Transport and Capacity Planning for Reusable Containers under Fixed-Step Transport Costs

Christian Krudewig

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor Rerum Politicarum

Faculty of Business Administration
Catholic University of Eichstätt-Ingolstadt
June 2015
Contents

1 Introduction
  1.1 Motivation and Overview ............................................. 1
  1.2 Management of Container Systems ................................. 2
    1.2.1 Container Systems ............................................. 2
    1.2.2 Planning Problems Raised by Container Systems ............ 6
    1.2.3 Integrated Distribution and Capacity Planning ............. 8
  1.3 Literature Review .................................................. 12
    1.3.1 Container Systems ............................................. 13
    1.3.2 Transport Planning for Reusable Containers ............... 15
    1.3.3 Capacity Planning for Reusable Containers ................. 21
  1.4 Scope and Outline .................................................. 26

2 A Basic Model for Container Transport Planning .................. 33
  2.1 Loaded versus Empty Container Movements ....................... 34
  2.2 Assumptions ........................................................ 35
  2.3 Model .............................................................. 37
  2.4 Complexity and Solution Procedures ............................. 39

3 Extensions to the Basic Model ....................................... 45
  3.1 Transport Planning Extensions .................................... 45
    3.1.1 Allowing Lost Orders ....................................... 45
    3.1.2 Allowing Delayed Deliveries ............................... 47
    3.1.3 Take-along Transports ..................................... 50
  3.2 Integrating Capacity Planning .................................... 51
    3.2.1 Purchasing and Disposing of Containers ................. 52
    3.2.2 Container Leasing ........................................... 54
## 5 Summary and Outlook

### Appendix A: Data and Results of the Calculations in Section 4.4

<table>
<thead>
<tr>
<th>A.1 Data</th>
<th>119</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1.1 Network 1: Distribution Network</td>
<td>119</td>
</tr>
<tr>
<td>A.1.2 Network 2: Hub-and-Spoke Network</td>
<td>121</td>
</tr>
<tr>
<td>A.1.3 Network 3: Procurement Network</td>
<td>122</td>
</tr>
<tr>
<td>A.1.4 Network 4: Complex Network</td>
<td>124</td>
</tr>
<tr>
<td>A.2 Optimal Solutions</td>
<td>125</td>
</tr>
<tr>
<td>A.2.1 Network 1</td>
<td>125</td>
</tr>
<tr>
<td>A.2.2 Network 2</td>
<td>126</td>
</tr>
<tr>
<td>A.2.3 Network 3</td>
<td>127</td>
</tr>
<tr>
<td>A.2.4 Network 4</td>
<td>128</td>
</tr>
<tr>
<td>A.3 Test of Candidate Selection and Branching Rules</td>
<td>129</td>
</tr>
<tr>
<td>A.4 Results for the Test of the Additional Constraints</td>
<td>130</td>
</tr>
</tbody>
</table>

## Bibliography

131
List of Figures

1.1 A container cycle. ......................................................... 3
1.2 Container flows in differently structured networks. .............. 5
1.3 The hierarchy of planning tasks for container systems. .......... 8

2.1 A model for minimising transport and inventory costs. .......... 38
2.2 The minimum cost flow problem. ................................. 40
2.3 A static network and the corresponding time-expanded network. 41
2.4 The basic model as time-expanded network. ..................... 42

3.1 Graph representation of lost orders with an additional node providing fictional supply. ............................................. 47
3.2 Some occurrences of backorders in the graph representation of the model. ............................................................. 49
3.3 Graph of the basic model with three additional arcs for take-along transports. ............................................................ 51
3.4 An example network with purchase and disposal of containers. . 54
3.5 A network with leasing of containers. ............................... 56
3.6 Example network 1 – locations, distances and container demands. 58
3.7 Example network 2 – locations, distances and container demands. 58

4.1 Fixed-step transport costs. ............................................. 71
4.2 Costs per unit with fixed-step transport costs. .................... 71
4.3 Continuously decreasing transport cost function. .................. 72
4.4 Transportation planning under fixed-step costs (non-linear version). 76
4.5 Transportation planning under fixed-step costs (linear version). . 78
4.6 Example network – locations, distances and container demands. 81
List of Figures

4.7 Subsets of nodes focused by the feasibility conditions. . . . . . . . . . 86
4.8 Two paths between the same source and destination node. . . . . . . . 91
4.9 The branch-and-bound algorithm. . . . . . . . . . . . . . . . . . . . . . 97
4.10 The branch-and-cut algorithm. . . . . . . . . . . . . . . . . . . . . . . . 105
4.11 Average calculation time after adding different conditions. . . . . . 111
4.12 Calculation time with and without the rounding heuristic. . . . . . 112
## List of Tables

1.1 Survey papers on topics related to container systems. .................. 13
1.2 Publications on container systems in general. ...................... 14
1.3 Overview of the literature on transport planning. .................. 16
1.4 Overview of the literature on capacity planning. ................. 22
1.5 Overview of all model variations in this thesis. ................... 31

2.1 The symbols used for the basic model. .......................... 35

3.1 Additional symbols used for the transport planning extensions. ... 46
3.2 Additional symbols used for the capacity planning extensions. ... 52
3.3 Results for Example 1. ........................................... 59
3.4 Initial stock at the network nodes in Example 2. .................. 61
3.5 Results for Example 2. ............................................ 62
3.6 Results for Example 3. ............................................ 64
3.7 Results for Example 4. ............................................ 66

4.1 Additional symbols used for the model with fixed-step transport costs. 70
4.2 Results for different vehicle capacities. .......................... 82
4.3 Number of variables in the model with fixed-step costs. ......... 85
4.4 General parameters for the example networks. ..................... 107
4.5 Branching and candidate selection strategies of the branch-and-bound algorithm. ................................. 108
4.6 Optional improvements of the branch-and-bound algorithm. ...... 108
4.7 Results of the test of the candidate selection and branching rules. 109
4.8 Number of constraints added to the optimisation model. ......... 110
4.9 Results of the test of the cutting rules. .......................... 114
List of Tables

A.1 Optimal transports in the distribution network example. . . . . . . 125
A.2 Optimal stock holding in the distribution network example. . . . 126
A.3 Optimal transports in the hub-and-spoke network example. . . . 126
A.4 Optimal stock holding in the hub-and-spoke network example. . . 127
A.5 Optimal transports in the procurement network example. . . . . 127
A.6 Optimal stock holding in the procurement network example. . . . 128
A.7 Optimal transports and stock holding in the complex network example 128
A.8 Detailed results of the test of the candidate selection and branching rules. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 129
A.9 Detailed results of the test of the additional constraints. . . . . 130
1 Introduction

1.1 Motivation and Overview

Many companies use reusable containers to store, transport or handle products. Under good conditions this can help to reduce costs and improve the ecological footprint of logistic operations because containers can be used many times before they have to be replaced. On the other hand, the acquisition, maintenance and distribution of the containers need to be planned. Also additional transport and storage capacity is needed to provide the necessary containers at a certain place and time.

This research considers a logistics network and a fleet of reusable containers. It aims at developing mathematical models to support the short- to medium-term planning tasks which comprise of planning container transports and necessary capacity adaptations to the container fleet. These two aspects are combined into an integrated optimisation process in order to account for the trade-off between more transports and a larger fleet. The models developed in the following optimise container movements and storage as well as capacity measures such as leasing, acquisition and disposal. In order to avoid unnecessary tours of transport vehicles, two further aspects are allowed for: take-along opportunities on the one hand, and non-linear costs on the other. These are particularly important for transports of reusable containers and especially the latter aspect is not covered sufficiently by the existing literature. The models are designed as a flexible tool to be integrated into a decision support system.

The thesis is organised as follows. The remainder of this first chapter introduces the main terms and planning problems, reviews the existing literature and then
1 Introduction

outlines the chosen approach. It also gives an overview of the developed models. In Chapter 2, a basic model for container distribution planning is introduced to provide an elementary design. The subsequent Chapters 3 and 4 show how this simple model can be extended to cover different aspects of container transportation and capacity planning. Chapter 4 introduces a model with a changed cost structure for transports, namely fixed-step transport costs. The resulting model is an integer program that cannot be solved in acceptable time with standard solvers. Therefore a branch-and-cut method is presented to solve the problem. The findings and limitations of this research are summarised in Chapter 5.

1.2 Management of Container Systems

At the beginning of this thesis, it is necessary to define the key terms such as ‘container’ and ‘container system’, as they are used in the following. This is done in this section. Further on, the object of investigation is defined, which means that the main characteristics of container systems are outlined as well as the associated planning tasks. The last subsection then focuses on distribution and capacity planning for containers, which are the two planning tasks that are covered by the remainder of the thesis. The characteristics of these tasks are outlined and it is explained why it can be valuable to integrate both planning tasks into one decision.

1.2.1 Container Systems

As a first step the most central terms are explained and defined. These are ‘container’, ‘container networks’ and ‘container systems’.

Containers

This thesis focuses on reusable containers which can be used for transporting, storing or processing goods. A reusable container is defined here as a device to carry products that can be used repeatedly without reconditioning (see Arnold,
1.2 Management of Container Systems

_Isermann and Kuhn_ 2008, p. 418; _Hofmann and Bachmann_ 2006, p. 18 for similar definitions). Some examples of such containers are pallets, skeleton containers, barrels, swap bodies or maritime containers. Although such containers can have very different physical forms, they are still similar in the planning problems they generate when a company decides to use them. They circulate through a company's supply chain and are loaded, used, unloaded, cleaned, repaired and stored over and over again as shown in Figure 1.1.

![Figure 1.1: A container cycle.](image)

Non-returnable packaging as well as recyclable wrapping are not regarded in the following. Returnable packaging that is used to deliver goods to customers, such as returnable bottles, also has other characteristics. It cycles through an open network which means it is not under the control of one company or decision maker. It is comparatively cheap and must be reconditioned after each use.

**Container Attributes**

In general, reusable containers differ within a set of attributes. An extensive overview is given by Wildemann (1995, p. 5) and Arnold, Isermann and Kuhn (2008, p. 701). The most important attributes in this context are:

- Capacity, i.e. the maximum size or weight of goods that can be carried,
- compatibility with storage or loading devices and transport vehicles,
- suitability for different goods,
1 Introduction

- costs for acquisition and maintenance,
- durability,
- time and effort for cleaning and maintenance,
- costs and required capacity for storage and transport,
- substitutability with other types of containers.

These attributes have to be regarded when planning the acquisition and distribution of containers. Therefore they are used below to classify possible planning models.

Network Structure

The main difference between reusable containers and one-way packaging is that the latter is disposed of after unloading while the former are used again. Therefore they have to be transported to the locations where they are needed next. They also possibly have to be cleaned or repaired. So reusable containers do not only move loaded with goods but also separately. These additional movements, also referred to as empty flows, have to be planned. The locations where the containers are loaded, unloaded or delayed for some time, form a network. The network nodes are connected by transport routes, each with its own type of transport vehicle, capacity, costs and transport time. Between two nodes different connections may exist, for example one connection by lorry and another by train. If the containers are used by more than one company, then of course the network also extends to several companies.

Four different types of network structures may occur (see Figure 1.2). As the goods are shipped in containers which have to be sent back empty, the container flows are opposite to the flows of goods. The first type (a) is a distribution network where goods are distributed to different locations from a central plant or warehouse. In such a system, the containers become available at their destinations from where they are returned to the warehouse. An opposite system results from a procurement network (b). Automotive manufacturers for instance deliver empty containers to their suppliers, where they are loaded with components and returned to the
1.2 Management of Container Systems

(a) Distribution network

(b) Procurement network

(c) Hub-and-spoke network

(d) Complex network

Figure 1.2: Container flows in differently structured networks.

automotive plant. The third type is a hub-and-spoke network (c) where all flows
of goods are routed through a central hub. This means that container flows may
occur in both directions from the central position to the satellites or the other way
around. Such a structure also arises if the containers are kept in a central pool. In
these cases, they are sent to other locations if they are needed and are returned
after they are unloaded. The last type of network (d) is a complex one where the
container flows do not follow a specific structure but occur arbitrarily between all
network nodes.

Container Systems

The set of network nodes, their connections and the circulating containers are
defined here as a ‘container system’. So, all locations where the containers under
consideration could move to, are part of the container system, all others are not.
Such a system can be a closed network where containers are unable to enter or
leave the system during their life cycle. But open networks exist as well, where containers enter or leave all the time. One example is the system of Euro-pallets, another would be the market of maritime containers. For the purpose of this thesis, it is only important that the scope of the container system matches to the area of influence of the planner. It is assumed that within the container system the decisions to buy, sell or move containers are in the hands of only one central authority, for instance one company.

1.2.2 Planning Problems Raised by Container Systems

Overview

In this section the planning tasks that arise in connection with container systems are explained. The overall objective is to create a container system with sufficient transport capacity to allow loading, transporting and unloading goods at a high speed. Furthermore low capital expenditure and operating costs should be reached. Also technical requirements have to be met as best as possible. This thesis focuses on quantifiable management decisions, while other aspects such as technical parameters are considered to be predetermined. The reason for this is, that the mathematical tools available for optimising such a decision can only deal with quantifiable aspects.

The planning tasks to be performed in a container system can be classified by the decision variables that are involved as illustrated in Figure 1.3 (for similar classifications of the planning tasks see Fleischmann 2005 pp. 233–235; S.-W. Lam, L.-H. Lee and Tang 2007 p. 267). These tasks are explained in the following paragraphs.

Long-Term Level

On the strategical long-term level, so for the next years, the number and locations of container specific network nodes have to be defined. For example, the question is to be answered, where and how many depots, cleaning and repair facilities are needed. On the same level, the transport system is to be designed. This includes
choosing the system type and the connections between the network nodes. The aim is to provide sufficient capacity for the moving, handling and storing of containers with minimum transport costs for full and empty movements.

### Medium-Term Level

In the medium term (that means for the next few months), two decisions must be made: which, and how many, containers to acquire. This is the task of capacity planning. Apart from differing costs, the type of containers is mainly determined by technical parameters such as the ease of handling or compatibility with the goods and existing facilities. In addition, appropriate sizes of containers have to be chosen to allow acceptable batch sizes of goods. The question how many containers of the selected types are needed is crucial. From the capacity of the container system results a capacity restriction for each process which relies on the reusable containers. This could mean that the number of available containers limits the amount of goods that can be stored, transported or even produced. Therefore a sufficient number of containers is very important. However, too many containers lead to higher costs and organisational problems. For example empty containers occupying too much space on the shop floor in a factory could interfere with production. So the question is, how many containers are needed to achieve the plans for processing, storage and transportation of goods. Also, how many of them should be purchased directly and how many should be rented on demand. The objective is to minimise the costs to reach a given service quality.

### Short-Term Level

The short-term level ranges from the next hours until the next few weeks. On this level the distribution of containers must be organised. This task is referred to as ‘distribution’ or ‘transport planning’. If the network forms a closed system, it is also called ‘fleet management’. It means deciding where the containers should be transported to or where they should be stored. In this respect, also the frequency of transports and the capacity of the vehicles must be considered. The decision about container movements is based on distribution plans for the goods to be carried. From these, the movements of loaded containers can be derived directly. What
remains to be scheduled are the movements of empty containers. Every planned usage of these containers can be seen as an order. It is to be fulfilled by delivering the needed quantity of empties, either from the company’s own pool or by renting an additional amount.

Figure 1.3: The hierarchy of planning tasks for container systems.

Summary

These three planning levels lead to a hierarchy of planning tasks as shown in Figure 1.3. Each step generates data that has to be regarded as fixed in the more short-term levels. First, all nodes and connections are defined, then the types of containers in the systems are chosen. Next the number of containers to buy, rent or lease is specified and lastly the movements of the given containers are planned.

1.2.3 Integrated Distribution and Capacity Planning

Integrated Planning

In the previous section capacity planning was classified as a mid-term decision, and distribution planning as an operational one. However, these two tasks can be integrated. The rationale for this is, that they are not independent of each other.
1.2 Management of Container Systems

When planning the amount of containers to acquire, it must be known how fast the containers would circulate through the network. But this speed is a variable in itself and is determined by the operational transport schedules.

So it is reasonable to combine capacity and transport planning in some situations. This is the case if a trade-off between acquisition and transport of empty containers exists. If it is possible to rent or buy additional containers, it is also possible to substitute containers that have to be transported from their current position to the location where they are needed. On the other hand additional transports can compensate missing containers to a certain degree. Containers can be moved between several locations where they are needed, instead of buying more containers and positioning them at every location. These alternatives become even more apparent when the vehicle utilisation is regarded. With more containers in the system it is less often necessary to move them on partially loaded vehicles, which reduces the total transport costs. At the same time, the additional containers would inevitably lead to an increase in acquisition costs.

This leads to the conclusion that the most cost-efficient compromise can be determined if transportation and acquisition of containers are both considered in one optimisation model. Depending on the particular planning environment, it should be evaluated to use models which integrate capacity planning and transport planning aspects.

However, an integrated planning approach is only reasonable if the two planning results can be implemented within a comparable period. Thus, many authors only include short-term capacity measures such as leasing or backordering into their distribution models (see the literature review in the next section). Which particular controls are available on the short-term level is an individual aspect of each container system.

**Model Classification**

Possible capacity and distribution planning models can be classified by the following set of characteristics which are explained in more detail below:
1 Introduction

- Decision: capacity planning, distribution planning or both
- Time horizon: static or dynamic
- Demand and supply of empties: deterministic or stochastic
- Transport duration: deterministic or stochastic
- Transportation batch size: individual shipment, fix-sized transports or variable-sized transports
- Container types: only one or several types

First of all, the scope of the decision is to be defined. Sizing and distribution of a fleet of containers do not necessarily have to be planned simultaneously. Thus, models can be distinguished by the type of decision they make. Combined models can be used to find the optimal trade-off between transportation and acquisition costs.

Regarding the time horizon, two options exist. A dynamic horizon can be chosen to plan the development of a container system over time, while models in static time are suited to generate plans for systems that tend to remain in a steady state with not too many changes in the amount of containers needed. Another application for static models is rough long-term capacity planning. For this task, static stochastic models are suited. Dynamic models need more detailed data and the expectations of the changes over time must be expressed explicitly. In turn also more detailed plans are generated as they are needed especially on a short-term level.

The demand and supply of containers are either modelled stochastically or are assumed to be deterministic. The latter option disregards the often stochastic behaviour of real systems. But especially in dynamic models, stochastic planning is much more complex and it can also be difficult to estimate the probability distribution of the input parameters. So better results from stochastic models are tied to much more effort to collect the input data and also to calculate the optimal solution. Therefore, deterministic models are often applied to stochastic systems in a rolling horizon approach, exploiting the fact that estimates become more precise the shorter the forecast horizon is.
1.2 Management of Container Systems

A defined usage duration is clearly part of models that account for the use of containers as specific orders. In this case, the containers are bound to an order for either a deterministic or stochastic amount of time. Also the orders themselves may arrive stochastically. Other models may consider the logistic network as open (see Section 2.1 for an explanation on how these two approaches can be transferred into each other). In this case stochastic orders correspond to random departures and arrivals of empty containers in the network which appear as demand or supply.

