frac{1}{2} e_k (1 - \zeta_k) + \frac{1}{2} l_k (1 + \zeta_k)\big) \tau_i^{k,m} \qquad \forall k, m, \zeta, \forall i \in \{o(k)\} \tag{16.46}$$

$$\tau_i^{k,m} = 1 \Rightarrow L_{i,0}^m + \sum_{k\in K} L_{i,k}^m \zeta_k \geq A_{i,0}^m + \sum_{k\in K} A_{i,k}^m \zeta_k - due_k \quad \forall k, m, \zeta, \forall i \in \{d(k)\}. \tag{16.47}$$

As shown earlier, the uncertainty ζ_k can be removed from the constraints by assuming a worst-case realisation of the data. For example, constraints (16.43) can be written as follows:

$$\theta_i^{m,l} = 1 \Rightarrow D_{i,0}^l - s_i - A_{i,0}^m \geq \sum_{k\in K} (A_{i,k}^m - D_{i,k}^l) \zeta_k \quad \forall m, l \in M, \forall i \in V_m \cap V_l, \forall \zeta \in Z.$$

Since this inequality should hold for any realisation of ζ_k, we impose the following constraint:

$$\theta_i^{m,l} = 1 \Rightarrow D_{i,0}^l - s_i - A_{i,0}^m \geq \sum_{k \in K} |A_{i,k}^m - D_{i,k}^l| \quad \forall m, l \in M, \forall i \in V_m \cap V_l.$$

Moreover, we note that in the constraints above there is no uncertain parameter anymore and all the decision variables are to be determined in the first stage. Moreover, the absolute value can be removed from the expression by introducing an additional decision variable $\alpha_{m,l}^{i,k}$:

$$\theta_i^{m,l} = 1 \Rightarrow D_{i,0}^l - s_i - A_{i,0}^m \geq \sum_{k \in K} \alpha_{m,l}^{i,k} \qquad \forall m, l \in M, \forall i \in V_m \cap V_l$$

$$-\alpha_{m,l}^{i,k} \leq A_{i,k}^m - D_{i,k}^l \leq \alpha_{m,l}^{i,k} \qquad \forall k \in K, \forall m, l \in M, \forall i \in V_m \cup V_l.$$

One can reformulate constraints (16.44)–(16.47) in a similar fashion and obtain the following inequalities:

$$y_{i,j}^m = 1 \Rightarrow D_{i,0}^m + t_{i,j}^m - A_{j,0}^m + \sum_{k \in K} \beta_{k,m}^{i,j} \leq 0 \qquad \forall m, \forall (i,j) \in A \tag{16.48}$$

$$-\beta_{k,m}^{i,j} \leq D_{i,k}^m - A_{j,k}^m \leq \beta_{k,m}^{i,j} \qquad \forall (i,j) \in A, \forall k, m$$

$$D_{i,0}^m - A_{i,0}^m - s_i \geq \sum_{k \in K} \gamma_{k,m}^i \qquad \forall m, \forall i \in V_m \tag{16.49}$$

$$-\gamma_{k,m}^i \leq A_{i,k}^m - D_{i,k}^m \leq \gamma_{k,m}^i \qquad \forall i \in V_m, \forall k, m$$

$$D_{i,0}^m \geq \delta_{k_0,m} + \sum_{k \neq k_0} \epsilon_{i,k}^m \qquad \forall k_0 \in K, \forall i \in \{o(k)\}, \forall m \tag{16.50}$$

$$-\delta_{k_0,m}^i \leq \frac{1}{2}(l_{k_0} - e_{k_0})\tau_i^{k,m} - D_{i,k_0}^m \leq \delta_{k_0,m}^i \qquad \forall k_0 \in K, \forall i \in \{o(k)\}, \forall m$$

$$-\epsilon_{i,k}^m \leq D_{i,k}^m \leq \epsilon_{i,k}^m \qquad \forall k, m, \forall i \in \{o(k)\}$$

$$\tau_i^{k,m} = 1 \Rightarrow L_{i,0}^m + due_k - A_{i,0}^m \geq \sum_{k \in K} \eta_{k,m}^i \qquad \forall k, m, \forall i \in \{d(k)\} \tag{16.51}$$

$$-\eta_{k,m}^i \leq A_{i,k}^m - L_{i,k}^m \leq \eta_{k,m}^i \qquad \forall k, m, \forall i \in \{d(k)\}.$$

We observe that in the constraints above there is no uncertain parameter ζ_k present anymore and all the decision variables are to be determined in the first stage.

The adjustable robust counterpart of the deterministic model presented in Chap. 4 is thus composed from constraints (16.1)–(16.9), (16.15)–(16.22) and (16.48)–(16.51). The objective of the mixed integer program is modelled according to the method described in the beginning of Sect. "The Robust Counterpart", meaning that the following expression is added to finalise the model:

$$\max \quad t \qquad \text{where} \qquad \omega_1 \sum_{k\in K} \sum_{m\in M} \sum_{(i,j)\in A} t^m_{i,j} x^{k,m}_{i,j} + \omega_2 \sum_{m\in M} \sum_{i\in V} \big(L^m_{i,0} + \sum_{k\in K} \mu^m_{i,k}\big) \leq t$$
$$-\mu^m_{i,k} \leq L^m_{i,k} \leq \mu^m_{i,k}, \quad \forall k \in K, \forall i \in \{d(k)\}, \forall m \in M. \tag{16.52}$$

Solving the robust model will determine all the routing, assignment and transshipment variables. The value obtained for the objective value corresponds to the worst-case realisation of the data. Nevertheless, using the adjustable time variables in the second stage, when the data is revealed, we can improve the value of the objective without re-solving the model. That is due to the fact that in the second stage the lateness term in the objective can still be adjusted and reduced when the realisation of the data is favourable.

Computational Results

In this section, we report on the solutions found for the deterministic and robust formulations of the freight routing problem and compare them to past approaches. We explain how the test instances were generated and show the results for the deterministic model and the robust approach.

Instance Generation

In order to test the models that were given before, we generated multiple problem instances. These were inspired from the work of Kishan Kalicharan [8], who has designed a transport network of eight terminal locations based on Google Maps data. Since some of these locations represented clustered terminals, the original instances were modified to include only nodes which correspond to actual physical locations in real life. For comparison purposes, the number of locations was kept the same. The barge travel times on waterways were assumed to be fixed and their values were approximated using online tools which compute sea distances based on the speed of the vessel. In our transport infrastructure, we assume that some of the locations are terminals, where containers can be transshipped, and some of them are customers, serving as end-locations for the containers. There is also direct connection between every pair of locations in our model. Furthermore, we assume that the service time is the same at every location. There is a set of commodities (bookings with one or more containers) that need to be transported from the terminals to the customer locations. As in [8], the demand value of each commodity is randomly chosen in the interval [0, 125].

Barges and trucks are available for container transport. The capacity of barges is assumed to be of 100 containers. These barges always start at a particular terminal which in real-life interpretation in a hub-location. We assume that there is an infinite amount of trucks of large capacity available at every location. To ensure that all containers can be transported, the total capacity of all vehicles is always larger than the total demand of all commodities [9]. Finally, the due dates and release times of the containers are chosen in such a way that the difference between them is strictly larger than the time required by barge to travel on the direct connection arc from the origin of the containers to their destination.

To assess the computational difficulty of our models we create instances with 8 nodes, 6 and 12 barges, and 5, 10, 20 and 30 commodities. In total, we generate 10 instances which are tested for three different objective functions by varying the values of the weights ω_1 and ω_2. We denote a problem instance by km, where k the number of commodities and m the number of available barges. Both the deterministic and the robust model were implemented in AIMMS Developer version 4.53, a mathematical optimisation modelling tool, and solved with CPLEX optimisation solver (Version 12.8, 32-bit). Numerical experiments were carried out on a DELL Latitude E7240 laptop with an Intel(R) Core(TM) i5-4310U CPU 2.00 GHz 2.60 GHz processor and 8 GB RAM memory. This laptop is operational on a 64-bit operating system.