Another aspect which is important for movements of empty containers is, if the vehicle capacity needs to be regarded, which means that the transportation batch size is to be determined. Simple models treat transport costs as linear, disregarding the fact that they are often non-linear because the transport vehicles have a capacity for batches of a certain size. This is also linked to the question of the optimal transport frequency. Some models may determine an optimal frequency which directly leads to a corresponding batch size. The most flexible types of models explicitly regard the non-linear nature of the transport costs and thus determine optimal batches of variable size.

The last aspect addressed here are container types. Basic models only consider one type of container. It is of course possible to plan for several types simultaneously. This is worthwhile if the different types interact. This is the case if they either use one common capacity, for example a common transport system or storage areas, or if they represent one capacity, because the containers of different types can substitute each other. In this case one can decide which type of container to use in a certain situation, for instance if small or large pallets should be used for a certain job.

These characteristics determine the structure of the decision models for container transport or capacity planning. The classification is used in the next section to organise the literature review.
1 Introduction

1.3 Literature Review

A wide range of research topics is related to transport and capacity planning of reusable containers. But only little research has been dedicated directly to reusable containers in general. Most published work concerns special types of reusable containers or other types of resources that behave similarly, for example railway waggons, maritime freight containers, rental cars or other transport vehicles. Therefore, this section gives a short overview of the literature in these areas. It does not aim to cover each topic exhaustively but to show the most important research fields and to reference the key literature therein.

This literature review is split into three parts. The first section summarises the existing literature about container systems and especially the adherent planning problems. Section 1.3.2 then focuses on transport planning literature and outlines the research devoted to optimising the transportation of containers or similar items through a network. As the models presented throughout this thesis also cover capacity planning topics, an additional literature section (Section 1.3.3) then summarises the relevant research for capacity planning. The approaches listed in sections 1.3.2 and 1.3.3 are classified on three levels, according to the type and attributes of the models used:

1. Routing or distribution models (transport planning) versus those focusing on fleet sizing (capacity planning),

2. static versus dynamic models and

3. deterministic versus stochastic models.

Here, some of the capacity planning models also include transport planning aspects. These are the same model characteristics as discussed in the previous section.

Former research has already produced several review articles, mostly referring to a certain direction of research. The papers dedicated to a particular type of container rather than a certain planning task are mentioned here first (see Table 1.1 for an overview).
Table 1.1: Survey papers on topics related to container systems.

Dejax and Crainic (1987) provide a well-structured overview of the research regarding flows of empties. It is a good introduction to fleet management literature up to the late 1980s (‘fleet management’ in this context means the distribution of empty vehicles). Approaches for railway wagons, maritime containers as well as for the haulage industry are covered. An update is available by Crainic and Laporte (1997), summarising not only the literature itself but also outlining the different classes of mathematical models. Focusing on routing models, Hoff et al. (2010) show the different variations of the ‘vehicle routing problem’ and the ‘fleet size and mix vehicle routing problem’. Concerning the vehicle routing problem, many more surveys are available that are referenced below.

For a broad overview of research for the railway industry, one may refer to Cordeau, Toth and Vigo (1998). Particularly freight wagon management models are described therein (see pp. 389–391). A comprehensive overview of the literature regarding the repositioning of intermodal standard containers is given by Song and Dong (2015). Also Steenken, Voß and Stahlbock (2004) and Stahlbock and Voß (2008) concentrate on this area, but focus more on terminal operations in seaports.

1.3.1 Container Systems

The first subsection of the literature review summarises the literature about container systems and about reusable containers in general. The publications mentioned below are listed in Table 1.2 on the following page.
1 Introduction

<table>
<thead>
<tr>
<th>Publication</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hofmann and Bachmann 2006</td>
<td>Study on container management in practice</td>
</tr>
<tr>
<td>Kroon and Vrijens 1995</td>
<td>Returnable containers as an example of reverse logistics</td>
</tr>
<tr>
<td>McKerrow 1996</td>
<td>Reusable containers in the automotive and retail industries</td>
</tr>
<tr>
<td>Twede and Clarke 2004</td>
<td>Organisation of container systems</td>
</tr>
<tr>
<td>Wildemann 1995</td>
<td>Characteristics and design of container systems</td>
</tr>
</tbody>
</table>

Table 1.2: Publications on container systems in general.

*Hofmann and Bachmann (2006)* show results from a survey of 180 central European companies. They show in detail, that an increasing number of companies employ reusable containers, explain the reasons for this and the resulting challenges. A notable finding of this study is, that the container systems are mostly controlled manually, partly without support of IT utilities. On the other hand, they are shown as the cause of a substantial part of the logistics costs. In some cases insufficient supply of empty containers also contributes to out-of-stock costs.

Challenges related to managing container systems are discussed by *McKerrow (1996)* and *Twede and Clarke (2004)*. The former article analyses the implications of different ownership structures for a fleet of containers. The author focuses on the problem of containers not being returned or being lost. Reasons for this are identified and general recommendations are given how to set incentives, so that the containers actually get reused. *Twede and Clarke (2004)* show trends and current practice of reusable container systems in US and UK. They discuss problems in the management of reusable containers and which factors determine whether they can be employed successfully and with cost benefits.

Only little conceptual work has been done on container systems as a whole, giving an overview of the planning and management processes in container systems. One approach is that of *Wildemann (1995)* who analyses the cyclic systems reusable containers circulate in, categorising types and attributes of containers, listing their advantages or disadvantages. He identifies the problem of capacity planning and
1.3 Literature Review

develops a simulation framework to support the choice of the right type and number of containers. Kroon and Vrijens (1995) distinguish different structures of ownership and responsibility for container systems and discuss their implications on logistics. In a case study with a central agency managing the system they show four planning problems: how many containers to acquire, where to establish depots, how to organise the container distribution and how to define appropriate service fees for the central agency. Solutions are proposed for the first two tasks, the depot location and the fleet sizing decisions.

1.3.2 Transport Planning for Reusable Containers

A large volume of literature has been published on topics related to transport planning. These range from distribution models for empty maritime containers to fleet management models for vehicles, especially for lorries and freight wagons. This section focuses on the approaches which can be adapted to unspecific containers. Specialised concepts, which are not usable for the problems faced in this thesis, are beyond the scope of this review. The research discussed in this section is listed on the next page in Table 1.3.

Static and Deterministic Transport Planning

Static and deterministic models are not well suited for the repositioning of empties as they can only depict a one-level snapshot of a logistics network. However, Allman (1972) proposes applying the classic transportation problem to the relocation of freight waggons. The research quickly extends to multiperiod models as shown below in the paragraph describing dynamic deterministic models.

Static and Stochastic Transport Planning

Static and stochastic models such as inventory models can also be utilised for the redistribution of items between multiple locations. Dang, Yun and Kopfer (2012) use static inventory policies to redistribute empty containers among several
### 1 Introduction

<table>
<thead>
<tr>
<th>Publication</th>
<th>Time</th>
<th>Data</th>
<th>Planning object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuja, Magnanti and Orlin 1993</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>(Network flows)</td>
</tr>
<tr>
<td>Allman 1972</td>
<td>Static</td>
<td>Deterministic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Aronson 1989</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>(Network flows)</td>
</tr>
<tr>
<td>Bourbeau, Crainic and Gendron 2000</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Cheung and Chen 1998</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Cheung and W. B. Powell 1996</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Choong, Cole and Kutanoglu 2002</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Cordeau, Laporte et al. 2007</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Crainic, Gendreau and Dejax 1993</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Crainic and Laporte 1998</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Dang, Yun and Kopfer 2012</td>
<td>Static</td>
<td>Stochastic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Erera, Morales and Savelsbergh 2005</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Erera, Morales and Savelsbergh 2009</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Generic</td>
</tr>
<tr>
<td>Feng and Chang 2008</td>
<td>Static</td>
<td>Stochastic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Fisher 1995</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Francesco, Crainic and Zuddas 2009</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Francesco, Lai and Zuddas 2013</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Frantzeskakis and W. B. Powell 1990</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Glickman and Sherali 1985</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Godfrey and W. B. Powell 2002b</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Godfrey and W. B. Powell 2002a</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Jordan and Turnquist 1983</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Jula, Chassakiakos and Ioannou 2006</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Kikuchi 1985</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Laporte 1992</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Laporte, Gendreau et al. 2000</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Laporte and Osman 1995</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Marinakis and Migdalas 2007</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Moon, Ngoc and Konings 2013</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Olivo et al. 2005</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>W. B. Powell 1986</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Shen and Khoong 1995</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Topaloglu and W. B. Powell 2006</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Toth and Vigo 2002</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>White 1972</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>White and Bomberaault 1969</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Yun, Y. M. Lee and Cho 2011</td>
<td>Static</td>
<td>Stochastic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Toth and Vigo 1998</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vehicles</td>
</tr>
</tbody>
</table>

Table 1.3: Overview of the literature on transport planning.
1.3 Literature Review

depots. Cost-optimising parameters are determined with simulations and genetic algorithms. A similar approach is taken by Yun, Y. M. Lee and Choi (2011). Also Feng and Chang (2008) use an inventory model to determine the desired safety stock of empty containers at different ports. They use this as the first step in a two-stage optimisation. For the second step, a dynamic and deterministic repositioning model is used.

**Dynamic and Deterministic Transport Planning**

For the distribution of empty freight wagons as well as empty maritime containers, several authors propose linear programming models to solve a problem formulation based on a time-expanded network. Aronson (1989) surveys the basic literature on this class of models, also referred to as dynamic network flow models. The key models and algorithms are also explained very well in Ahuja, Magnanti and Orlin (1993).

One of the first publications to use this technique is White and Bomberault (1969). With an adapted version of the out-of-kilter algorithm (Fulkerson 1961) the total cost for moving wagons through the space-time network is minimised. It is assumed that the exact demands and supplies as well as the travel times are known.

A similar approach by White (1972) applies the transshipment model to empty containers. Here, a cost-efficient distribution of empty containers to demand nodes in a network is determined. The result is an optimal flow from nodes with an excess supply of containers to demand nodes. This model formulation is very basic, for instance it is assumed that travel times between all locations are equal to one period. This type of model is widely used as a basic setup for modelling the distribution of empty items (see Dejax and Crainic 1987, p. 231). Kikuchi (1985) adapts it to pooled freight wagons, rendering it more detailed as they allow for node-dependent storage costs and travel costs that can be chosen independently of the multiperiod travel times. The freight waggon pool concept is also addressed by Glickman and Sherali (1985). They show how the network structure can be transferred to a classic transportation problem by distinguishing between supply and demand nodes and
1 Introduction

connecting them directly, disregarding possible transshipment nodes. This allows for solving the problem with efficient specialised algorithms. A special feature of their model is, that it accounts for different types of interchangeable waggons. Such a classic transportation problem is also used in a more recent publication by Jula, Chassiakos and Ioannou (2006) to optimise the short-term redistribution of empty maritime containers in a port area on an hourly basis.

Shen and Khoong (1995) describe a decision support system for distribution of empty maritime containers that is also based on this time-expanded network approach. They show how leasing can be taken into consideration by adding artificial nodes. For solving the problem they stick to standard solvers instead of using specialised algorithms. Choong, Cole and Kutanoglu (2002) develop a similar model applied to multimodal transportation of empty standard containers. They analyse the impact of the planning horizon length on the solution for such a model. They show that the solution can be subject to an end-of-horizon effect, especially if strong activities follow directly after the end of the planning horizon and are thus not within the scope of the optimisation. Another important factor is the travel time compared to the length of the planning horizon. If it is rather long, extending the planning to further periods changes the solution considerably. In the model developed by Erera, Morales and Savelsbergh (2005), the node demands are no longer defined for fixed periods but have to be served within time windows. In addition, their approach integrates planning the routing for both, empty and loaded containers. Olivo et al. (2005) provide a dynamic transshipment model for the multimodal transportation of empty cargo containers, optimised with a period length of only one hour for a planning horizon of one week. They show that also large instances of these models can be solved efficiently. A contrary result is presented by Moon, Ngoc and Konings (2013) who integrate the planning for foldable and standard cargo containers into one linear repositioning model. They develop a heuristical solution approach as they encounter performance issues with standard solvers. However, they used very large problems with more than 100 ports and 50 planning periods. These problems could still be solved within about 30 minutes in the worst case. This confirms, that this type of models can be solved within a time frame which is viable for many applications.
1.3 Literature Review

Bourbeau, Crainic and Gendron (2000) develop a multi-commodity transshipment model for empty container flows. The linear flow model is combined with a depot location problem. Also, the authors show an interesting approach for parallelising a branch-and-bound problem.

Extensive research exists regarding the vehicle routing problem (see Bolduc, Renaud and Montreuil 2006, p. 211; Dejax and Crainic 1987, p. 236), a multiperiodic, deterministic model to optimise the movement of vehicles through a network, so that a set of transport tasks are performed in a cost-efficient manner. This problem class is mentioned here, because it could, in principle, also be used to optimise the movements of empty containers. The main difference to the transshipment class of models is that vehicle routing problems are designed to create routes for each individual item instead of only moving a flow of identical items. This results in a large number of integer variables and therefore a higher complexity.


Exact solution algorithms were developed for example by Toth and Vigo (1998) who also proved the problem to be NP-hard. As a result more efficient branch-and-cut algorithms as well as heuristic solutions have been developed. An overview of solution algorithms is provided by Laporte (1992), Fisher (1995), Laporte, Gendreau et al. (2000) and Cordeau, Laporte et al. (2007).

Further research has been dedicated to generalisations of the vehicle routing problem such as a heterogeneous fleet or time windows setting limits for transports. These extensions are also described in the aforementioned reviews.

Dynamic and Stochastic Transport Planning

Based on the popular dynamic deterministic transshipment models for the repositioning of freight waggons or cargo containers, several authors propose a stochastic integer minimum cost flow model. Often variants of the Frank-Wolfe gradient
1 Introduction

algorithm (Frank and Wolfe [1956]) are used to solve these models. As the resulting stochastic minimisation problems are not necessarily convex, this method cannot guarantee finding the global optimum. Therefore, several authors develop alternative approaches. Jordan and Turnquist [1983] suggest a dynamic network model for the distribution planning of freight wagons. Supply and demand at the yards, as well as the travel time, are assumed to be stochastic. The objective is to maximise the profit from serving transport orders while minimising the costs. The problem is solved with a heuristic method based on the Frank-Wolfe algorithm. Similarly, Crainic, Gendreau and Dejax [1993] include random variables for supply and demand into their empty container allocation model. To keep the model computationally tractable, they develop a two-stage approach which is only partially stochastic and can be solved recursively. For the haulage industry the stochastic dynamic vehicle allocation problem has been formulated (see W. B. Powell [1986]; Frantzeskakis and W. B. Powell [1990]). The main difference to the models used for freight wagons and containers is, that both empty and loaded flows are considered, so that the network represents a closed system with a constant number of vehicles. As these models can often only be solved heuristically, efficient solution algorithms have been in the focus of research (see for instance Frantzeskakis and W. B. Powell [1990]; Cheung and W. B. Powell [1996]). Topaloglu and W. B. Powell [2006] extend the model to multiple substitutive vehicle types. Cheung and Chen [1998] consider maritime containers but stick to a similar model, extended by the leasing of external containers as an additional option to minimise costs. Specialised heuristics are used as solution methods. In this category, another model for repositioning planning of freight containers under uncertain demands needs to be mentioned. It has been developed by S.-W. Lam, L.-H. Lee and Tang [2007], who also use a dynamic programming approach and provide a heuristic solution method. Godfrey and W. B. Powell [2002a] propose a stochastic dynamic programming approach to assign a fleet of vehicles to randomly arising tasks. They show that for these stochastic problems their approach can outperform deterministic programs applied in a rolling-horizon manner. This is concluded from the ex-post comparison of objective values after the random variables had been realised. The authors also consider the repositioning between several locations but only using deterministic travel times of exactly one period. This limitation is addressed by Godfrey and
1.3 Literature Review

W. B. Powell (2002b) who extend the former model to multiperiod travel times.

Another option to cope with stochastic data is robust optimisation as proposed by Erera, Morales and Savelsbergh (2009). The authors develop a method to calculate robust solutions to the dynamic empty repositioning problem for generic resources such as wagons, containers or lorries. Their aim is to obtain a cost-efficient flow with uncertain supply and demand data. The solution incorporates a set of recovery actions that can be implemented in order to re-establish a feasible solution. Instead of regarding the uncertainty in the model itself, Francesco, Crainic and Zuddas (2009) apply a deterministic repositioning model to multiple scenarios. So, instead of calculating an optimal decision based on point forecasts for future demand and supply, several scenarios have to be defined. The different scenarios are then included into a combined optimisation model that aims at producing the flow with the best sum of weighted objective values. This idea is further developed by Francesco, Lai and Zuddas (2013) to provide a planning which is robust in the case of port disruptions when handling and transportation of containers are possible only to a limited extent.

1.3.3 Capacity Planning for Reusable Containers

In the previous section, transport planning approaches were surveyed. The following section reviews the available research regarding the capacity planning tasks in container systems. Table 1.4 on the following page provides an overview of the discussed literature.