Results of Deterministic Model

In the freight routing problem we are interested in providing an assignment of containers to vehicles whilst minimising the total number of trucking hours and overall lateness. In order to get an idea of how the allowed lateness affects the solution time of the freight routing problem, the deterministic model in Chap. 4 was tested for three different objective functions. These were obtained by varying the weights w_1 and w_2. Since we have no knowledge of the real costs of trucking activities in practice, we shall gradually increase the weight ω_2 of lateness and keep the first weight $\omega_1 = 1$. The following three objective functions are considered:

- Objective 1: $\sum_{k\in K}\sum_{m\in M}\sum_{(i,j)\in A} t_{i,j}^{m} x_{i,j}^{k,m} + \sum_{m\in M}\sum_{i\in V} L_i^m$
- Objective 2: $\sum_{k\in K}\sum_{m\in M}\sum_{(i,j)\in A} t_{i,j}^{m} x_{i,j}^{k,m} + 0.1 \cdot \sum_{m\in M}\sum_{i\in V} L_i^m$
- Objective 3: $\sum_{k\in K}\sum_{m\in M}\sum_{(i,j)\in A} t_{i,j}^{m} x_{i,j}^{k,m} + 1000 \cdot \sum_{m\in M}\sum_{i\in V} L_i^m$

These values give us a reasonable way to assess which objective yields a solution fast enough. Since these weights are chosen in a way that highly penalises lateness, the results of our model should be comparable to those obtained when completely

Table 16.1 Objective value, solution time (CPU seconds) and gaps between the lower and upper bounds for the freight routing model with lateness allowed. * an upper bound of 3600s was set on the running time of the solver

Data instance	Obj. 1	Gap (%)	Runtime (s)	Obj. 2	Gap (%)	Runtime (s)	Obj. 3	Gap (%)	Runtime (s)
k5m6	100	0	1.94	100	0	1.94	100	0	1.8
k10m6	149.5	0	22.56	149.5	0	22.92	149.5	0	72.53
k20m6	235.5	0	263	235.5	0	245.36	235.5	0	565.52
k30m6	na	na	3600*	na	na	3600*	na	na	3600*
k5m12	100	0	2.06	100	0	3.52	100	0	2.22
k10m12	149.5	0	19.23	149.5	0	24.02	149.5	0	18.9
k20m12	235.5	0	303.47	235.5	0	380.89	235.5	0	178.36
k30m12	na	na	3600*	na	na	3600*	na	na	3600*

Table 16.2 Objective value, solution time (CPU seconds) and gaps between the lower and upper bounds for the freight routing model with no lateness allowed. * an upper bound of 3600s was set on the running time of the solver

Data instance	Objective	Gap (%)	Runtime (s)
k5m6	100	0	4.84
k10m6	149.5	0	114.08
k20m6	na	na	3600*
k30m6	na	na	3600*
k5m12	100	0	30.38
k10m12	149.5	0	21.66
k20m12	235.5	0	319.8
k30m12	na	na	3600*

removing lateness variables. Table 16.1 shows the solution and computation time for each instance and each of the three objective functions that were chosen. The results obtained for the original model with no lateness allowed are given in Table 16.2.

What immediately stands out from the results above is the computational difficulty of the deterministic models for freight routing as for instances with thirty commodities k30m6 and k30m12 the solver could not find a feasible solution within 1 h for any of the models considered. However, we see that the model incorporating lateness performs better than the original version in terms of the computational time required and the solution found. For example, instance k20m6 can be solved to optimality for all the three objective functions considered in the case of allowed lateness but not for the original model. This might be due to the fact that the model including lateness is always feasible, and therefore, it is easier for the solver to find an initial feasible solution than in the situation of hard due dates for the commodities.