Static and Stochastic Capacity Planning

Focusing only on strategic fleet sizing decisions, several authors provide methods of obtaining an optimal long-term fleet size. Imai and Rivera (2001) determine the number of maritime containers used exclusively in the shuttle service between a source and a destination port. In the case of ‘unbalanced trade’ and multiple harbours, a simulation based approach is used. Lange and Semal (2010) instead analyse the problem of finding the necessary number of reusable containers as a
1 Introduction

<table>
<thead>
<tr>
<th>Publication</th>
<th>Time</th>
<th>Data</th>
<th>Planning object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avi-Itzhak, Benn and B. A. Powell 1967</td>
<td>Static</td>
<td>Stochastic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Baldacci, Battarra and Vigo 2008</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Beaujon and Turnquist 1991</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Bojovic 2002</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Bräysy et al. 2008</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Dantzig and Fulkerson 1954</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vessels</td>
</tr>
<tr>
<td>Dong and Song 2009</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Du and Hall 1997</td>
<td>Static</td>
<td>Stochastic</td>
<td>Lorries</td>
</tr>
<tr>
<td>George and Xia 2011</td>
<td>Static</td>
<td>Stochastic</td>
<td>Rental vehicles</td>
</tr>
<tr>
<td>Imai and Rivera 2001</td>
<td>Static</td>
<td>Stochastic</td>
<td>Standard containers</td>
</tr>
<tr>
<td>Köchel, Kunze and Nieländer 2003</td>
<td>Static</td>
<td>Stochastic</td>
<td>Rental vehicles</td>
</tr>
<tr>
<td>Koenigsberg and R. C. Lam 1976</td>
<td>Static</td>
<td>Stochastic</td>
<td>Vessels</td>
</tr>
<tr>
<td>Lange and Semal 2010</td>
<td>Static</td>
<td>Stochastic</td>
<td>Generic containers</td>
</tr>
<tr>
<td>List et al. 2003</td>
<td>Static</td>
<td>Stochastic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Papier and Thonemann 2008</td>
<td>Static</td>
<td>Stochastic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Parikh 1977</td>
<td>Static</td>
<td>Stochastic</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Savin et al. 2005</td>
<td>Static</td>
<td>Stochastic</td>
<td>Rental vehicles</td>
</tr>
<tr>
<td>Sayarshad, Javadian et al. 2010</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Sayarshad and Ghoseiri 2009</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Sayarshad and Marler 2010</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Sayarshad and Tavakkoli-Moghaddam 2010</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Railway waggons</td>
</tr>
<tr>
<td>Sherali and Tunchilek 1997</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vessels</td>
</tr>
<tr>
<td>Turnquist and Jordan 1986</td>
<td>Dynamic</td>
<td>Stochastic</td>
<td>Special containers</td>
</tr>
<tr>
<td>Vemuganti, Oblak and Aggarwal 1989</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Vessels</td>
</tr>
<tr>
<td>Wu, Hartman and Wilson 2005</td>
<td>Dynamic</td>
<td>Deterministic</td>
<td>Lorries</td>
</tr>
</tbody>
</table>

Table 1.4: Overview of the literature on capacity planning.
stock optimisation problem. They propose a method based on the classic economic order quantity model, to determine the necessary amount of containers, so that they can be transported back in optimal lot sizes. A robust optimisation model is provided by List et al. (2003). It consists of a two-stage decision model, where in the first stage the fleet capacity is determined using incomplete information. In the second stage it has to be decided if demand is covered, if vehicles are moved between locations or if backorders are created. In this step, the data is considered to be deterministic. The solution method is based on simulating the financial results of the fleet sizing decision in several scenarios.

A fleet sizing model based on inventory theory is presented by Du and Hall (1997). They show how, based on an (s, Q) policy, a minimum fleet size can be determined to reach a certain service level for transportation tasks. The model is used to optimise a fleet of lorries in a hub-and-spoke network with lorry repositioning.

Another approach is to model a rental fleet with stochastic processes or queueing models. Especially in the rental business, fleet sizing models based on queueing theory are used. One early approach for a pool of freight waggons is described by Avi-Itzhak, Benn and B. A. Powell (1967). They model incoming orders as a stochastic process with independent inter-arrival times. These orders are then served by waggons which are used for a stochastic number of periods before they are returned to the pool. The probability distribution of the number of busy waggons can then be determined and used as a basis to choose the optimum quantity of waggons. In fact, this approach uses an infinite server queueing model. Koenigsberg and R. C. Lam (1976) explicitly employ the queueing theory to determine the optimum fleet size for a fleet of liquid natural gas vessels circulating between up to three ports. It is a simple queueing model that provides closed-form solutions via mean-value analysis. Parikh (1977) uses multiserver queues to model fleets of vehicles at different locations that serve randomly arriving transport requests. The number of vehicles (servers) can then be chosen so that the probability for customers having to wait is reduced to below a defined limit. This basic pattern is varied by other authors. Köchel, Kunze and Nieländer (2003) optimise a different objective: the total return of a fleet of rental vehicles in the steady state of their queueing model. The model itself is extended by the option of returning the rental
vehicles at different locations and by the decision to relocate them. This is no longer analytically solvable, so a combination of simulations and genetic algorithms is used. Papier and Thonemann (2008) also use a multiserver queueing model to optimise a fleet of freight wagons which consists of different types. Instead of backlogging customer requests, they use a queueing system with losses to represent lost sales. Savin et al. (2005) show how a similar queueing system can also be integrated into a control theoretic revenue management problem. Instead of a steady state analysis, they use a dynamic approach to optimise not only the fleet sizing itself, but also the allocation of the resources to two classes of customers. George and Xia (2011) extend the focus and use a closed queueing network to model a rental company with several stations. The vehicles circulate between the stations and the company’s customers are represented as the ‘servers’ of the queueing theoretic model. With this model, the vehicle availability at the different stations can be determined depending on the fleet size. Common to all these queueing models is that they focus on long-term, strategic decisions and can only roughly represent the details of the underlying system. They are able however, to provide means for an approximate analysis of the long-term behaviour of logistic systems with different fleet sizes.

Dynamic and Deterministic Capacity Planning

Another branch of literature proposes solving fleet sizing problems with deterministic linear programming models. This approach often includes the planning of movements; so effectively they are combined transportation and capacity planning models.

One of the first models of this type was published by Dantzig and Fulkerson (1954). It is a continuous linear program that determines the number of tankers needed to meet a fixed schedule of transports between several harbours.

Vemuganti, Oblak and Aggarwal (1989) develop network models based on the fleet management transshipment models described in Section 1.3.2. They extend this class of models to incorporate vehicle replacement. For this purpose, network flows
1.3 Literature Review

are modelled for vehicles of different ages. The model can also cope with several types of vehicles, and still remains efficient to solve because it is a continuous linear program which shows the integrality property (see Section 2.4). Sherali and Tuncbilek (1997) extend a multiperiodic vehicle routing problem for freight wagons by a fleet sizing decision. This class of models is computationally complex, so a decomposition heuristic is developed. Nevertheless an even more complex model class is available, known as the ‘fleet size and mix vehicle routing problem’. Here, the individual vehicles are differentiated. So it has to be decided which vehicles should be acquired and simultaneously, how the individual vehicles should be routed. Baldacci, Battarra and Vigo (2008) provide an overview of the research in that area. For example Wu, Hartman and Wilson (2005) show how such a model can be solved using a decomposition approach together with lagrangian relaxation. Bräysy et al. (2008) instead use a deterministic annealing metaheuristic for their similar model.

Sayarshad and Ghoseiri (2009) formulate a model for the fleet sizing and allocation of homogeneous railway wagons. It provides the option of delaying transports as backorders which leads to penalty costs that are subtracted from the profit-maximising objective function. As this model can only solved optimally for small problems, the authors develop a simulated annealing approach to generate heuristic solutions. One limitation of this model is that the network nodes are divided into origin and destination nodes. Empty wagons are sent back from the destination to the origin nodes before they can be used again.

Based on this approach Sayarshad and Marler (2010) and Sayarshad, Javadian et al. (2010) show how to optimise the fleet size with regard to multiple objectives. They use the concept of pareto-optimality as a tool to identify a set of reasonable solutions.

Dynamic and Stochastic Capacity Planning

In the light of stochastic data, another field of activity focuses on extending the dynamic models to allow for uncertainty of travel times, supply and demand data.
For example Turnquist and Jordan (1986) examine containers used to transport parts from one location to a group of destinations. They show how the needed amount of containers can be calculated based on the schedule of the production process that fills the containers. First assuming deterministic travel times they afterwards extend their model by stochastic transportation times. In this case the optimum fleet size can be determined that is needed to keep the shortage probability below a certain level. Beaujon and Turnquist (1991) extend these models to analyse a network with arbitrary flows and backorder costs. The container fleet is allocated to several locations in order to serve transportation demands. The objective function corresponds to the expected revenue and also factors in the variance of the stochastic travel times. The result is a minimum cost flow problem with a non-linear but concave objective function that can be solved iteratively using a gradient search method. Bojovic (2002) shows how to determine the optimal fleet capacity and allocation when demand as well as movement times are uncertain. Based on control theory, an optimal policy can be found to balance the costs of providing additional freight wagons and the costs of accumulating backorders. Sayarshad and Tavakkoli-Moghaddam (2010) instead choose simulated annealing heuristics to tackle a similar problem with stochastic demand and travel times. Dong and Song (2009) focus on maritime containers circulating between ports, where the stochastic demand and disposal are imbalanced. Using a multiperiodic network model they show how an optimal long-term relocation policy and fleet size can be determined. They combine a mixed integer program with a demand simulation and optimise the parameters with genetic algorithms.

1.4 Scope and Outline

In the previous sections the key terms and planning problems regarding container systems were explained and the existing literature was surveyed. This information is used to define the scope of the thesis in this section and to evaluate which existing research results are most useful in this context. Afterwards, the selected approach is introduced and it is outlined how the remainder of the text is organised.
1.4 Scope and Outline

Scope

The aim of this thesis is to develop methods for planning transports and capacity adjustments in a container system for generic multi-use containers as defined in Section 1.2. These methods should help to define the best trade-off between more acquisitions and faster circulation of the containers. The scope ranges from very short-term transportation planning, on an hourly basis, to more medium-term capacity planning for multiple weeks. A special aspect needs to be taken into consideration here: joint movements of containers on the same vehicles. Containers are often not carried individually but are small enough to share one transport vehicle with many other containers. This characteristic needs to be regarded in order to avoid expensive transports with low vehicle utilisation. The resulting fixed-step transport costs (see Section 4.1) are to be considered.

Appraisal of the Available Research

The aim formulated above leads to requirements for an appropriate solution approach:

- An optimal transportation plan should be determined, which also integrates the best choice for lost orders or delayed deliveries if necessary.
- Capacity measures such as acquisitions or leasing of containers should be integrated.
- The fixed-step costs induced by the vehicle capacities should be considered.
- The approach should be usable for any kind of reusable container.

These requirements can be compared to the available literature in order to identify the most suitable methods.

It was shown in the literature section (Section 1.3) that although only very little research has been dedicated to general containers, particular types of containers such as intermodal standard containers have been studied. There is no need to distinguish between different types in terms of mathematical planning methods; the results can be generalised.
1 Introduction

Aspects from both transport and capacity planning should be integrated, so static long-term fleet sizing models, as well as short-term models which are not capable of incorporating capacity planning aspects, are not suitable.

In the light of uncertain data, it seems to be valuable to solve such a planning task with a stochastic approach. However, the available stochastic models, both from the transport planning, as well as from the capacity planning area, can hardly be extended to the other area without becoming computationally intractable. It is currently not effective to integrate all necessary capacity planning aspects into any of the available transport planning models, or reversely. In addition, stochastic models need more input data to operate on. In situations where it may already be difficult to gather sufficient data about the containers needed in the next weeks, information about the probability distribution of these demands and of other model data would also have to be collected. On the other hand Godfrey and W. B. Powell (2002a) show that deterministic models applied on stochastic data (in a rolling-horizon approach) are outperformed by a stochastic approach. Interesting in this context is the idea of developing methods that work well with insufficient information as suggested by Crainic, Gendreau and Dejax (1993). Nevertheless, for practical purposes, it is recommended here to use deterministic models and to rely on a rolling-horizon approach that would generate stable solutions for the next few planning periods where the data is then fully available.

Two further groups of literature remain to be considered for the planning task: vehicle routing models and fleet management models with dynamic network model structure. The first group is normally used for vehicles. It is therefore important to distinguish between the individual items and necessary to plan round trips, because drivers need to get back home or the vehicles have to be returned to their depot location. Although this does not seem to be necessary for containers, with a few changes these models could still be applied to containers instead of vehicles. However, the aspect of fixed-step transport costs would need to be integrated, which would increase the model complexity. Already in their original form, the vehicle routing models are comparatively complex integer optimisation problems. It is therefore not effective to apply this type of model to a fleet of containers, which may often be much larger than a fleet of vehicles.
1.4 Scope and Outline

The last type of literature concerns deterministic, multiperiod network flow models, which are in the simplest form equivalent to the transshipment model. These have been developed especially for maritime containers and railway wagons and are very flexible. Several aspects have been integrated, such as leasing (e.g. by Shen and Khoong [1995]), delivery within time windows (e.g. by Erera, Morales and Savelsbergh [2005]) and lost sales (e.g. by Kikuchi [1985]). However, the important aspect of fixed-step transport costs has apparently not been examined yet.

Selected Approach

The approach chosen here follows the tradition of classic empty repositioning models, based on dynamic networks such as described by White and Bomberault [1969] and White [1972]. This basic method is applied to the transportation and capacity planning of reusable containers. Several aspects are integrated into one flexible toolbox. With regards to transports, delayed deliveries and lost orders can be planned, as it is already known from existing models. On top of that, a model extension is developed in this thesis to exploit the opportunity to move containers free of charge, whenever transports of other items lead to remaining capacities on transport vehicles. Considering short-term capacity planning, container leasing as well as purchases and disposals of containers are available as optional model aspects. Comparable models often only provide leasing as a capacity measure, partially because the possibility to lease an unlimited number of containers guarantees a transport planning model to have a feasible solution in every situation. However, the most important distinguishing aspect to the existing research is that container transport costs are assumed to be non-linear. They increase in steps. This approach is unique in this area of research. Existing publications on dynamic network flow problems either chose linear transportation costs or sometimes also fixed charges (‘fixed-charge network flow problem’, see Section 4.1.3). All these extensions and the linear and non-linear cost structure can be combined. Thus, they can be considered as a toolbox rather than multiple independent models, because they can be tailored to the needs of a particular container system.

The models presented here use a dynamic time horizon but are completely deterministic in terms of the usage and transport durations. The transportation planning
1 Introduction

is modelled explicitly. That means that, for each transport, a route as well as the
time of the departure have to be determined. All models focus only on one type of
container.

For all models it is discussed how they can be solved efficiently. It is shown that
the basic models can simply be solved as linear programs. Specialised methods are
only necessary if the non-linear objective function is used. Based on the experience
of multiple solution approaches which have been tested, it is explained here how
a branch-and-cut algorithm can be set up best exploiting some special model
characteristics.

Outline

The remainder of this thesis is organised in the following way. The general structure
of the models developed here is explained in Chapter 2 which outlines a basic
model. This is extended afterwards. Chapter 3 shows how several important
aspects for container systems can be integrated. Namely the transport planning
extensions in Section 3.1 such as lost orders, delayed deliveries and the option to
exploit take-along opportunities for moving containers. Section 3.2 shows how
the more long-term capacity aspects of purchases, disposals and leasing can be
included. All of the model extensions up to this point do not change the model
complexity to a large degree. This only changes in Chapter 4 which examines the
effects of the bundled transportation of containers on vehicles. Here an extension is
introduced that integrates the non-linear transport costs that arise in this situation.

From the operations research perspective, the resulting model basically represents
a transshipment model with fixed-step transport costs. It is shown how this model
can be transferred to a linear mixed integer program, which then is solved by a
tailored branch-and-cut approach.

Table 1.5 summarises the models and extensions developed in this thesis. All models
and aspects can be combined into one model, depending on the purpose and on
the nature of the container system. All models are integer programs, but most of
them can be mathematically reduced to the transshipment problem, and therefore
efficiently be solved as continuous linear programs. These are marked accordingly
1.4 Scope and Outline

<table>
<thead>
<tr>
<th>Model element</th>
<th>Section</th>
<th>Integrality property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic model</td>
<td>2</td>
<td>✓</td>
</tr>
<tr>
<td>Lost orders</td>
<td>3.1.1</td>
<td>✓</td>
</tr>
<tr>
<td>Delayed deliveries</td>
<td>3.1.2</td>
<td>✓</td>
</tr>
<tr>
<td>Take-along transports</td>
<td>3.1.3</td>
<td>✓</td>
</tr>
<tr>
<td>Purchases and disposals</td>
<td>3.2.1</td>
<td>✓</td>
</tr>
<tr>
<td>Container rental</td>
<td>3.2.2</td>
<td>✓</td>
</tr>
<tr>
<td>Fixed-step transport costs</td>
<td>4</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1.5: Overview of all model variations in this thesis.

... in the column ‘Integrality property’ in Table 1.5. This is not the case if fixed-step transport costs are to be regarded. An integer optimisation algorithm, such as the branch-and-bound developed in Section 4.3.4 needs to be used then in order to obtain integer results.
The objective is to find the optimal movement of empty containers through a network of nodes in which transporting and storing of containers are permitted. Both lead to costs which in their sum need to be minimised. This chapter defines a simple model to solve this task. It serves as a basis for the more elaborate models presented later.

The storage costs are assumed to be proportional to the number of containers, and are a factor that can vary for each network node. The transportation costs are linear to the moved quantity, with a factor that depends on the connecting edge between two nodes. As explained above, only costs for the movement of empties are taken into consideration. The transportation of loaded containers is planned independently and therefore the arising costs are not relevant here.

As the model is designed to optimise only empty container movements, it is first shown in Section 2.1 how these are related to the movements of loaded containers and how orders for empties can be derived from existing plans for the transport goods. Section 2.2 then explains the model assumptions, while the model itself is issue in Section 2.3 Chapter 2 is concluded with Section 2.4 where the complexity of the model, the integrality property as well as solution methods are discussed.
2.1 Loaded versus Empty Container Movements

It is important to distinguish between the movements of empties and those of loaded containers in a logistic network. The full container movements can be derived from transport or production plans for the carried goods. In the subsequent planning of container movements, these have to be regarded as fixed data with each movement resulting in a certain demand of empty containers at a particular location and point in time. Furthermore, these plans provide information about the locations where, and the time when the used containers will be available again, because they are then no longer in use. So, only the movement of empties is an open decision in a container transport plan. The full movements can be considered as fixed.

The amount of empty containers needed for several uses is regarded as a set of orders \( o \in O \). To each order belongs a quantity \( x^O_o \in N \) of containers, a network node \( D_o \in N \) were they are needed, a node \( S_o \in N \) where they are discharged and the periods indicating the time of demand \( t^D_o \in \mathcal{T} \) and discharge \( t^S_o \in \mathcal{T} \). While it might be the case, that the locations specified by one order are identical, we know that for the periods the relation \( t^S_o > t^D_o \) must be satisfied. This means, that the point in time when a container is released is always after the moment it was taken into use.

The total periodic demand \( d_{i,t} \) at a certain location and time is the sum of the quantities needed to cover all orders. The supply \( s_{i,t} \) correspondingly results from the sum of released containers:

\[
d_{i,t} = \sum_{o \in O} x^O_o \quad \text{and} \quad s_{i,t} = \sum_{o \in O} x^O_o \quad \forall t \in \mathcal{T}, i \in N
\]

The shortest usage of a container that can be expressed this way is one whole period. Even if a container were only needed for a shorter amount of time, and could theoretically be made available again, it would have to be booked for the whole period in the context of this model. This is an important data requirement. If it were violated, the demand and supply would compensate each other, and during
the optimisation no containers would be reserved. It can be seen as a limitation of this model, that containers which are available for parts of a period cannot be utilised during the whole time frame. Nevertheless, this issue can be avoided by shortening the period length. This allows multiple short usages to be expressed.

2.2 Assumptions

Before a model is developed, the underlying assumptions need to be explained and the notation has to be introduced. Table 2.1 lists the main assumptions of the model expressed in indices, data and decision variables. These aspects are explicated in the following paragraphs.

### Indices:
- Periods
- Locations / Nodes
  - \( t \in \mathcal{T} = \{0, 1, 2, ..., T\} \)
  - \( i \in \mathcal{N} = \{1, 2, ..., N\} \)

### Data:
- demand quantities
  - \( d_{i,t} \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \)
- supply quantities
  - \( s_{i,t} \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \)
- transport times
  - \( m_{i,j} \quad \forall i, j \in \mathcal{N} \)
- transport costs
  - \( c_{t,i,j} \quad \forall i, j \in \mathcal{N} \)
- storage costs
  - \( c_{stock} \quad \forall i \in \mathcal{N} \)
- maximum stock
  - \( y_{i,t}^{max} \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \)
- maximum transport quantity
  - \( x_{i,j,t}^{max} \quad \forall i, j \in \mathcal{N}, t \in \mathcal{T} \)

### Variables:
- transport quantities
  - \( x_{i,j,t} \quad \forall i, j \in \mathcal{N}, t \in \mathcal{T} \)
- unused containers at the end of a period
  - \( y_{i,t} \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \)

Table 2.1: The symbols used for the basic model.
Basic Setting

The model operates on discrete time, which means that the planning horizon is divided into periods $t \in \mathcal{T}$, which can be days or even time slots of only a few hours. The usage of containers is modelled as ‘supply’ and ‘demand’, as explained in Section 2.1. Consequently, the container network is represented as an open network. Over time containers enter or leave the system. So the total number of containers in the system, full and empty ones, cannot be deduced directly from the optimisation results. The containers are moved through a logistic network consisting of nodes $i \in \mathcal{N}$, each of these being either a supply node, a demand node or just a transshipment node. The assignment of the nodes to each of these types is not fixed but may change over time. Physically, these nodes can be any site where a company’s containers may be located. For example, a plant of a component supplier, where special empty containers are needed, could also be such a node.

Decision Variables

Decision variables in this basic setting are first the quantity of containers that are moved from node $i$ to $j$ starting in period $t$, which is written as $x_{i,j,t}$. And secondly, the variable $y_{i,t}$ represents the quantity of containers that does not move between nodes, but is stored at node $i$ from period $t$ to $t + 1$.

Data

The demand and supply quantities of empty containers $d_{i,t}$ and $s_{i,t}$, introduced in Section 2.1, are used as input data. Of course, transportation between the nodes affords some time, which is explicitly included in the model as a transport time $m_{i,j}$. It represents the number of periods that a transport from node $i$ to node $j$ takes. It is assumed to have an integer value greater than zero. The containers transported between two nodes $i$ and $j$ may also produce costs per unit of $c_{i,j}^{\text{transport}}$. Also local storage over time on individual nodes may lead to node-dependent costs of $c_{i}^{\text{stock}}$ per unit. This allows the differentiation between locations with low or high storage costs. In addition, the use case may require to define an initial and a final stock at the network nodes.
2.3 Model

It is important to note that all quantities and transportation times must be integers, to allow this model to be optimised efficiently (see Section 2.4 for details).

2.3 Model

The model introduced in this section is a special case of the transshipment model applied to empty container transportation. It is intended to be the foundation of a more sophisticated framework for short-term container transportation and acquisition planning. The planning problem described by this basic model is just finding the cost-optimal solution to move containers between the facilities of a company so that each location receives the desired number of empty containers. The model is listed in Figure 2.1 on the next page.

Objective

The objective function consists of two terms which depict the two cost factors that have to be regarded here. The first term describes the transport costs as the product of the transport quantity between each pair of nodes and the corresponding transport cost factor. The inventory costs are included with the second term. For each node and period they equal the product of the stored quantity and a node-dependent inventory cost factor. The objective is to minimise the sum of these costs subject to the constraints which are explained in the next paragraphs.

Conservation of Flow Constraint

The movement of containers between locations over time is kept track of in constraints 2.2 and 2.3, the ‘conservation of flow’ conditions. Due to practical reasons, for the first period a separate constraint is necessary. It assures that the amount of containers stored at each node at the end of the first period corresponds to the number of containers arriving in that period minus the number of containers departing, either to be used \( (d_{i,0}) \) or to be moved to another node \( (x_{i,j,0}) \). Similarly, the conservation of flow for the remaining periods is managed in constraint 2.3. Two additional variables have to be taken into consideration here: first, the
Model: Minimising transport and inventory costs

\[
\min_{x_{i,j,t}, y_{i,t}} \sum_{r \in \mathcal{T}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} x_{i,j,t} \cdot c_{\text{transport}}^{i,j} + \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} y_{i,t} \cdot c_{\text{stock}}^{i} \\
\text{subject to the conservation of flow,}
\]

\[
y_{i,0} = s_{i,0} - \sum_{j \in \mathcal{N}, j \neq i} x_{i,j,0} \quad \forall i \tag{2.2}
\]

\[
y_{i,t} = y_{i,t-1} + s_{i,t} - d_{i,t} + \sum_{j \in \mathcal{N}, j \neq i} x_{j,i,t} \cdot m_{j,i} - \sum_{j \in \mathcal{N}, j \neq i} x_{i,j,t} \quad \forall i, t > 0 \tag{2.3}
\]

demand fulfilment and inventory constraints

\[
0 \leq y_{i,t} \leq y_{i,t}^{\text{max}} \quad \forall i, t \tag{2.4}
\]

and upper and lower bounds of transport quantities.

\[
0 \leq x_{i,j,t} \leq x_{i,j,t}^{\text{max}} \quad \forall i, j \neq i, t \tag{2.5}
\]

Figure 2.1: A model for minimising transport and inventory costs.
remaining stock $y_{i,t-1}$ from the preceding period, and secondly, arriving transports $x_{j,i,t-m_{j,i}}$ from other nodes. These represent containers that departed at node $j$, took $m_{j,i}$ periods to travel and arrive at node $i$ in period $t$. Obviously, the first transport from node $i$ to $j$ would arrive in period $m_{i,j}$ if it started in period 0. Transports coming in earlier than in period $m_{i,j}$ do not need to be considered.

**Fulfilment and Inventory Constraint**

Constraint [2.4] assures that the demands represented by $d_{i,t}$ are fulfilled. This is the case if all $y_{i,t}$ are non-negative as ensured by this inequality. On the other hand, this constraint guarantees that the predefined inventory limits $y_{i,t}^{max}$ are not exceeded.

**Transport Quantity Constraint**

The last constraint [2.5] defines the upper and lower limits for the number of containers transported between the nodes. They have to be non-negative but upper bounds can also exist.

**Initial and Final Stock**

In general, there might be some empty containers in the network initially. These items are to be regarded as initial stock. The simplest way is to add these containers to the supply $s_{i,0}$ in the first period. Final stock can be defined in the same way. If there is a desired distribution of empty containers at the end of the last period, this amount can be added to the last demand quantities $d_{i,T}$. The appropriate figure for the final stock of containers at time $T$ is defined as part of the higher-level capacity planning as explained in Section [1.2.2].

---

**2.4 Complexity and Solution Procedures**

In this section, some mathematical characteristics of the model are discussed. In particular, it is shown that it can be expressed as a ‘minimum cost flow problem’. Hence, the findings about the complexity and solution procedures can be transferred.
Similarity to the Minimum Cost Flow Problem

The model presented in the last section is mathematically identical to the minimum cost flow problem, which is listed in Figure 2.2 (for a formulation of the minimum cost flow problem see Ahuja, Magnanti and Orlin 1993, pp. 4 ff.).

**Theorem.** The basic model (Figure 2.1) can be expressed as a minimum cost flow problem.

**Proof.** To convert the basic model into the minimum cost flow problem (Figure 2.2), only the notation needs to be changed. Each node $i$ can be seen as a different node for every point in time. So nodes $(i, t)$ and $(i, t + 1)$ can be represented by different nodes in the minimum cost flow model. The amounts of stock transferred from one period to another are then modelled as regular flows on an arc connecting these different nodes. In addition, the conservation of flow can be reformulated, so that the excess supply $(s_{i, t} - d_{i, t})$ is combined as $b(i)$. The upper and lower bounds of the variables can be used without modification with only a change of the variable names. 

\[
\begin{align*}
\text{Model: Minimum cost flow problem} \\
\min & \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}, j \neq i} x_{i, j} c_{i, j} \\
\text{subject to} & \sum_{j \in \mathcal{N}, j \neq i} x_{j, i} - \sum_{j \in \mathcal{N}, j \neq i} x_{i, j} = b(i) & \forall i \in \mathcal{N} \\
& l_{i, j} \leq x_{i, j} \leq u_{i, j} & \forall i, j \neq i \in \mathcal{N}
\end{align*}
\]

Figure 2.2: The minimum cost flow problem.
2.4 Complexity and Solution Procedures

This relationship can be illustrated by the concept of ‘time-expanded networks’, as rendered by Ford and Fulkerson (1962). This is a network (or graph) with nodes that are observed over a set of periods. Each node is copied for every period. Thus, such a ‘dynamic’ graph consists of one vertex $n_{i,t} \in \mathcal{N}_T$ for every node $n_i \in \mathcal{N}$ in the ‘static’ graph. The arcs and the flows routed over these arcs represent flows over time. Two types of flow occur in such a network. The first one is flow from a Node $n_{i,t}$ to the node $n_{i,t+1}$, which corresponds to keeping it as stock from one period to the next. The second type is a flow between different locations $i$ and $j$. In this context, such flow is assumed to consume some transportation time $m_{i,j}$, so that a flow between nodes in the same period is not possible. All arcs are directed from one node in period $t$ to another node in a future period $t + m_{i,j}$. Figure 2.3 shows an example of a static network with three nodes and the corresponding time-expanded network with four periods. The resulting graphs show an acyclic structure because all arcs from a node are directed to nodes in later periods. In the literature these graphs are called ‘fully dynamic networks’ (W. B. Powell, Jaillet and Odoni 1995, p. 222). Figure 2.4 on the following page shows the basic model with its transports and stocks as a time-expanded network.

Integrality Property

The basic model (Figure 2.1) has another important feature that should be mentioned. It is an integer linear programming problem because the decision variables, $x_{i,j,t}$ and $y_{i,t}$, must be integers. They represent amounts of containers which are
counted in integral numbers. But due to the special structure of the problem, its optimal solution is equal to that of its continuous linear programming relaxation. This property is inherited from the minimum cost flow problem (for the integrality property of this problem see Suhl and Mellouli 2009, p. 84).

That means the model can be solved efficiently without employing an integer optimisation algorithm although the variable values are required to be integral. The reason for this is, that the coefficient matrix of the linear programming formulation is totally unimodular as long as all supplies, demands and capacities are also integer. In effect all feasible solutions of the continuous linear problem are bound by an integral polyhedron (Schrijver 2000, p. 266). A proof of this integrality property can be found in Ahuja, Magnanti and Orlin (1993, Chapter 11).

**Complexity**

So, the basic problem defined on a network with $N$ nodes and $T$ periods corresponds to a minimum cost flow problem with $N \cdot T$ nodes. The number of arcs in the time expanded network depends on the connections that exist in the container network and the number of nodes that can store containers. At most one storage arc can exist for every node in each period and one transportation arc between each pair of nodes per period. That would result in $N^2 \cdot T$ transportation arcs and $N \cdot T$ storage
arcs, provided that sending transports and storing are both allowed in all periods. For the minimum cost flow problem, algorithms are known which solve it in polynomial time in the number of nodes $n$ and arcs $m$. For an overview of the complexity of several of these algorithms see Ahuja, Magnanti and Orlin (1993, p. 395). For the time-expanded network representing the basic model, this parameters would maximally amount to $n = N \cdot T$ and $m = N^2 \cdot T + N \cdot T$. So this type of algorithm would also need polynomial time in the parameters $N$ and $T$. 

2.4 Complexity and Solution Procedures
3 Extensions to the Basic Model

The basic model introduced in the previous chapter can be used as the starting point for a more sophisticated modelling of a container system. In the following chapter two groups of extensions are developed. The first one focuses on the transportation process and adds the decisions about the most cost-efficient lost orders and delays as well as the option to use available take-along opportunities for cheaper transportation. The second group of extensions aims at incorporating capacity planning aspects, in particular the purchasing, disposing and leasing of containers.

3.1 Transport Planning Extensions

This section discusses the first set of extensions, which enhances the model to render the transportation process in more detail. Table 3.1 on the next page lists the additional notation used for this purpose.

3.1.1 Allowing Lost Orders

Some companies are able to cancel orders if it is too difficult to provide the needed containers. To allow for such lost orders, a new decision variable $l_{i,t}$ is introduced. This represents the number of containers that are ordered but not supplied. Lost orders must be penalised to provide an incentive to avoid them. So, the objective function is equipped with an additional term representing the fictional penalty.
3 Extensions to the Basic Model

Data:
- Penalty costs for lost orders: $c_{i,t}^{\text{lost}}$ for all $i \in N, t \in T$
- Maximum amount of lost orders: $l_{i,t}^{\text{max}}$ for all $i \in N, t \in T$
- Penalty costs for backorders: $c_{i,t}^{\text{back}}$ for all $i \in N, t \in T$
- Maximum amount of backorders: $b_{i,t}^{\text{max}}$ for all $i \in N, t \in T$

Variables:
- Lost orders at node $i$ in period $t$: $l_{i,t}$ for all $i \in N, t \in T$
- Backorders at node $i$ in period $t$: $b_{i,t}$ for all $i \in N, t \in T$
- Transport vehicles used on a connection: $f_{i,j,t}$ for all $i, j \in N, t \in T$
- Capacity per transport vehicle: $k$

Table 3.1: Additional symbols used for the transport planning extensions.

In this term a new cost factor $c_{i,t}^{\text{lost}}$ is used:

$$+ \sum_t \sum_i l_{i,t} \cdot c_{i,t}^{\text{lost}} \quad (3.1)$$

Constraints 2.2 and 2.3 from the basic model must also be modified. Demanded but not delivered containers lead to a higher stock at the end of a period. So the new constraints are:

$$y_{i,0} = s_{i,0} + l_{i,0} - d_{i,0} - \sum_{j \in N, j \neq i} x_{i,j,0} \quad \forall i \quad (3.2)$$

$$y_{i,t} = y_{i,t-1} + s_{i,t} + l_{i,t} - d_{i,t} + \sum_{j \in N, j \neq i, m_{j,i} \leq t} x_{j,i,t-m_{j,i}} - \sum_{j \in N, j \neq i} x_{i,j,t} \quad \forall i, t > 0 \quad (3.3)$$

In a final step, the lost orders need to be limited. The variable must be non-negative
3.1 Transport Planning Extensions

and also an upper limit is necessary:

\[ 0 \leq l_{i,t} \leq l_{i,t}^{\text{max}} \quad \forall i, t \quad (3.4) \]

The upper limit is important, not just in its apparent meaning, but also because the lost orders cannot exceed the original demand quantity. So \( l_{i,t}^{\text{max}} \) has to be chosen in a way that \( l_{i,t} \leq d_{i,t} \). Otherwise it would be possible to select a very high value of \( l_{i,t} \), to 'generate' containers that can then be used in the system. If this source of containers is cheap because of a low cost factor for lost orders, the optimisation would make use of this option and generate a plan that is mathematically correct but practically infeasible. If the value of \( l_{i,t}^{\text{max}} \) is chosen carefully, this problem does not arise.

The additions to the basic model, from the perspective of its representation as a time-expanded network, are shown in Figure 3.1

![Graph representation of lost orders](image)

Figure 3.1: Graph representation of lost orders with an additional node providing fictional supply.

3.1.2 Allowing Delayed Deliveries

Instead of losing orders, one could consider to deliver the requested containers with a delay. In this case, orders or parts of orders are queued for one or more
3 Extensions to the Basic Model

periods. As this section will show, this is mathematically similar to lost orders. The
difference is, that orders are not only cancelled, but that additional changes to the
basic model are necessary in order to keep track of the queued orders.

First, some imputed, time-proportional penalty costs are introduced and added to
the objective function:

\[ + \sum_{i} \sum_{t} b_{i,t} \cdot c_{i,t}^{\text{back}} \] (3.5)

They consist of a node and period specific cost factor \( c_{i,t}^{\text{back}} \) and the amount of
backorders \( b_{i,t} \), i.e. the number of containers in delay at a specific node in a certain
period.

When the containers are delivered with a delay, the new decision variable \( b_{i,t} \) is
used to transfer demanded containers from one period to the next. So backorders
in one period lead to excess demand in the following period. But, if containers are
still too scarce, the amount of backorders can be transferred to later periods using
the variable \( b_{i,t+1} \) and so on. This leads to new constraints for the conservation of
flow:

\[ y_{i,0} = b_{i,0} + s_{i,0} - \sum_{j \in N, j \neq i} x_{i,j,0} \quad \forall i \] (3.6)

\[ y_{i,t} = b_{i,t} - b_{i,t-1} + y_{i,t-1} + s_{i,t} - d_{i,t} + \sum_{j \in N, j \neq i} x_{j,i,t-m_{j,i}} - \sum_{j \in N, j \neq i} x_{i,j,t} \forall i, t > 0 \] (3.7)

So, from the amount of containers available in a period, the outstanding orders are
satisfied first. If the rest is not sufficient, new backorders are generated.

As before, the amount of backorders can be limited for every specific period and
network node:

\[ 0 \leq b_{i,t} \leq b_{i,t}^{\text{max}} \quad \forall i, t \] (3.8)
Similar to the lost orders extensions, the backorders can also lead to, mathematically correct, but practically infeasible solutions. This happens if it is not prevented that the order queue variable $b_{i,t}$ is misunderstood as an additional supply of physical containers. Therefore the following equation is added:

$$b_{i,t} - b_{i,t-1} \leq d_{i,t} \quad \forall i, t > 0 \quad (3.9)$$

This inequality ensures that only additional demand from the current period can lead to an increase in the total amount of backorders.

This extension does not touch the integrality property of the model. As one can easily see, the backorders are mathematically identical to transports. Figure 3.2 shows the flows created by backorders for one of the network nodes in the time-expanded network. In principle the time-expanded network is extended by an additional edge between every two subsequent network nodes. So, the backorders $b_{i,t}$ correspond to the container storage $y_{i,t}$, but signify a movement into the opposite direction, namely from the future into the past.

![Diagram of network flows](image)

Figure 3.2: Some occurrences of backorders in the graph representation of the model.
3 Extensions to the Basic Model

3.1.3 Take-along Transports

When transporting empties, it is advantageous to use the remaining capacity of vehicles that move between several destinations for other reasons. If the available vehicle space is filled with containers, this additional load can be carried free of charge. It is mathematically simple and practically important to regard such opportunities while planning the container transports.