In general, we note that the computational time significantly increases for all cases considered when the number of commodities increases. On the other hand, the number of vehicles does not seem to drastically influence the computational time of the instances that we have tested since there are no compelling differences between instances with six or twelve barges. In particular, for the instance with twenty commodities the solver found an optimal solution three times faster when the number of available barges was doubled. This result confirms our expectation, as a larger fleet of barges offers more routing possibilities and requires less transsshipments of containers to trucks.

Regarding the two model formulations, with and without allowing for lateness, we see that the results obtained are the same. This suggests that despite the penalties, the overall lateness obtained if all containers were transported by barge on the main leg of the trip is still much larger than the cost resulting from trucking everything. This is fully due to the choice of values for the parameter ω_2. One could indeed assign lower numerical values to this weight to obtain solutions with late arrivals of commodities. However, since lateness is used mostly for computational reasons here, we will not look into those situations. In terms of the objective function used to generate the results in Table 16.1, optimising the problem for Objective 3 is the most computationally expensive at least in the case of the first three instances. However, when the number of available barges is increased to twelve, the running time of the solver for Objective 3 is much lower than for Objectives 1 and 2. One possible explanation for this is the fact that the weight $\omega_2 = 1000$ adds a large contribution to the cost solution and thus the solver begins by finding a very expensive feasible solution and then reduces it by re-assigning the commodities over the available barges. If more barges are enabled, then more capacity is available for re-assigning and transporting containers by water instead.

Overall, the results in this section provide an important insight into the computational difficulty required by the deterministic model and help us set an expectation on the numerical effort for the robust model. The largest instances that we could solve, namely k20m6 and k20m12 that have been used are comparable to the transportation activity of a real logistics service provider.

Results of Robust Model

In this section we focus on solving the robust model explained in Sect. “Robust Model”, in which the release times of the commodities belong to a predefined uncertainty set. As we already know, the robust mixed integer linear program will determine the routing of vehicles and assignment and transshipment of containers in the first stage, leaving the time variables to be determined in the second stage, when the uncertain release times have been revealed. It is expected that the robust solution is more conservative and thus, of higher cost, than the original deterministic solution. Our goal is to investigate the difference between these solutions and assess whether the ‘price of robustness’ is acceptable given the size of the instance, the level of uncertainty and the practical implications. Moreover, we would like to know what is the influence of the adjustable variables on the solution and objective function when lateness has a high and low penalty.

High Lateness Penalties

For our numerical test we will only consider an instance of manageable size, namely instance k5m12 with Objective 3. This objective is chosen because it has the highest lateness penalty and it has recorded the fastest computational time for the deterministic case, a fact which can be noticed in the last column of Table 16.1. Larger instances have not been considered due to two main reasons. Firstly, the solver would require a very large amount of time to solve them. Secondly, the simple k5m12 instance is already sufficiently diverse to allow us to study different features of the solution. In view of comparison purposes, we assume that all containers have an uncertain release date in an interval of fixed length. We consider six possible interval lengths of 2, 4, 6, 8, 10, and 12 h, which encompass scenarios ranging from small to extremely large delays. For clarification, an uncertainty interval of 2 h, for instance, suggests that the release of a container can occur 1 h before or after its nominal release value.

The solutions obtained by solving the robust model with different sizes for the interval uncertainty are given in Table 16.3. Some remarks are in place concerning the last two columns of this table. When using robust optimisation, it is important to assess what is the 'price of robustness', namely what is the additional cost to be paid when immunising the solution with respect to the uncertain parameters. In order to do that, we have also considered the situation when the release times are already available at the beginning of the planning and solved the deterministic problem for two different realisations: the best case, in which every container is released at the earliest opportunity ($\zeta = -1$) and an 'average' case ($\zeta = 0$), when the release times occurs at the midpoint of the uncertainty interval. Then we calculated by how much the robust cost increases from the deterministic solution for both cases, and displayed those values in Table 16.3. As we expected, the cost of the robust solution increases as the size of the interval of the release time is enlarged. Moreover, we observe that there is a certain amount of delay that the planning can handle. Namely, for instances with a release delay within 3 h, the solution attained is identical to the one obtained by solving deterministic model with nominal release time values.

Table 16.3 Objective value of the robust model, gap between the current solution and the best lower bound found so far, computation time (CPU seconds), percentage increase from the deterministic objective for $\zeta = -1$, and percentage increase from the deterministic objective for $\zeta = 0$. * an upper bound of 36,000s was set to the execution time of the solver

Data instance k5m12	Robust solution	Gap(%)	Runtime(s)	Increase best case	Increase 'average' case
2 h interval	100	0	198.63	0 %	0%
4 h interval	100	0	1854.22	0%	0%
6 h interval	100	0	379.25	0%	0%
8 h interval	1415	8.13	36,000*	1315%	8.84%
10 h interval	1465	11.26	36,000*	1365%	12.69%
12 h interval	1595.5	18.52	36,000*	1495.5%	22.73%

Table 16.4 Transportation characteristics of the robust planning

Data instance k5m12	Number of commodities fully trucked	Number of barges used
2 h interval	0	6
4 h interval	0	5
6 h interval	0	5
8 h interval	1	2
10 h interval	4	1
12 h interval	3	2

Table 16.5 Statistics concerning the number of time variables that are adjusted by the affine rules. A random realisation $\zeta \neq 0$ was used for checking whether the time variables have adjusted or not

Data instance k5m12	Number of adjusted time variables	Number of adjusted arrivals	Number of adjusted departures
2 h interval	7	3	4
4 h interval	0	0	0
6 h interval	3	1	2
8 h interval	0	0	0
10 h interval	0	0	0
12 h interval	10	5	5

Regarding the ‘price of robustness’, we see that for a data realisation at the midpoint of the uncertainty set (not a favourable situation), the difference between the robust solution and the best deterministic solution is at most 22.73% (corresponding to a ±6 h margin for delay). The objective value for the robust solution alone is not sufficiently insightful to assess how the transportation changes when the uncertainty interval increases. To give a measure of this, we include the number of commodities that are transported only by truck and the utilisation of barges in Table 16.4. It is apparent from this table that when commodities are released with significant delay (larger than 3h), it becomes impossible to transport them by barge. However, the uncertainty intervals that we considered were still not sufficiently large to enforce a transportation plan with no barge being utilised.

The numbers of adjusted time variables for every uncertainty interval are highlighted in Table 16.5. From here, we can immediately notice that the affine rules that were proposed indeed induce the adjusting of arrival and departure times. However, it is striking that for uncertainty intervals of 4 h and 8 h, no adjustment occurs. Another important observation from Table 16.5 is that, in the case when adjustment occurs, it does not affect all time variables which were assigned a numerical value in the solution.

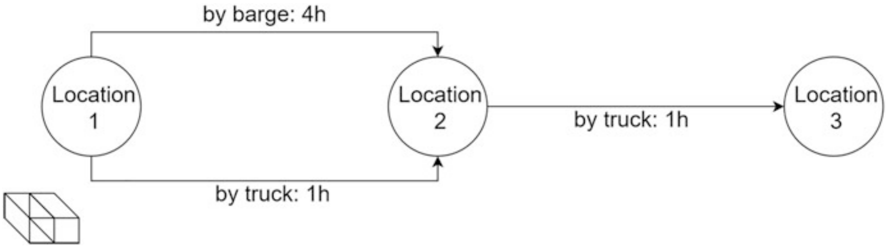

Fig. 16.3 Example of a transportation network with three nodes and two commodities

Low Lateness Penalties

When the lateness penalties are assigned sufficiently small numerical values, the adjustable variables can directly influence the value of the objective function. In order to illustrate this idea we have repeated the test from the previous section using Objective 1 (having a lateness penalty of 0.1). However, due to the high computational effort required by the solver, we were not able to record results in a reasonable amount of time as to include them in this report. Nevertheless, we will demonstrate the benefits of adjusting time variables using a simple example which still incorporates all the transportation elements that were shown by the other generated instances. Consider the transportation network in Fig. 16.3. We assume that there are two containers to be released at Location 1 in the interval [8,10]. One needs to arrive at Location 2 by time unit 14 whilst the other has the same due time but it is destined for Location 3. It is assumed that there are sufficient barges and trucks to carry out these transport requests.