Such take-along transports can be expressed by duplicating the arcs between each pair of nodes and assigning flow variables $x_{TA}^{i,j,t}$ to these connections. These variables do not need to be added to the objective function as the corresponding transports are assumed to be free of charge. However, the movements must be incorporated in the conservation of flow conditions:

$$y_{i,0} = s_{i,0} - d_{i,0} - \sum_{j \in N, \ j \neq i} x_{i,j,0} - \sum_{j \in N, \ j \neq i} x_{TA}^{i,j,0} \quad \forall i \quad (3.10)$$

$$y_{i,t} = y_{i,t-1} + s_{i,t} - d_{i,0} + \sum_{j \in N, \ j \neq i, \ m_{i,j,i} \leq t} (x_{j,i,t-m_{i,j,i}} + x_{TA}^{j,i,t-m_{i,j,i}}) - \sum_{j \in N, \ j \neq i} (x_{i,j,t} + x_{TA}^{i,j,t}) \forall i, t > 0 \quad (3.11)$$

The transport variables are duplicated so that the take-along transports are symbolised by similar variables. Of course, such transport opportunities are only available for certain connections and periods. This is expressed by a new condition, limiting the maximum flow over each arc to the available capacity as shown below:

$$0 \leq x_{TA}^{i,j,t} \leq x_{i,j,t}^{TA,\text{max}} \quad \forall i, j \neq i, t \quad (3.12)$$

The network structure of the model with take-along transports is illustrated in Figure 3.3. As one can see, additional connections are added to the network wherever take-along opportunities are available.
3.2 Integrating Capacity Planning

The models presented in the last sections claim that all orders can be satisfied by containers that are already available in the network. To be more precise, only containers that are generated by positive values of \( s_{i,t} \) are available. So only exogenous supply of containers is taken into consideration. In a next step, the supply can be endogenised in the one way or another. This would enable integrating the acquisition, elimination or leasing of containers into the optimisation process.

The reason for doing this is the trade-off between acquisition and transportation of empty containers. If transportation and acquisition of containers are both considered in one optimisation model, the most cost-efficient compromise between a higher movement frequency and a larger fleet of containers can be determined.

The additional notation used for the following extensions are listed in Table 3.2.
3 Extensions to the Basic Model

Data:

- Acquisition costs of new containers \( c_{i,t}^{acq} \), \( \forall i \in N, t \in T \)
- Residual value of sold containers \( c_{i,t}^{res} \), \( \forall i \in N, t \in T \)
- Rent per container and period \( c_{i,t}^{rent} \)

Variables:

- Containers acquired at \( i \) in \( t \), \( z_{i,t} \), \( \forall i \in N, t \in T \)
- Containers sold at \( i \) in \( t \), \( v_{i,t} \), \( \forall i \in N, t \in T \)
- Containers newly rented at \( i \) in \( t \), \( r_{i,t}^{new} \), \( \forall i \in N, t \in T \)
- Rented containers returned at \( i \) in \( t \), \( r_{i,t}^{return} \), \( \forall i \in N, t \in T \)
- Sum of rented containers at the end of \( t \), \( r_{t}^{total} \), \( t \in T \)

Table 3.2: Additional symbols used for the capacity planning extensions.

3.2.1 Purchasing and Disposing of Containers

The option to purchase new containers or dispose of containers which are not used anymore is easy to integrate into the basic model of Chapter 2. A new group of decision variables \( z_{i,t} \) is introduced to represent the number of containers acquired in period \( t \) on network node \( i \). A second decision variable \( v_{i,t} \) stands for containers that are sold or scrapped. The corresponding costs are included in the objective function by adding the following term:

\[
+ \sum_{i} \sum_{t} z_{i,t} \cdot c_{i,t}^{acq} - \sum_{i} \sum_{t} v_{i,t} \cdot c_{i,t}^{res}
\]

\( c_{i,t}^{acq} \) is the price of a newly bought container. \( c_{i,t}^{res} \) in contrast is not necessarily a cost factor. It represents the residual value of a container sold, which would generate revenues. But it might also be assigned a negative value. In this case, it would correspond to costs that arise when containers are scrapped.

The containers entering or leaving the system need to be accounted for in the
3.2 Integrating Capacity Planning

conservation of flow conditions:

\[
y_{i,0} = z_{i,0} - v_{i,0} + s_{i,0} - d_{i,0} - \sum_{j \in N, \ j \neq i} x_{i,j,0} \quad \forall i \tag{3.14}
\]

\[
y_{i,t} = y_{i,t-1} + z_{i,t} - v_{i,t} + s_{i,t} - d_{i,t} + \sum_{j \in N, \ j \neq i} x_{j,i,t-m_{j,i}} - \sum_{j \in N, \ j \neq i} x_{i,j,t} \quad \forall i, t > 0 \tag{3.15}
\]

So, new containers are added to the amount at the end of a period, while containers sold are subtracted.

For the two groups of decision variables introduced here, a constraint has to ensure that their values are non-negative in every solution of the problem. Otherwise, it would be possible to ‘sell’ containers for the purchase price and they could be ‘bought’ for only their residual value by choosing negative values of \(z_{i,t}\) and \(v_{i,t}\). In some situations it might be necessary that buying and selling only occurs on some locations or only in certain periods. So upper bounds for the decision variables can also be defined:

\[
0 \leq z_{i,t} \leq z_{i,t}^{\text{max}} \quad \forall i, t \tag{3.16}
\]

\[
0 \leq v_{i,t} \leq v_{i,t}^{\text{max}} \quad \forall i, t \tag{3.17}
\]

An issue raised by this extension is, that the optimal solution might suffer from an end-of-horizon effect. If it were possible to sell containers for a positive residual value, an optimising software would do this as soon as a container is no longer needed. Therefore, in the last period all remaining containers would be sold and the system would be empty. This can only be prevented by adding fictional demand in the last period, representing the desired amount of containers at the end of the planning horizon. The amount to be defined here could be the number of containers available at the beginning. Another benchmark could be the maximum amount of empty containers which is in use at the same time over the planning horizon. It
3 Extensions to the Basic Model

![Figure 3.4: An example network with purchase and disposal of containers.](image)

can be calculated as follows:

\[
\arg \max_{s \in \mathcal{T}} \sum_{t=0}^{s} \sum_{i \in \mathcal{N}} (d_{i,t} - s_{i,t})
\]  

(3.18)

Another solution to avoid the end-of-horizon effect is to define a long planning horizon during the optimisation procedure and to use only the optimal solution of a shorter fraction of time. No changes to the model are then necessary to solve this issue.

How this extension integrates into a time-expanded network can be seen in the example network delineated in Figure 3.4. Two artificial nodes are added compared to the basic model. The first node serves as the source of additionally purchased containers, while the second is a sink for the containers removed from the system.

3.2.2 Container Leasing

The planner has much more flexibility if additional containers can be rented or leased. For the purpose of this thesis, the terms ‘leasing’ and ‘renting’ are used
3.2 Integrating Capacity Planning

Synonymously. In this section it is demonstrated how these aspects can be included into the model. The mathematical method is applicable to both, leasing and renting. Similar to the extensions described before, a new term in the objective function is needed to include the rental fees:

\[ + \sum_{t} \sum_{i} r_{i,t}^{\text{total}} \cdot c_{\text{rent}} \quad (3.19) \]

The parameter \( c_{\text{rent}} \) represents the fee to be paid per period and container. The new decision variable \( r_{i,t}^{\text{total}} \) is the total number of currently rented containers (rented and to be returned at node \( i \)) at the end of every period \( t \).

Two more groups of decision variables are needed: the number of containers \( r_{i,t}^{\text{new}} \) that are newly rented at a certain location and time, and the number of containers \( r_{i,t}^{\text{return}} \) that are returned. All three groups of variables depend on each other. This is expressed as an additional set of conservation of flow conditions:

\[
r_{i,t}^{\text{total}} = \begin{cases} 
    r_{i,0}^{\text{new}} - r_{i,0}^{\text{return}}, & \text{for } t = 0 \\
    r_{i,t}^{\text{total}} + r_{i,t}^{\text{new}} - r_{i,t}^{\text{return}}, & \text{for } 0 < t < T \\
    0, & \text{for } t = T
\end{cases} \quad \forall i, t \quad (3.20) 
\]

In terms of the network representation, a new network node ('container rental') is added as shown in Figure 3.5. This new node needs its own condition ensuring its conservation of flow or, to be more precise, to ensure that its net demand (of zero containers) is satisfied exactly. In order to record the amount of rented containers at any time, the variable \( r_{i,t}^{\text{total}} \) is introduced. Like the amount of stock \( y_{i,t} \), it is computed in every period from the number of containers rented or returned. Its value is equal to the number of containers that in the end have to be returned. This way, it is guaranteed that all rented containers are either returned or are paid for.

The conservation of flow of the residual network nodes also has to be changed.
3 Extensions to the Basic Model

Containers rented or returned are included:

\[
\gamma_{i,0} = r_{i,0}^{\text{new}} - r_{i,0}^{\text{return}} + s_{i,0} - d_{i,0} - \sum_{j \in N_i} x_{i,j,0} \quad \forall i \tag{3.21}
\]

\[
\gamma_{i,t} = \gamma_{i,t-1} + r_{i,t}^{\text{new}} - r_{i,t}^{\text{return}} + s_{i,t} - d_{i,t} + \sum_{j \in N_i, j \neq i} x_{j,i,t-m_{j,i}} - \sum_{j \in N_i} x_{i,j,t} \quad \forall i, t > 0 \tag{3.22}
\]

The quantities described by the new decision variables are non-negative. Furthermore, it makes sense to limit the number of containers that can be rented or returned in certain periods or at certain locations, or even to limit the total number of rented containers. This is summarised in the following constraints:

\[
0 \leq r_{i,t}^{\text{total}} \leq r_{i,t}^{\text{total, max}} \quad \forall i, t \tag{3.23}
\]

\[
0 \leq r_{i,t}^{\text{new}} \leq r_{i,t}^{\text{new, max}} \quad \forall i, t \tag{3.24}
\]
3.3 Numerical Results

With the models developed throughout the previous sections, many aspects of container systems can be regarded in the planning. The behaviour of these models are demonstrated based on some simple examples in this section. The results show the impact of the input parameters, such as the cost factors, on the outcome.

After explaining the basic setting and the common assumptions in the next section, four examples are discussed. The first one illustrates the trade-off between transportation and inventory which already exists for the basic model. Example 2 shows how additional transports can be used to compensate missing containers. On top of that, one of the extensions is added to demonstrate the effect of take-along transports. The aspects of purchasing and leasing are discussed afterwards, using the Examples 3 and 4.

3.3.1 Setting of the Examples

All examples are based on a similar setup. Figure 3.6 summarises the basic network used here. It consists of three locations. Two of them (Nodes 2 and 3) are close to each other with a transportation time of 1 period in between, while the third (Node 1) is further away, with a transportation time to and from the other nodes of 2 periods.

In the examples, containers are needed periodically at the different nodes and are always used for only one period. After that, they can either be stored at the nodes for the next usage or transported to other nodes for use there. If they are transported between the nodes, less containers are needed in total because the distances between the nodes are comparatively small. Figure 3.6 shows how the
Extensions to the Basic Model

Containers can be moved between the nodes in order to minimise the number of containers needed. Depending on the cost parameters for storing and transporting the containers or acquiring an additional amount, either the transports shown in the figure would be the optimal solution, or it would be more cost-efficient to provide additional containers to avoid some of the movements and save transport costs.

The transport costs are assumed to be equal to the distance between the several nodes multiplied by two. All other costs are initially set to zero:

\[ c_{i,j}^{\text{transport}} = 2 \cdot m_{i,j} \]
\[ c_{\text{acq}}, c_{\text{res}}, c_{\text{stock}} = 0 \]

The parameters are then varied to show their impact on the optimal solution. The planning horizon for the examples comprises 30 periods.

The following examples are based on the network and assumptions explained above. However, to show some effects more clearly, the basic setting is varied in the following examples.

3.3.2 Example 1: Transport versus Stock

The first example shows results for the simplest model, the basic model from Chapter 2. This model only describes the routing and storage of containers. The optimal
3.3 Numerical Results

solution shows the most cost-efficient routing. In particular, the decision is taken whether containers should be stored at one location or moved to another one.

The network and demand structure used here are taken from the first example network (see Figure 3.6). All three network nodes are initially equipped with a sufficient number of containers to satisfy the demand, i.e. with 10 containers for Nodes 1 and 3 and with 20 containers for Node 2. To illustrate the trade-off between transportation and inventory costs, two of the network nodes (Nodes 1 and 3) are assigned increasing inventory costs while for the other node (Node 2) the inventory costs are zero. Such a situation could arise when some sites are short of sufficient space to store containers, while others offer plenty of it. The differing inventory costs set an incentive to save storage capacity at certain locations.

Five replications are calculated. These differ only in the inventory costs for network Nodes 1 and 3. In the first replication, all inventory costs are zero. The other replications are based on higher inventory costs for the two nodes, ranging from one to four.

<table>
<thead>
<tr>
<th>Replication</th>
<th>Inventory costs</th>
<th>Optimal total stock</th>
<th>Optimal total transports</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( c_1^{Stock} )</td>
<td>( c_2^{Stock} )</td>
<td>( c_3^{Stock} )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3.3: Results for Example 1.

Table [3.3] illustrates the optimal solutions for the example model. Each row represents the solution for one replication. These replications differ in the assumed inventory costs. Depending on these, the amount of containers stored at the different locations and the total amount of containers moved between the nodes vary. These numbers are also listed in the table. The ‘optimal total stock’ is the sum of all containers stored at the nodes, summed up over all periods. Similarly, the ‘optimal total transports’ is the sum of all containers in transit, summed up over the complete planning horizon. This number depicts the total transport quantity measured as
3 Extensions to the Basic Model

the number of containers multiplied by the distance for each movement.

It can be seen that, in the initial setting, it is best not to transport any containers. This is possible because all nodes are equipped with a sufficient initial stock, so that the demand can be satisfied. No transports are planned because they are not needed to reach a feasible solution but would lead to costs for each container moved.

When the costs for inventory increase, the optimal solution changes. The stock is lowered at the more expensive locations and the containers are sent to other nodes. This leads to an increase in the total transport volume. At first, mainly the stock level at network Node 3 is lowered because this one is relatively near to Node 2 which can be used to store the containers without inventory costs. But also the stock level at Node 1 decreases slightly. This is due to the end of the planning horizon. The containers are moved from Node 1 to Node 2 for the remaining periods as soon as they are no longer needed. As they do not have to be moved back in this case, this is cost-saving while it would not be profitable to move them to the other node and back in the first periods. When stocks are penalised with an even higher cost factor, the stock level at Node 1 is also reduced as far as possible. Then it is cheaper to move the containers from Node 1 to Node 3 and back than to leave them at the first node.

In general, this example shows that it is best to move containers to another network node whenever the sum of transport and inventory costs at the new location are less than the costs for storing the containers at their original location.

3.3.3 Example 2: Availability of Containers and Take-along Transports

The next example shows the effect of the number of containers available on the number of transports. The model is then extended by ‘take-along transports’ as explained in Section 3.1.3. Additional replications show the effect of this new planning option on the optimal solution.
The example is based on a network with four nodes (see Figure 3.7). The first three are already known from the previous examples and a fourth node is added to show the effect of take-along transports. This node has no supply or demand of containers, but in the last of the three replications, free transport routes across this fourth node are added. They represent a remaining transport capacity on a vehicle which is moving for other purposes. This vehicle can be filled up with empty containers, so that they can be moved without additional costs.

![Figure 3.7: Example network 2 – locations, distances and container demands.](image)

<table>
<thead>
<tr>
<th>Replication</th>
<th>Initial stock</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Node 1</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 3.4: Initial stock at the network nodes in Example 2.

The transport costs in this example are always assigned the value two, inventory costs are zero. Three replications are optimised with different initial stock assumptions as listed in Table 3.4. In the first replication, there is sufficient initial stock of containers at each node to cover the local demand. Therefore, no transports are necessary. This is also proved to be optimal by the results. The initial stock is changed in the second replication. Nodes 1 and 3 are still assigned an initial stock of 10, but Node 2 starts without containers. Therefore containers have to be
moved here from other nodes to provide what is needed. This leads to an increase in transports and to a decrease in stock at the same time. For the last replication, the setting is extended by the option of using remaining capacity of vehicles moving between Nodes 2, 3 and 4. Truck symbols in Figure 3.7 symbolise these vehicle movements. Since these take-along transports and storage at Node 4 are free of charge, it is optimal to replace the more expensive dedicated container transports with movements across Node 4. This corresponds to the optimisation results listed in Table 3.5. The example shows that direct transports are replaced by take-along transports even though detours have to be made.

<table>
<thead>
<tr>
<th>Replication</th>
<th>Total stock</th>
<th>Total transports</th>
<th>Objective value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Node 1</td>
<td>Node 2</td>
<td>Node 3</td>
</tr>
<tr>
<td>1</td>
<td>260</td>
<td>520</td>
<td>260</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>90</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 3.5: Results for Example 2.

This example lead to two conclusions: first, it shows that it is possible to compensate a lack of containers by additional transports, as long as enough time is left to move the containers between the nodes before and after they are needed. Secondly, that (free) take-along transports are preferred to dedicated transports, as long as there is enough time for detours that have to be made, because the total transport costs are then reduced. From a network flow perspective, the optimal solution always uses the cheapest path to move containers; the costs for a path consist of the sum of transport and inventory costs that a container causes on its way. Take-along transports add additional connections between the nodes and are preferred to other parallel connections if the total costs of a path decrease as a result.

### 3.3.4 Example 3: Transport versus Purchasing

The next example is again based on the simple network shown in Figure 3.6. Now the decision about how many containers to purchase is added to the optimisation problem. This requires the container purchasing extension introduced in Section 3.2.1.
3.3 Numerical Results

The transport costs per container and period are constant at a level of two, inventory costs are disregarded. In this example all containers must be bought in the beginning, so there is no initial stock. Containers can be purchased at each of the nodes in each period, but they cannot be sold again.

Two aspects are examined here. First, the price per container \( c_{acqi} \) is varied and secondly, the length of the planning horizon is extended. These two parameters influence the cost-benefit ratio of the containers.

The results are listed in Table 3.6. When containers are relatively cheap (replications one and two), the best solution is to buy a sufficient number of containers for each location to cover the demand without the need for transport. From a price of 20 on, it is better to buy less containers and instead transport them between Nodes 2 and 3. Transports between Nodes 1 and 3 are still not profitable. They are more expensive than buying additional containers because of the longer distance. But, as soon as the price for containers doubles again (in replication five), these transports also become preferable to the expenses for another 10 containers. Within the planning horizon, the containers from Node 3 have to be transported five times to the second node and four times back. So nine movements with total costs of \( 9 \cdot 10 \cdot 2 = 180 \) can be saved when 10 additional containers are bought. This is the cheaper alternative when their costs per piece are below 18.

If the planning horizon is extended to 60 (or 90) periods, 19 (or 29) transport are necessary with total costs of 380 (or 580). So it is preferable to buy 10 additional containers up to a price of 38 (or 58 respectively).