For this particular instance, the worst-case solution given by the robust model in Chap. 5 with Objective 1 has cost 1.2. This corresponds to the situation when these containers are released at time unit 10. Both containers will be taken by barge to Location 2, and from there one will be trucked to its final destination, where it records a lateness of 2 h. However, if the release occurs at time unit 8, then the adjusted cost will be of only 1 cost unit, since there is no lateness recorded at Location 3 anymore.

This result also shows that unlike in the results shown in the previous section, in this case it is possible to adjust the time variables along the routes which include a transshipment. This is due to the fact that we have included in the model constraints of the form

$$A^m_{i,k} \geq \lambda, \quad \forall i \in V, k \in K, m \in M, \tag{16.53}$$

where λ is some small scalar (chosen to be 1 in this example). These ensure in this case that the affine coefficients of the adjustable arrival times will be non-zero. We see that by including these constraints in the model, we can guarantee the adjustment of all time variables.

This example demonstrates that the affinely adjustable robust optimisation framework can be used to obtain improved solutions for the freight routing problem with a high tolerance for lateness. However, the choice for the lateness parameter as well as for the scalar λ is instance-specific. Therefore, at this stage of the research, it is difficult to make assertions about how the robust model can be used for any general instance.

Discussion

In this section, we discuss the most important findings from the numerical experiments and where necessary, provide more insight into the results obtained by closely inspecting the solutions. The deterministic formulation of the freight routing can be successfully solved exactly with the branch-and-bound method for instances as large in size as those including twelve vehicles and twenty commodities. These instances are comparable to what is encountered in practice. However, the relatively large computational time required by the solver is likely due to two main factors, namely the large amount of (binary) decision variables in the model as well the many symmetries of the problem. Given the results in Table 16.1, we see that allowing for lateness with high penalties yields a model which can be solved faster than the original version with hard deadlines.

First of all, the computational results of the simple instance showed that we can obtain robust feasible solutions for the container freight routing problem by solving exactly the robust counterpart. These solutions correspond to transportation plans that can be carried whenever the release time of a container falls within a pre-specified interval. An increase in the size of this uncertainty interval induces higher solution costs, since containers which have a short delivery span will not be transported by barge. This results is fully confirmed by the data in Table 16.2. However, we see that the price to be paid for the robustness of solutions is quite high. As an example we consider the k5m12 case with an 8 h interval. When we assume that the release times of the containers can deviate from their nominal value by 4 h, and they in fact are released on the earliest time possible, the transportation plan obtained is 1315% more expensive than the plan that could have been achieved if all data was known beforehand. If we assume a less favourable realisation, in which half of the commodities are released at their nominal value, and the other half at the latest time possible, the increase is only of 8.84%. Given the lack of information on the real-life situation, therefore difficult to assess if the price of robustness is acceptable when modelling highly uncertain releases for the containers. However, we can state that in a practical instance in which one can infer from historic data that parameters often attain ‘bad realisations’, the robust transport plan can be employed in exchange for a reasonable cost increase.

The price of robustness can also be regarded from a slightly different perspective. We consider a situation in which given some uncertainty intervals, one makes a deterministic plan assuming a certain nominal value for the releases. If the actual realisation of the parameters is worse than the nominal values, then the deterministic solution is likely to be infeasible. This enforces re-planning of the current vehicles and container assignment. Although there are many ways in which one can re-route, the newly obtained transportation solution might have a higher cost than the robust solution that could have prevailed over the delays.