This example shows that it is only optimal to provide additional containers as long as their price is lower than the costs of covering the demand by transporting containers between the different locations. This trade-off is the main reason why it makes sense to include both the transport and purchasing decision into the same optimisation model. The variation of the number of planning periods shows a problem that arises with this extension: the longer the planning horizon, the more containers are purchased to achieve the optimal solution. This happens, because with a longer planning horizon more opportunities exist to employ additional containers while the costs are independent of the planning horizon. It should be
### Table 3.6: Results for Example 3.

<table>
<thead>
<tr>
<th>Node</th>
<th>Replication periods</th>
<th>Optimal total stock</th>
<th>Optimal total acquisitions</th>
<th>Optimal total stock</th>
<th>Optimal total acquisitions</th>
<th>Optimal total stock</th>
<th>Optimal total acquisitions</th>
<th>Optimal total stock</th>
<th>Optimal total acquisitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
</tr>
<tr>
<td>1 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
</tr>
<tr>
<td>2 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
</tr>
<tr>
<td>3 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
</tr>
<tr>
<td>4 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
</tr>
<tr>
<td>5 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
</tr>
<tr>
<td>6 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
</tr>
<tr>
<td>7 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
</tr>
<tr>
<td>8 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
</tr>
<tr>
<td>9 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 290</td>
<td>0 290</td>
</tr>
<tr>
<td>10 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 290</td>
<td>0 290</td>
<td>0 290</td>
<td>0 290</td>
</tr>
<tr>
<td>11 300</td>
<td>0 290</td>
<td>0 300</td>
<td>0 290</td>
<td>0 290</td>
<td>0 290</td>
<td>0 290</td>
<td>0 290</td>
<td>0 290</td>
<td>0 290</td>
</tr>
</tbody>
</table>

**Objective:**
- Minimize total costs.
- Maximize total stock.

**Planning horizon:**
- 15 periods
mentioned that this effect would become less apparent, if inventory costs were regarded. Each additional container would lead to more inventory costs over his lifetime then, which would be an incentive not to acquire as much containers.

To cope with the problem that the planning horizon influences the number of acquired containers, the real prices for containers should be replaced by the depreciation over the planning horizon plus the capital costs that arise for the time the containers are used. This way, the costs are not dependent on the length of the planning horizon.

### 3.3.5 Example 4: Transport versus Leasing

Similar to container purchases, the leasing of containers can also be demonstrated with the example network in Figure 3.6.

For this example the inventory costs are again assumed to be zero and the transport cost factor equal to two. Nodes 1 and 3 are equipped with an initial stock of 10 containers each, which is enough to cover the demand of these nodes. At Node 2, no initial stock is available, so the demand has to be met with rented containers, or with containers that are obtained from the other two nodes. The model is extended by the option to lease containers as explained in Section 3.2.2. The rent per period and container is varied from zero to ten as listed in Table 3.7 which also shows information about the optimal solution of the different replications.

In the first two replications the rental fee is so low that it is best to lease the needed containers at Node 2. When the rental fee reaches the value of two, containers from Node 3 are used, which are are no longer needed there at the end of the planning horizon. In this case, the transport costs from the third node to the second (distance $2 = 1 \cdot 2$) are less or equal to the renting costs. But it is still not optimal to transport containers from Node 3 to two and back, because then the transport costs would equal four per unit. This is done with a renting fee of four or more, which explains the lower number of rented containers and the higher transport amount in replications five to eight. With even higher renting fees, of eight and above, it is also best to use containers from the first node at the second one. Therefore, from
3 Extensions to the Basic Model

<table>
<thead>
<tr>
<th>Replication</th>
<th>( c_{\text{rent}} )</th>
<th>Optimal total stock</th>
<th>Number rented</th>
<th>Optimal total transports</th>
<th>Objective value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>260 0 260</td>
<td>100 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>260 0 260</td>
<td>100 0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>260 20 220</td>
<td>90 10</td>
<td>100 290</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>260 20 220</td>
<td>90 10</td>
<td>100 290</td>
<td>380</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>210 40 100</td>
<td>40 100</td>
<td>100 380</td>
<td>290</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>210 40 100</td>
<td>40 100</td>
<td>100 380</td>
<td>380</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>210 40 100</td>
<td>40 100</td>
<td>100 380</td>
<td>420</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>210 40 100</td>
<td>40 100</td>
<td>100 380</td>
<td>460</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>10 40 100</td>
<td>0 180</td>
<td>180 540</td>
<td>500</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>10 40 100</td>
<td>0 180</td>
<td>180 540</td>
<td>540</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>10 40 100</td>
<td>0 180</td>
<td>180 540</td>
<td>540</td>
</tr>
</tbody>
</table>

Table 3.7: Results for Example 4.

this point on, no containers are rented.

Summing up, containers are rented if this is cheaper than collecting them at other nodes and moving them back after they are no longer used. Apart from this, rented containers can also be used to compensate a short-term lack of capacity at certain locations.

3.3.6 Summary and Recommendations

In the previous sections, different combinations of the basic model and the model extensions were applied to small example networks to show their general behaviour.

The first two examples demonstrated the logic of the transportation planning decision: a flow of containers should always be routed via the cheapest path over the considered locations and planning periods. If the inventory costs are high or if movements are relatively cheap, it can be optimal to transport containers, even if they are not needed on the destination site but at another location. This is particularly the case if take-along opportunities are available, so that detours effectively reduce the total costs.

Within the remaining examples it could be seen that the results are mainly driven by
the trade-off between more transportation and more capacity. A container demand
can either be covered by capacity measures such as acquisitions or leasing or by
additional transports. Which option is chosen is determined by the costs of moving
the containers to several nodes versus the costs of capacity extensions.

Example 3 showed that the acquisition of containers is dependent on the length
of the planning horizon, because of the inventory costs induced by the existence
of additional containers, and because of the opportunity to employ them more
often the longer the planning horizon is. To avoid the latter effect, for container
acquisitions not the real prices should be taken into consideration, but better a cost
factor, such as the depreciation, which is independent of the planning horizon.

The results of the experiments show general characteristics of the planning problem.
Therefore they also remain valid when the linear transport costs are substituted
by non-linear costs such as the fixed-step costs which are introduced in the next
chapter.
4 Planning under Fixed-Step Transport Costs

The approach developed in the preceding chapters can be used to optimise the planning of transportation and of capacity adjustments in a container system. However, the costs are still assumed to be linear, although it was mentioned in the introduction that this is often not the case in practice. In this chapter, an approach is developed to tackle this challenge. It is meant to be used instead of the basic model and can also be extended with the model extensions shown in the previous chapter.

To understand the problem, one should recall that containers are often not transported individually but in larger amounts, because a transport vehicle can carry several containers at once with nearly the same costs as if only one would be transferred. This is especially true for small containers, such as nestable or folding boxes or the specialised skeleton containers which are widely used for the supply of parts in automotive industries. Many types of containers are designed to require as little transport capacity as possible in their empty state. In such cases, a poor vehicle utilisation leads to a large increase in transport costs.

Such containers can often be transported on the remaining vehicle space when goods are moved to the same destination. This would not afford further changes to the model but the method proposed for take-along transports (see Section 3.1.3) could be used. Nevertheless, as soon as additional transport movements for empties become necessary, or if transports of empty containers utilise resources that would otherwise be used for other purposes, it can be beneficial to consider the vehicle capacity and model the resulting non-linear transport costs explicitly.
This chapter is organised as follows: the characteristics of the transport costs for bundled container movements are discussed first. An adapted model is then proposed to allow for fixed-step transport costs. In a third section, the solution procedures for the model are discussed. And in the last section, numerical tests are presented to demonstrate certain model characteristics and also to compare the performance of the various solution approaches.

<table>
<thead>
<tr>
<th>Data:</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport costs per-vehicle</td>
</tr>
<tr>
<td>vehicle capacity</td>
</tr>
<tr>
<td>$c_{i,j}^{\text{transport}}$</td>
</tr>
<tr>
<td>$\forall i, j \in N$</td>
</tr>
</tbody>
</table>

Table 4.1: The additional symbols used for the model with fixed-step transport costs (note the changed meaning of the transport cost parameter).

### 4.1 Modelling Transport Costs

All costs discussed before were assumed to be proportional to a decision variable. This kind of cost is very comfortable for optimisation purposes, although it might not be perfectly realistic. However, linear costs completely ignore the characteristics of transports with multiple containers on the same vehicle. Therefore, in this section, some options are shown to reformulate the transport costs. First, the real structure of the transport costs as fixed-step costs is discussed. Then, some alternative ways to model the costs are presented, that result in more simple models, but also imply some drawbacks.

#### 4.1.1 Fixed-Step Transport Costs

Once a vehicle capacity allows to carry multiple containers, this leads to costs that are not proportional but increase stepwise. Costs arise for every $k$ units to be
4.1 Modelling Transport Costs

transported, where \( k \) is the maximum number of containers that can be transported at once. So, the transport costs can be written in the following way:

\[
\left[ \frac{x_{i,j,t}}{k} \right] \cdot c_{i,j}^{\text{transport}}
\]  

(4.1)

Here \( c_{i,j}^{\text{transport}} \) is the amount of costs that arise for every \( k \) containers or part thereof. So the resulting cost function increases stepwise as shown in Figure 4.1.

![Figure 4.1: Fixed-step transport costs.](image1)

![Figure 4.2: Costs per unit with fixed-step transport costs.](image2)

Obviously this cost structure leads to a non-linear and non-continuous cost function. It is not even convex, since the average costs rise whenever a new vehicle is needed.
4 Planning under Fixed-Step Transport Costs

and then decrease until its capacity is exhausted. This can be seen in the plot of the average transport costs in Figure 4.2. So the former linear programming model is turned into a much more complex one: a non-linear integer program. As this type of cost function is integrated into one modelling approach for fixed-step costs (see Figure 4.4), this is discussed below.

4.1.2 Decreasing Marginal Costs

As a first idea for simplification, the transport costs per unit could be assumed to be a continuously decreasing function of the transport quantities, leading to total transport costs as shown in Figure 4.3. This way, an incentive is included in the model to increase the number of containers moved via the same connection to yield a ‘quantity discount’. At the same time, the objective function is still continuous, but non-linear. Of course, this is only a rough approximation of the real costs. So the optimal solution of this problem does not necessarily show the optimal transportation batch size. On the one hand the decreasing transport costs are an incentive to transport more containers at once. But on the other hand situations are neglected where some more containers lead to a high increase of costs because an additional transport vehicle is needed.

![Figure 4.3: Continuously decreasing transport cost function.](image)

Also another problem arises with this approach. In practice, the real costs of transportation are linear – not in the number of containers transported, but in
4.1 Modelling Transport Costs

the number of vehicles used. In Figure 4.3 these costs are shown as black dots (at a transport quantity of 0, \(k\) and \(2k\)), connected by a straight dotted line. If the marginal costs are defined as decreasing this would lead to a strictly concave cost function starting at zero (shown as a solid line in the figure). Per definition every strictly concave function first runs above a straight line up to the point where it crosses it and then continues below it. As a result, the transport costs would be overestimated for small quantities, and underestimated for larger quantities. By shifting the point of intersection to the right and thus flattening the curve the deviations would decrease. But then also the incentive to transport larger quantities would become less attractive.

Due to these difficulties this approach of modelling transport costs is not used here.

4.1.3 Fixed Charges

As another extreme, one could add a fixed charge to the objective function that arises if a connection between two nodes is used as shown in term 4.2. An additional binary decision variable \(I_{i,j,t}\) would track the use of each connection. The additional constraint 4.3 would be needed to connect the new decision variable to the transported quantities \(x_{i,j,t}\).

\[
I_{i,j,t} \cdot c_{i,j}^{\text{fix}} \quad \text{(In the objective function)} \quad (4.2)
\]

\[
x_{i,j,t} = I_{i,j,t} \cdot x_{i,j,t} \quad \text{(As additional constraint)} \quad (4.3)
\]

\[
I_{i,j,t} \in \{0, 1\} \ \forall i, j, t \quad (4.4)
\]

This modelling technique is identical to the setup costs in a lot-sizing decision. In this context it has two disadvantages: from the technical perspective it makes the optimisation problem much more challenging, because the new set of binary decision variables lead to a combinatorial problem. And from the logical perspective another difficulty arises: only one transport per edge and period would be represented in the cost function, while the capacity of that connection is not taken into consideration. If the optimal value of a transport quantity exceeded the capacity
of a vehicle, another vehicle would have to be provided to implement the plan, generating higher costs than expected.

Nevertheless there is already literature available, which uses this type of cost structure. In particular, the ‘fixed charge network flow problem’ needs to be mentioned here. It is mathematically similar to the models proposed here and includes fixed costs as explained above (see Nemhauser and Wolsey 1988, pp. 8 f. and 495 ff.).

### 4.1.4 Further Methods

As an alternative to a reformulation of transport costs, additional constraints could also be added to enforce a better utilisation of vehicles.

One option is to predefine a fixed batch size for the transports. Then, only fully loaded vehicles would be suggested as result of this optimisation. This is a convenient method to enforce efficient operation of the transport vehicles. However, the model would lack an option to allow movements of partial loads, which can be necessary to collect containers from rarely visited network nodes, or in situations where it is cheaper to move a small amount of containers from a nearby node, instead of requesting a full load from a distant location. This issue could be bypassed by doubling all connections in the network and allowing both cheap full loads and expensive individual movements of containers.

Alternatively, a minimum transport amount could be set as a threshold. As this would not be a simple lower limit for the variables, but a minimum amount that only needs to be regarded in case of a transport on a connection, it is necessary to introduce additional binary variables. These variables $I_{i,j,t}$ would then indicate a transport similar to the one already suggested for the fixed charge approach in the last section:

\[
\begin{align*}
  x_{i,j,t} &= I_{i,j,t} \cdot x_{i,j,t} && (x > 0 \text{ enforces } I = 1) \quad (4.5) \\
  x_{i,j,t} &\geq I_{i,j,t} \cdot k \quad \text{(Minimum transport quantity)} \quad (4.6) \\
  I_{i,j,t} &\in \{0, 1\} \quad \forall i, j, t \quad (4.7)
\end{align*}
\]
4.2 A Model with Fixed-Step Transport Costs

The first of the two new constraints ensures that every positive amount to be transported is depicted in a value of 1 for the corresponding variable \( I_{i,j,t} \). The second constraint only becomes relevant if \( I_{i,j,t} = 1 \). In this case, it is ensured that at least the minimum transport amount of \( k \) is sent over the connection. The problem with this approach is, that it is only slightly more flexible than a fixed transportation batch size, but is computationally much more complex, because of the new integer variables. The only advantage in terms of planning flexibility is, that an arbitrary amount may be added once it has been decided to send a certain number \((k)\) of containers.

4.2 A Model with Fixed-Step Transport Costs

In the previous section, multiple approaches were shown to model transport costs. In the following, fixed-step transport costs as discussed in Section 4.1.1 are assumed. This section intends to present two alternative formulations for the basic model of Chapter 2, which incorporate the ‘real’ stepwise structure of the transport costs. They can also be used as the mathematical basis for the transportation planning and be extended as described in Chapters 3.1 and 3.2.

The first subsection introduces the model formulations. A numerical example then demonstrates the characteristics of the changed optimisation problem. In the concluding Subsection 4.2.2, it is shown that the new model no longer features the unimodularity property.

4.2.1 Model Formulation

In a first approach, the new cost function is included into the objective function which results in the model listed in Figure 4.4 on the following page. In this formulation the objective function inherits the properties of the integrated transport costs function, which is neither linear, nor continuous, nor convex. This complicates the process of finding an appropriate optimisation algorithm. Therefore an alternative option is discussed to model the transport costs in a more efficient way.
Model: Transportation planning under fixed-step costs (non-linear version)

\[
\begin{align*}
\min & \quad \sum_{i\in T} \sum_{j\in N} \sum_{t\in T} c_{i,j,t}^{\text{transport}} (x_{i,j,t}) + \sum_{i\in T} \sum_{t\in N} y_{i,t} \cdot c_{i,t}^{\text{stock}} \\
\text{transport costs} & \quad \text{inventory costs}
\end{align*}
\]

\[\text{subject to the cost function,}\]
\[c_{i,j,t}^{\text{transport}} (x) = \left[ \frac{x}{k} \right] \cdot c_{i,j,t}^{\text{transport}}\]

\[\text{conservation of flow,}\]
\[y_{i,0} = s_{i,0} - \sum_{j \in N, j \neq i} x_{i,j,0} \quad \forall i \quad (4.10)\]
\[y_{i,t} = y_{i,t-1} + s_{i,t} - d_{i,t} + \sum_{j \in N, j \neq i, m_{i,j} \leq t} x_{j,i,t-m_{i,j}} - \sum_{j \in N, j \neq i} x_{i,j,t} \quad \forall i, t > 0 \quad (4.11)\]

\[\text{demand fulfilment and inventory constraints}\]
\[0 \leq y_{i,t} \leq y_{i,t}^{\text{max}} \quad \forall i, t \quad (4.12)\]

\[\text{and upper and lower bounds of transport quantities.}\]
\[0 \leq x_{i,j,t} \leq x_{i,j,t}^{\text{max}} \quad \forall i, j \neq i, t \quad (4.13)\]
4.2 A Model with Fixed-Step Transport Costs

Instead of adding a non-linear objective function term, the number of used transport vehicles can be modelled explicitly, with a new integer decision variable $f_{i,j,t}$. A vehicle movement leads to costs of $c_{i,j}^{\text{transport}}$ per vehicle. So, the total transport costs are assumed to remain linear. In the objective function, the following term substitutes the transport cost term:

$$\sum_{i \in I} \sum_{j \in J} \sum_{t \in T} f_{i,j,t} \cdot c_{i,j}^{\text{transport}}$$

(4.14)

In an additional constraint, the transport quantities are linked to the number of employed vehicles. Here it is assumed that all vehicles have a unique capacity of $k$ empty containers:

$$x_{i,j,t} \leq f_{i,j,t} \cdot k$$

(4.15)

For this extension, it is not necessary to modify the conservation of flow constraint. The flow of containers is still tracked by the variable $x_{i,j,t}$ and the new constraint 4.15 guarantees that the capacity of vehicles is not exceeded. The full model resulting from this extension is listed in Figure 4.5 on the following page.

Both models presented in this section are identical in the sense that they both express exactly the fixed-step transport costs, so that optimal transport quantities can be obtained. The two formulations differ only in the algorithms that can be used to solve them. The second model is solvable as a linear program while the first one is not. Hence, the second model is utilised throughout the following sections.

4.2.2 Unimodularity Property

The model remains a linear program, although the fixed-step transport costs are explicitly included. Unfortunately, unlike the basic model, this model must be solved as an integer program, because in the only meaningful case where $k \not\in \{-1, 0, 1\}$ the coefficient matrix is not totally unimodular.

**Proof.** Using the Unimodularity Theorem (Ahuja, Magnanti and Orlin 1993, p. 448)
4 Planning under Fixed-Step Transport Costs

Model: Transportation planning under fixed-step costs (linear version)

\[
\begin{align*}
\min_{\substack{f_{i,j,t},\ x_{i,j,t},\ y_{i,t}\ \forall i,j\in\mathbb{N},\ t\in T}} & \sum_{i\in\mathbb{N}} \sum_{j\in\mathbb{N}, j\neq i} \sum_{t\in T} f_{i,j,t} \cdot c_{i,j} + \sum_{i\in\mathbb{N}} \sum_{t\in T} y_{i,t} \cdot c_{i} \\
\text{subject to} & \\
\qquad & x_{i,j,t} \leq f_{i,j,t} \cdot k \quad \forall i, j \neq i, t \\
\qquad & y_{i,0} = s_{i,0} - \sum_{j\in\mathbb{N}, j\neq i} x_{i,j,0} \quad \forall i \\
\qquad & y_{i,t} = y_{i,t-1} + s_{i,t} - d_{i,t} + \sum_{j\in\mathbb{N}, j\neq i} x_{j,i,t-m_{j,i}} - \sum_{j\in\mathbb{N}, j\neq i} x_{i,j,t} \quad \forall i, t > 0 \\
\qquad & 0 \leq y_{i,t} \leq y_{i,t}^{\max} \quad \forall i, t \\
\qquad & 0 \leq x_{i,j,t} \leq x_{i,j,t}^{\max} \quad \forall i, j \neq i, t \\
\end{align*}
\]

Figure 4.5: Model for transportation and inventory planning under fixed-step costs; formulation with linear objective function, but additional integer variables.
it can be shown that the coefficient matrix of the model with fixed-step costs (see Figure 4.5) is not unimodular (and thus also not totally unimodular).

The Unimodularity Theorem states that an integer matrix $A$ with linearly independent rows is unimodular if, and only if, every basic feasible solution defined by the constraints $Ax = b, x \geq 0$ is integer for any arbitrary integer vector $b$.

From restriction 4.17 (see Figure 4.5) results the following equation as part of the equation $Ax = b$. According to the Unimodularity Theorem, every feasible combination of $x_{i,j,t}$ and $f_{i,j,t}$ must be an integer for any integer number $b$.

$$x_{i,j,t} - k \cdot f_{i,j,t} = b \quad (4.23)$$

So, assuming that $k > 0$,

$$f_{i,j,t} = \frac{x_{i,j,t} - b}{k} \quad (4.24)$$

must be an integer for every integer $b$. If one would take for example $b = 0$, the equation

$$f_{i,j,t} = \frac{x_{i,j,t}}{k} \quad (4.25)$$

does only define an integer value for $f_{i,j,t}$ if $x_{i,j,t}$ is divisible by $k$. If one then decrements the number $b$ to $-1$ the equation would become

$$f_{i,j,t}' = \frac{x_{i,j,t} + 1}{k}. \quad (4.26)$$

In this case $f_{i,j,t}'$ must also be an integer because, if the combination of $f_{i,j,t}$ and $x_{i,j,t}$ is part of a feasible solution, then also each solution with $f_{i,j,t}' > f_{i,j,t}$ is feasible. There is no constraint in the model that forms an upper limit to the variables $f_{i,j,t}$. But, obviously in equation 4.26, $f_{i,j,t}'$ cannot be an integer if $k \notin \{-1, 1\}$, because 1 and $-1$ are the only numbers which divide two consecutive integers evenly. So, according to the definition in the Unimodularity Theorem, the coefficient matrix of the model is not unimodular.
4 Planning under Fixed-Step Transport Costs

However, it needs to be shown that all assumptions of the Unimodularity Theorem are met. The theorem only holds, if $A$ has linearly independent rows. For the coefficient matrix of the problem in Figure 4.5 this is the case. The definition of linear dependence claims that there must be a linear combination of the row vectors $a_i$ of $A$ with not all factors $c_i$ being zero that sums up to the zero vector:

$$c_1 a_1 + c_2 a_2 + \ldots + c_n a_n = 0$$

(4.27)

This definition can be used to prove row by row that it is not possible to find such a linear combination for the coefficient matrix of the model. Each of the rows represented by constraint [4.17] have a coefficient (for $f_{i,j,t}$) which is unique for this particular row, so these rows cannot be part of such a linear combination; i.e. if the factor for that particular vector was different from zero, the corresponding element of the resulting vector would also not be equal to zero. So, the result would not be the zero vector. The row vectors corresponding to the conditions [4.18] and [4.19] can also not be composed into such a linear combination. If all of them are combined, the last row vector has an element (for $y_{i,T}$) that is unique for this vector. So, any linear combination of these vectors with only non-zero factors adds up to a vector which is different from the zero vector. If the last line is left out of the linear combination (i.e. setting its factor to zero), the remaining subset of the row vectors also has unique elements ($y_{i,T-1}, y_{i,T-2}, \ldots$) and thus is also not linearly dependent.

So the constraint matrix of the model with fixed-step costs is not unimodular and therefore can also not be totally unimodular, the model does not have the integrality property.

This shows that the continuous linear programming relaxation does not necessarily satisfy the integrality constraints. It cannot be guaranteed to have an integral solution. Even if large numbers of containers made it possible to accept fractional values for the container quantities as an approximate solution, nevertheless the number of vehicles used would always be comparatively small. Therefore, it must have integer values for a solution to be useful. An integer optimisation algorithm

---

80
should be employed to optimise this model.

### 4.2.3 A Numerical Example

The last section introduced a model formulation to allow for fixed-step transport costs. In this section a small example is presented to demonstrate how this type of costs changes the optimal solution of the planning problem compared to linear costs. First, the basic model with linear costs (see Figure 2.1 on p. 38) is applied to the data. Secondly, the number of vehicles moved in this solution is calculated to obtain the real costs that would arise if it was implemented in an environment where the costs follow a fixed-step structure. Thirdly, the mixed-integer model with fixed-step transport costs (see Figure 4.5 on p. 78) is used to calculate the cost-optimal solution. It is shown that the optimal solutions differ more or less depending on the vehicle size.

The example chosen here is based on the small network used for Example 1 in Section 3.3. It spans three network nodes, two close to each other and one further away. Containers still seized, used for one period, and then released. This is repeated at the different locations at regular intervals but with a changed pattern compared to the example in Section 3.3. The network structure and container demands are depicted in Figure 4.6. The initial supply of containers only amounts to 20. This is just enough to cover the demand throughout the planning horizon.

![Figure 4.6: Example network – locations, distances and container demands.](image)
but only if the containers are transported between the nodes once they are released. The containers are either used at Nodes 2 and 3 (first in periods 0 and 1) or at Node 4 (first in period 4).

The transport costs are assumed to be equal to the distance between the several nodes multiplied by 2, while the inventory costs are initially set to zero:

\[ c_{i,j}^{\text{transport}} = m_{i,j} \cdot 2 \]
\[ c_{\text{stock}} = 0 \]

This experiment focuses on the impact of the vehicle capacity parameter \( k \) on the optimal solution. It shows, that this parameter is the decisive factor which determines whether the optimal solutions of the two model formulations differ. The vehicle capacity parameter is varied between 10 and 100.

The optimal solutions to the container flow problem for \( k > 10 \) are delineated in Figure 4.6. The solution represented by solid black arrows shows the container movements which are optimal if fixed-step costs are considered. The dashed grey arrows symbolise the optimal flows produced by the basic model for the case with linear transport costs.

<table>
<thead>
<tr>
<th>( k )</th>
<th>Basic model</th>
<th>Fixed-step costs model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total stock</td>
<td>Total transports</td>
</tr>
<tr>
<td>10</td>
<td>120</td>
<td>160</td>
</tr>
<tr>
<td>50</td>
<td>120</td>
<td>160</td>
</tr>
<tr>
<td>100</td>
<td>120</td>
<td>160</td>
</tr>
</tbody>
</table>

Table 4.2: Results for different vehicle capacities.

The results of the optimisations are listed in Table 4.2. For a better understanding, it is necessary to recall the total costs for a transport in both of the model formulations. For the basic model with linear costs, it is assumed that the per-item transport costs are proportional to the quantity, regardless whether a vehicles is loaded with only one item or fully. So the costs to move a quantity of \( x \) containers are:

\[ \frac{x}{k} \cdot c_{i,j}^{\text{transport}} = \frac{x}{k} \cdot m_{i,j} \cdot 2 \]
4.2 A Model with Fixed-Step Transport Costs

The model with fixed-step costs assumes the following transport costs for the same quantity:

\[
\left\lfloor \frac{x}{k} \right\rfloor \cdot c_{i,j}^{\text{transport}} = \left\lfloor \frac{x}{k} \right\rfloor \cdot m_{i,j} \cdot 2
\]

This means that when only one item is transported, the full costs for a vehicle movement are already considered.

These assumptions are reflected in the numerical results. First, for a small vehicle size of 10, there is no difference between the solutions. In this scenario, all vehicles are fully loaded, because all the demand and supply quantities are multiples of \( k \). In consequence, they are only moved in amounts which exactly equal one or more full vehicle loads.

For larger vehicle capacities the solutions differ. The integer program tends to avoid the partial loads and prefers to combine movements. This leads to the flow indicated above in Figure 4.6. The model with linear costs adheres to the same solution, the change in vehicle capacity only leads to a corresponding change in the total costs. When the total number of vehicles is calculated from the transport quantities, it becomes apparent that the costs assumed by the continuous model are much less than the real costs imposed by the solutions. The transport costs are underestimated and therefore they are not minimised correctly.

The central message of this example is, that it is important to use the correct modelling of fixed-step transport costs if this corresponds to the characteristics of a container system. The larger the vehicle capacity is, the larger is the difference between the assumed linear per-item costs and the real transport costs. As a result of this, the continuous model does not allow for grouping different transports in order to utilise less vehicles. So, the larger the vehicle capacity is, the larger is also the need for the mixed-integer model with fixed-step costs.
4 Planning under Fixed-Step Transport Costs

4.3 Solution Algorithms

In this section the question of how the model can be solved efficiently is discussed. Due to its complexity, the problem cannot be solved quickly with a standard solver for networks with many nodes or if the planning horizon covers more than a few periods.

After discussing general properties of the model in the next section, special conditions are shown in Section 4.3.2 which hold true for feasible and optimal solutions of the model. These conditions can be used to speed up solution algorithms. Another useful property of the model is discussed in Section 4.3.3: the linear programming relaxation can always be used to generate a feasible solution to the integer program. Throughout the Section 4.3.4 and 4.3.5, a tailored branch-and-bound algorithm and – based on that – a branch-and-cut procedure are presented. The efficiency of these algorithms and the best settings are shown in Section 4.4.

4.3.1 Model Properties and Complexity

As mentioned above, the model with fixed-step costs (Figure 4.5) is a (mixed) integer program with a coefficient matrix that is not totally unimodular in the general case. That means that the problem is NP-complete (Schrijver 2000, pp. 245–247) and needs to be solved with integer optimisation algorithms. In fact, tests show that it is difficult to solve this problem with even a moderate network size in an acceptable time. To show the influence of the data on the effort needed to find an optimal solution, one could take a look at the number of variables listed in Table 4.3. The most critical group of variables is the number of transport vehicles $f_{i,j,t}$. If a solution algorithm chose any (integer) set of transport vehicles, the other variables would also be integer. If all $f_{i,j,t}$ are fixed, the problem is reduced to the basic model of Chapter 2, which shows the integrality property. This is why this problem does not need to be solved as an integer program but only as a mixed-integer program.

The amount of variables is determined by two parameters. The first is the number of nodes $N$. It can be seen as fixed for many real problems as it would be difficult to
4.3 Solution Algorithms

<table>
<thead>
<tr>
<th>Variable set</th>
<th>Number of variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{i,j,t} )</td>
<td>( N^2 \cdot T )</td>
</tr>
<tr>
<td>( x_{i,j,t} )</td>
<td>( N^2 \cdot T )</td>
</tr>
<tr>
<td>( y_{i,t} )</td>
<td>( N \cdot T )</td>
</tr>
</tbody>
</table>

Table 4.3: Number of variables in the model with fixed-step costs.

only partially optimise a company’s network. The other determinant is the number of periods \( T \). Table 4.3 shows the impact of these parameters on the number of variables. At first it seems that the number of periods only has a linear influence on the problem size. But, due to the fact that no algorithms are known to solve the problem in polynomial time, each increase in the number of periods potentially means a multiplication of time needed to solve the problem.

The model with fixed-step costs remains identical to a minimum cost flow model but with one important extension: it features the non-linear costs. For this particular type of problem no special solution algorithms are known, which is why a new approach is developed in this section.

4.3.2 Feasibility and Optimality Conditions

For this mixed-integer problem some additional restrictions can be specified that are known to be met by an integer solution, but not necessarily by a solution with continuous variable values. In the following, some inequalities are shown that provide additional bounds to the integer variables, i.e. to the number of transport vehicles \( f_{i,j,t} \). These inequalities represent necessary optimality conditions that must be fulfilled in an optimal solution to the problem with fixed-step costs as it was delineated in Figure 4.5.

The conditions focus on subsets of the network nodes, as illustrated in Figure 4.7 in order to identify constraints which need to be met by the flows entering or leaving these subsets. The conditions are explained in detail in the following.
4 Planning under Fixed-Step Transport Costs

Figure 4.7: Subsets of nodes focused by the feasibility conditions. Nodes $C_1$ and $C_2$ form a segment as it is treated by condition 1; conditions 2 and 3 deal with single nodes like $A_3$.

Feasibility Condition 1: Provision of Net Demand over Time

A first approach to tighten the lower bound focuses on individual network nodes. The net demand of containers up to each period is summed up for each node. If this cumulated net demand is positive, this means it must be supplied in time by transports from other nodes. And then, if transports have to be planned, the sum of incoming transport vehicles until the period under consideration must be at least equal to the number of vehicles that would be needed to carry exactly the net demand. This is expressed by inequality 4.28.

\[
\sum_{j} \sum_{\tau=0}^{t-m_{i,j}} f_{j,i,\tau} \geq \left\lceil \frac{\sum_{\tau=0}^{t} d_{i,\tau} - s_{i,\tau}}{k} \right\rceil \quad \forall i, t \quad (4.28)
\]

Proof. One of the constraints of the model with fixed-step costs is the conservation of flow condition 4.19 (see Figure 4.5), which can be transformed to express the
incoming transports at a node for a certain period as

$$\sum_{j\in N, \ j\neq i, \ m_{ij} \leq t} x_{j,i,t-m_{ij}} = y_{i,t} - y_{i,t-1} + d_{i,t} - s_{i,t} + \sum_{j\in N, \ j\neq i} x_{i,j,t} \quad (4.29)$$

So, using also constraint [4.18] of the model, the sum of transports that arrive at a node up to a period $t$ is

$$\sum_{\tau=0}^{t} \sum_{j\notin N, \ j\neq i, \ m_{ij} \leq t} x_{j,i,t-m_{ij}} = \sum_{\tau=0}^{t} y_{i,\tau} - \sum_{\tau=0}^{t-1} y_{i,\tau-1} + \sum_{\tau=0}^{t} (d_{i,\tau} - s_{i,\tau}) + \sum_{j\in N, \ j\neq i} x_{i,j,\tau} \quad (4.30)$$

Because the final stock, as well as the outgoing transports, are assumed to be positive, it can be concluded that the sum of incoming transports must be at least as high as the net demand:

$$\sum_{\tau=0}^{t} \sum_{j\notin N, \ j\neq i, \ m_{ij} \leq t} x_{j,i,t-m_{ij}} \geq \sum_{\tau=0}^{t} (d_{i,\tau} - s_{i,\tau}) \quad (4.32)$$

The following term results from the vehicle capacity constraint [4.17]

$$f_{j,i,t} \geq \frac{x_{j,i,t}}{k} \quad (4.33)$$

Summing up over all periods up to $t - m_{ij}$ and recalling the results from [4.31] one
4 Planning under Fixed-Step Transport Costs

can easily see that also

\[ \sum_{j} \sum_{\tau=0}^{\tau_{m,j}} f_{j,i,\tau} \geq \frac{\sum_{\tau=0}^{\tau_{i}} (d_{i,\tau} - s_{i,\tau})}{k} \]  

(4.34)

holds. Considering that the number of transporters \( f_{i,j,t} \) has to be an integer, this is equivalent to

\[ \sum_{j} \sum_{\tau=0}^{\tau_{m,j}} f_{j,i,\tau} \geq \left\lceil \frac{\sum_{\tau=0}^{\tau_{i}} (d_{i,\tau} - s_{i,\tau})}{k} \right\rceil \]  

(4.35)

This is essentially the same as stated in relation 4.28, with only a modified indexing.

\[ \square \]

The restriction 4.28 provides a lower bound to the number of transporters that is met by any feasible solution of the problem with fixed-step costs. It does not contain the amount of stock \( y_{i,j,t} \) which must still be considered as continuous. Of course, it should only be introduced for pairs \((i, t)\) where the net demand \( \sum_{\tau=0}^{\tau_{i}} d_{i,\tau} - s_{i,\tau} \) is greater than zero. In all other cases, the inequality would also be met because the right side would then have a value less or equal to zero, but the new conditions would be redundant because for all \( f_{i,j,t} \) a non-negativity constraint already exists.

A similar condition, but without rounding up, has also been identified for the related fixed-charge network flow problem (see Nemhauser and Wolsey 1988, p. 498; Padberg, Roy and Wolsey 1985).

Feasibility Condition 2: Treatment of Net Supply

Another bound can be found for all nodes in each period if a net demand or a net supply exist. If a net supply exists, the containers must either be transported or kept in stock. This idea can be expressed by the following inequality:

\[ y_{i,t} + \sum_{j} f_{i,j,t} \geq \left\lceil \frac{s_{i,t} - d_{i,t}}{k} \right\rceil \quad \forall i, t \text{ with } s_{i,t} > d_{i,t} \]  

(4.36)
4.3 Solution Algorithms

In the case of \( y_{i,t} = 0 \), this is a good lower bound for the number of vehicles. It forces the value of \( f_{i,j,t} \) to be lifted above the continuous value in the linear programming solution. Otherwise (for \( y_{i,t} > 0 \)) the inequality is less useful but still holds.