We have applied the affinely adjustable robust optimisation framework in order to allow the arrival and departure times of the vehicle to change according to the realisation of the container release and induce some degree of flexibility in the planning. Whilst the objective value of a solution remained unchanged, due to very high lateness penalties, some vehicles might be able to arrive or depart earlier at certain locations. Concerning the actual adjustment of variables, it was at first surprising to notice the low proportion of time variables are affected by the changes in data realisation. However, at a closer inspection of the container routes given by the solutions we were able to find a possible explanation for this. We have found that adjustment is only effective for arrival and departure times on a particular kind of route. In other words, for direct routes, on which a commodity is shipped from its origin location to their destination by means of a single vehicle, adjustment takes place. Otherwise, if a switch of vehicle, transshipment or additional commodity pickup occurs on the way, then only arrivals and departures that can be adjusted are those at the beginning location of the route. When inspecting the solution, we have found that the instances k5m12 with 4 h, 8 h and 10 h intervals, in which no adjustment took place, indeed included no direct routes. Moreover, adjustment seemed to be particularly successful for the case with the highest uncertainty interval. Essentially, since all commodities are directly trucked from origin and destination in this case, all the arrivals and departures are adjusted.

It is difficult to further explain why adjustment only affects direct routes, but it might be related to the fact that the vehicle synchronisation and transshipment constraints in the robust model force the affine coefficients of the adjustable variables to take the value zero. This suggests that on non-direct routes it is difficult to ensure adjustment with respect to all the decisions made on that route: transshipment, vehicle switch, loading or unloading of commodities. Another factor which may influence adjustment is the symmetry of solutions and the fact that the same objective can be achieved by many different routes. For instance, there are solutions in which a barge is assigned for every commodity, resulting in significantly less transshipments, which have the same objective as a solution with a smaller barge utilisation. Nevertheless, we were able to produce an example in which the robust model with an additional sets of constraints gives a fully adjusted solution, which is cheaper than the worst-case scenario, if the data assumes a favourable realisation. Therefore, we have shown that the adjustment of variables can result in a direct improvement of the objective function. For a generalisation of this result, a sensitivity analysis of the instance parameters on the lateness term in the objective is required.

References

1. Agra, A., Christiansen, M., Hvattum, L. M., & Rodrigues, F. (2018). Robust optimization for a maritime inventory routing problem. *Transportation Science, 52*(3), 509–525. https://doi.org/10.1287/trsc.2017.0814.
2. Ben-Tal, A., Ghaoui, L. E., & Nemirovski, A. (2009). *Robust Optimization (Princeton Series in Applied Mathematics)*. Princeton University Press. https://www.amazon.com/Robust-Optimization-Princeton-Applied-Mathematics/dp/0691143684?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0691143684.
3. Ben-Tal, A., Goryashko, A., Guslitzer, E., & Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming, 99*(2), 351–376. https://doi.org/10.1007/s10107-003-0454-y.
4. Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research, 52*(1), 35–53. https://doi.org/10.1287/opre.1030.0065.
5. Birge, J. R., & Louveaux, F. (2011). *Introduction to Stochastic Programming (Springer Series in Operations Research and Financial Engineering)*. Springer
6. Chouman, M., Crainic, T. G., & Gendron, B. (2017). Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. *Transportation Science, 51*(2), 650–667.
7. Conforti, M., Cornuéjols, G., & Zambelli, G. (2014). *Integer programming, volume 271 of graduate texts in mathematics.*
8. Kalicharan, K. (2018). *Intermodal transport: Routing vehicles and scheduling containers*. Master's thesis, TU Delft.
9. Sharypova, K. (2014). Optimization of hinterland intermodal container transportation. Ph.D. thesis, Eindhoven University of Technology.
10. Yanıkoğlu, İ., Gorissen, B. L., den Hertog, D. (2019). A survey of adjustable robust optimization. *European Journal of Operational Research, 277*(3), 799–813.