**Proof.** Using constraints 4.18 and 4.19 of the model, the number of containers on outgoing transports for a certain node and period can be expressed as

\[
\sum_{j \in N, \ j \neq i} x_{i,j,t} = s_{i,t} - d_{i,t} + \sum_{j \in N, \ j \neq i, \ m_{j,t} \leq t} x_{j,i,t-m_{j,t}} + y_{i,t-1} - y_{i,t} \geq 0 \tag{4.37}
\]

The variables representing the residual inventory from the last period and the incoming transports are bound to be positive numbers. So it can be concluded that

\[
\sum_{j \in N, \ j \neq i} x_{i,j,t} \geq s_{i,t} - d_{i,t} - y_{i,t} \tag{4.38}
\]

\[
\Leftrightarrow \sum_{j \in N, \ j \neq i} x_{i,j,t} \geq \frac{s_{i,t} - d_{i,t} - y_{i,t}}{k} \tag{4.39}
\]

To find a relation for the number of vehicles, the vehicle capacity constraint 4.17 can be combined with the integrality constraint for the variables \( f_{i,j,t} \). This provides the information that

\[
\sum_{j \in N, \ j \neq i} f_{i,j,t} \geq \left\lceil \sum_{j \in N, \ j \neq i} x_{i,j,t} \right\rceil \tag{4.40}
\]

\[
\Rightarrow \sum_{j \in N, \ j \neq i} f_{i,j,t} \geq \left\lceil \frac{s_{i,t} - d_{i,t} - y_{i,t}}{k} \right\rceil \tag{4.41}
\]
4 Planning under Fixed-Step Transport Costs

If this holds for $\frac{y_{i,t}}{k}$ it is also true for the larger number $y_{i,t}$:

$$\sum_{j \in N, \ j \neq i} f_{i,j,t} \geq \left[ \frac{s_{i,t} - d_{i,t}}{k} - y_{i,t} \right]$$

(4.42)

And in a last step the variable $y_{i,t}$ can be taken out of the ceiling function because it is known to be an integer.

$$\sum_{j \in N, \ j \neq i} f_{i,j,t} \geq \left[ \frac{s_{i,t} - d_{i,t}}{k} \right] - y_{i,t}$$

(4.43)

**Feasibility Condition 3: Provision of Net Demand**

Inequality 4.36 refers to nodes with a net supply. If in contrast a net demand exists, a similar condition can be formulated:

$$y_{i,t-1} + \sum_j f_{j,i,t-m_{j,i}} \geq \left[ \frac{d_{i,t} - s_{i,t}}{k} \right] \quad \forall i, t \text{ with } s_{i,t} < d_{i,t}$$

(4.44)

This means that a net demand must be fulfilled by incoming transports or by the quantity stored in the last period. Similar to the last inequality, a tight lower bound is provided whenever $y_{i,t-1} = 0$, but greater values of the stock variable do not lead to a violation of this inequality. This inequality can be proved in almost the same way as the last inequality 4.36.

**Optimality Condition: Avoid Consecutive Transports**

The conditions developed before are valid for all feasible solutions of the mixed integer program. Further conditions can be found which apply only for the optimal solution of the problem. For this problem, one additional condition has been identified.
4.3 Solution Algorithms

Such a condition can be derived from the fact that no unnecessary vehicles would be scheduled in an optimal solution (provided that it is not free of charge to use the vehicles). Therefore, if there are multiple paths which could be used to route an amount of containers from a node $i$ in a period $t$ to the same destination node and period, there is at most one vehicle on the different paths which is not completely full. Otherwise, the partial load from one of the vehicles could be transported by the other vehicle on the ‘parallel’ connection. The following Figure 4.8 illustrates these thoughts: both the green and the red path connect the same source node ($C_1$) and destination node ($A_4$). So every flow on the first path could also be sent the other way if there were enough capacity left on the vehicles. Optimal flows would therefore never imply the use of more vehicles than necessary for the total amount sent via both paths. Put differently, this means that the container flow can only be transported with the chosen number of vehicles and not with fewer of them. Mathematically, this can be written as in the following inequality (with $w_g$ being the flow on the green connection):

$$w_g + w_r \geq (f_g + f_r - 1)k$$ (4.45)

For the ease of understanding, the indices are replaced by letters indicating the colour (green or red) of the connection in Figure 4.8. The following variables are used here to express the flow from a node $(i, t)$ (corresponding to $C_1$ in the figure)
4 Planning under Fixed-Step Transport Costs

to a node \((j, t + m_{i,j} + 1)\) (corresponding to \(A_4\) in the figure):

\[
\begin{align*}
  f_g &:= f_{i,j,t+1} & f_r &:= f_{i,j,t} \\
  x_g &:= x_{i,j,t+1} & x_r &:= x_{i,j,t} \\
  y_g &:= y_{i,t} & y_r &:= y_{j,t+m_{i,j}}
\end{align*}
\]

(4.46)

(4.47)

(4.48)

If the inequality \[4.45\] above were violated, one could decrease one of the \(w\) variables and increase the other, until one of the \(f\) variables could be reduced. This would save transport costs, which means that the original solution was not optimal. This inequality must hold whenever the flows on either of the paths are not limited by capacity restrictions. And it must still hold when the costs for inventory and transports are different on the two connections, because the two are completely exchangeable, as long as there are no effectual capacity restrictions. So any amount could be sent along the cheaper connection.

This result is independent of the inventory costs on both nodes as long as there are two connections with remaining vehicle capacity on both of them. If the inventory costs are not equal, this is a further reason to reduce the flow of one of the connections and add it to the other.

The problem that remains is to define a valid inequality for the flows which holds for every possible solution. In terms of the existing variables, the flow on each of the connections is

\[
\min(x_g, y_g) \quad \text{and} \quad \min(x_r, y_r)
\]

respectively for the two paths between the two nodes. The inequality \[4.45\] can be rewritten using these terms:

\[
\min(x_g, y_g) + \min(x_r, y_r) \geq (f_g + f_r - 1)k
\]

(4.50)

Note that this constraint is only valid if the transport cost parameter is positive on both connections.

As the minimum operation is not linear, the new constraint cannot be added to a
4.3 Solution Algorithms

linear programming model. But in a branch-and-bound approach, this condition could be used to create a branching rule together with a strong constraint, which is added to the subproblems. For this thesis another approach was chosen. Four possible cases were distinguished in order to create four valid groups of inequalities. In this case, the following inequalities can be added prematurely to the optimisation problem:

1. If \( s_{i,t+1} \leq d_{i,t+1} \) and \( s_{i,t+m_{ij}} \geq d_{i,t+m_{ij}} \):

\[
x_{i,j,t+1} + x_{i,j,t} \geq (f_{i,j,t} + f_{i,j,t+1} - 1)k
- M \sum_{l \in \mathcal{N}, l \neq i} x_{j,l,t+m_{ij}}
- M \sum_{l \in \mathcal{N}, l \neq i} x_{l,i,t+1-m_{ij}}
\]

\[ (4.51) \]

2. If \( s_{i,t+1} \geq d_{i,t+1} \) and \( s_{i,t+m_{ij}} \geq d_{i,t+m_{ij}} \):

\[
y_{i,t} + x_{i,j,t} \geq (f_{i,j,t} + f_{i,j,t+1} - 1)k
- My_{i,t+1} - M \sum_{l \in \mathcal{N}, l \neq \{i,j\}} x_{i,l,t+1}
- M \sum_{l \in \mathcal{N}, l \neq i} x_{l,i,t+1-m_{ij}}
\]

\[ (4.52) \]

3. If \( s_{i,t+1} \leq d_{i,t+1} \) and \( s_{i,t+m_{ij}} \leq d_{i,t+m_{ij}} \):

\[
x_{i,j,t+1} + y_{j,t+m_{ij}} \geq (f_{i,j,t} + f_{i,j,t+1} - 1)k
- M \sum_{l \in \mathcal{N}, l \neq j} x_{j,l,t+m_{ij}}
- M y_{i,t+m_{ij}} - M \sum_{l \in \mathcal{N}, l \neq \{i,j\}} x_{l,j,t+m_{ij}-m_{ij}}
\]

\[ (4.53) \]
4 Planning under Fixed-Step Transport Costs

4. If \( s_{i,t+1} \geq d_{i,t+1} \) and \( s_{i,t+m_{i,j}} \leq d_{i,t+m_{i,j}} \):

\[
y_{i,t} + y_{j,t+m_{i,j}} \geq (f_{i,j,t} + f_{i,j,t+1} - 1)k - M y_{i,t+1} - M \sum_{\{l \in N, \ l \neq (i,j)\}} x_{i,l,t+1} - M \sum_{\{l \in N, \ l \neq (i,j)\}} x_{i,l,t+m_{i,j} - m_{i,j}}
\]

(4.54)

Here ‘\( M \)’ represents a sufficiently large number, so that the constraint is always valid in the cases where it is not certain that the variables on the left take the smaller value in the respective set as claimed in inequality 4.50.

These constraints can simply be proved based on the general inequality 4.50 and on the conservation of flow condition (equation 4.18) of the model with fixed-step costs.

4.3.3 LP-Relaxation and Rounding

A very simple heuristic to find an integer solution is to use the continuous solution attained by optimising the linear programming relaxation, and rounding up all decision variables \( f_{i,j,t} \) to the next integer. This always results in a feasible solution.

Theorem. The optimal solution to the continuous linear problem relaxation with all variable values rounded up is a feasible solution to the integer problem.

Proof. The statement above is correct, if two conditions are met: all \( x_{i,j,t} \) and \( y_{i,t} \) must already be integer in the continuous solution and all \( f_{i,j,t} \) can be rounded up without violating a restriction.

If the number of vehicles is allowed to be continuous, the problem is effectively identical to the basic problem with linear transport costs. In an optimal solution, the number of vehicles would be as low as possible to reduce the transport costs in the objective function. Therefore, the vehicle capacity restriction 4.17 is met with
4.3 Solution Algorithms

equality, so that

\[ x_{i,j,t} = f_{i,j,t} \cdot k \]  

(4.55)

Only the optimal values for the variables \( x_{i,j,t} \) and \( y_{i,t} \) must be found. The \( f_{i,j,t} \) values are determined by the \( x_{i,j,t} \). So, if this problem is mathematically identical to the basic problem, that also means that the integrality property holds and all container quantities are integer in the optimal solution. Only the vehicle quantities \( f_{i,j,t} \) can have non-integer values.

The second condition for the validity of the theorem is that it needs to be possible to round up all vehicle quantities \( f_{i,j,t} \) without violating one of the model constraints. The only restriction containing this type of variable is the inequality \( 4.17 \) which gives only a lower bound and the variable bounds defined in condition \( 4.22 \). In the formulation given above in the model (Figure 4.5), no upper bound exists that could be violated. In principle one could be defined, but it would be redundant as the same limit can be expressed by the existing upper bounds for the transport quantities given in condition \( 4.21 \). If, nevertheless, an upper limit for the amount of vehicles on some connections should be defined directly, one would probably choose an integer number as limit. So, if a continuous solution results in a fractional number of vehicles, this number would always be lower than the limit. Rounding up would not violate the limit in such a situation but increase the value at most to exactly the limiting integer. The fractional result would not be equal to the limit as it would then already be an integer. It also could not exceed the limit, as the condition must be fulfilled for the relaxed solution, and a number of vehicles greater than its predefined maximum would result in an infeasible relaxation solution.

So, it can be seen that this simple rounding heuristic always provides a feasible solution to the problem with fixed-step transport costs. Unfortunately, this is only true if the decision variables are rounded up. If one tried to round up or down to the next integers respectively, the solution would often not be feasible. Rounding down the number of vehicles unavoidably results in a violation of the vehicle capacity restriction, because the number of containers on the same connection would not be adjusted so that they would not fit into the vehicles.
4 Planning under Fixed-Step Transport Costs

The solutions found by this simple heuristic are not very good, because in the optimisation process, the transport costs are regarded as proportional to the number of containers, while the objective value found by rounding up is much higher. Therefore the costs of the optimal solution are overestimated. The main advantage of this method is, that the lower bound for the optimal solution derived by the relaxation can very quickly be transformed to an upper bound, just by rounding up some variables and without a separate optimisation process. So it is very useful in the context of a branch-and-bound algorithm.

4.3.4 A Branch-and-Bound Algorithm

Outline

A common approach for solving (mixed) integer optimisation problems is the branch-and-bound algorithm introduced by Land and Doig (1960) and Dakin (1965). In this section, a specialised modification of this method that allows solving the problem with fixed-step transport costs faster than a standard implementation is proposed. This can be achieved by adding some additional constraints to the problem in order to improve the lower bounds on the one hand, and using a heuristic to improve the upper bounds on the other.

The main idea of a branch-and-bound algorithm is to branch the problem into several subproblems and to compare the upper and lower bounds for their objective function value. This way subproblems that can be proved not to lead to the optimal solution for the initial problem can be excluded. Others have to be examined further and are therefore branched into smaller subproblems. This procedure can be continued until an acceptable solution is found, or otherwise terminated when it is certain that a better solution would not be found by further examination of the remaining subproblems.

The general structure of the optimisation process for a minimisation problem is listed in Figure 4.9. At the beginning of the process a candidate list is defined, which holds all subproblems that must be examined. It is initialised with only
4.3 Solution Algorithms

Initialisation

\[ C := \{ \text{Basic Problem} \} \]

upperBound := \infty

bestSolution := None

Optimisation

\[ \text{while } |C| > 0 \text{ do} \]

Choose one \( p \in C \) \hspace{1cm} \text{Candidate selection (1)}

Delete \( p \) from \( C \)

Optimise a relaxation of \( p \) \hspace{1cm} \text{Relaxation (2)}

\[ \text{if } p \text{ has no feasible relaxation solution then} \]

\[ \text{Quit this iteration (and drop candidate } p \text{)} \]

\[ s := \text{Relaxed solution of } p \]

\[ z := \text{Objective value of } s \]

\[ \text{if } z \geq \text{upperBound then} \]

\[ \text{Quit this iteration (and drop candidate } p \text{)} \]

\[ \text{else if } s \text{ is integral then} \]

\[ \text{bestSolution := } s \]

\[ \text{upperBound := } z \]

\[ \text{Possibly:} \]

\[ \text{Remove all candidates } p \in C \text{ with lower bound } \geq \text{upperBound} \]

\[ \text{else} \]

\[ \text{Possibly:} \]

\[ \text{Use a heuristic to find a new best solution within } p \]

\[ \text{Branch } p \text{ into subproblems and insert them into } C \]

\[ \text{Branching rule (5)} \]

end

end

Figure 4.9: The branch-and-bound algorithm.
the basic problem as member. The variable ‘upperBound’ represents the objective value for the best feasible solution found. Starting without an initial solution for a minimisation problem, this variable is set to infinity at the beginning. The best solution itself is stored in the variable ‘bestSolution’.

Each iteration of the algorithm starts with the selection of one subproblem from the list of candidates (line 7 in Figure 4.9). Then, the relaxed solution of this problem is calculated. This is the solution of a similar problem where some constraints, for example the integrality constraints, are removed or weakened. If the relaxation has no feasible solution, then neither can the original (non-relaxed) problem have a feasible solution. So this subproblem would also not lead to the global optimal solution and could be removed from the list of candidates (line 11). If the relaxation has a feasible solution, it is compared to the best integer solution found until then (line 15). If the relaxation’s objective value is greater than the minimal objective value from the best solution, the integer solution of the subproblem would also not have a feasible solution. Alternatively, it could happen that the relaxation has an integer solution or, to express it more generally: it has a solution that is also feasible to the original problem (line 17). In this case, an optimal solution for the subproblem is found. This condition is only tested if the solution is known to have a better objective value than the best solution found until then. So, the solution found can be stored as the new best solution and its objective value becomes the new global upper bound. It is possible to use the new upper bound to clear the list of candidates and remove all those which have a lower bound exceeding the new upper bound as indicated in line 21. The third case (line 22) is, that the subproblem is neither infeasible nor does its relaxation lead to an integer solution. In this situation, the subproblem is branched into even smaller subproblems that are added to the list of candidates and examined later (line 25). At this point an additional step could accelerate the optimisation process. A heuristic can be used to find a feasible solution for the given subproblem. This can result in a new best solution, and thus a new upper bound (line 24).

The algorithm described so far is the general form of a branch-and-bound minimisation procedure. In the listing of the algorithm in Figure 4.9 five steps are marked in the right margin. At these steps adjustments can be made to develop a specialised
approach for a particular optimisation problem. In the following sections, some improvements, which make it possible to optimise the problem with fixed-step transport costs much faster, are proposed.

**Candidate Selection**

At the beginning of each iteration, a subproblem must be chosen from the list of candidates (line 7 in Figure 4.9). The selected problem would then be analysed and possibly branched. As a result, the list of candidates has the form of a tree. Starting with the original problem as the root, it is branched into more and more detailed subproblems, until the last subproblem in a branch is shown to be infeasible, or until its integer solution is found by the relaxation.

Usually, the subproblem selection is made by simple rules. Three of these have been tested for the problem under consideration. They are breadth-first search, depth-first search and best-first search.

Breadth-first means that the problems are selected by the level of branching. All problems at a given level are examined before the algorithm goes deeper into the newly generated branches. So the search tree is developed in breadth. This can be implemented simply by placing the new subproblems in a queue from which they are extracted with the first-in, first-out method. New subproblems at deeper levels are added to the end of the queue. The first in the queue, which is extracted at the beginning of an iteration is then the one on the highest level. This search strategy is not viable for this problem, because it needs a very high amount of memory during the calculation. Therefore it is not possible to apply the breadth-first search to large networks.

The opposite is the depth-first search, which means that every branch is followed into its depth until it ends with either a feasible (integral) or an infeasible solution, even for the relaxation. So, the search tree is developed branch by branch until the last leaf. The tree looks ‘imbalanced’ during the procedure because some subproblems are split into very detailed branches, while at the same time other high-level subproblems have not yet been examined. From a programming perspective,
this behaviour can be achieved with a candidate set that is stored in a stack data structure with the last-in, first-out method. So at each new iteration the most detailed subproblems stored last are extracted and examined first.

The third strategy tested is a best-first search. This approach requires that a rule exists which allows the comparison of the stored candidates. Because the relaxation solution has to be calculated for each candidate anyway, its objective value can already be calculated when new subproblems are created. So, this objective value can be stored in the candidate list and the one with the smallest value is chosen as the next problem to consider at the beginning of each iteration.

Relaxation Solution

The purpose of a relaxation solution is to find an objective value that is guaranteed to be at least as good as the optimal solution of the integer problem (line 9 in Figure 4.9). So all integer solutions with an objective value worse than the lower bound are proved not to be optimal, when they are compared to the objective value of the relaxation. Therefore, a relaxation is better in terms of its usefulness for a branch-and-bound algorithm, if it provides a very tight bound. It should be as close as possible to the optimal objective value, so that many feasible but non-optimal solutions can be excluded.

Several approaches to calculate relaxation solutions for an integer problem are discussed in the existing literature. For example Ahuja, Magnanti and Orlin (1993, Chapter 16) suggest a lagrangian relaxation especially for network flow problems. The method used here is another well-established one: the linear programming relaxation (see for example Nemhauser and Wolsey 1988, p. 355 ff.). It is determined by removing the integrality constraints from an integer program so that it is turned into a continuous linear program that can be optimised much faster.

Here, the linear programming relaxation has two advantages. First, it can be calculated by specialised network algorithms as the relaxed problem can be solved as a standard s-t-flow problem; for example with the successive shortest path algorithm or with the network simplex algorithm. The second advantage is, that
4.3 Solution Algorithms

from the linear programming relaxation a feasible integer solution is immediately available (as explained above). All non-integral values need only to be rounded up in order to get an integer solution and probably a better upper bound.

Unfortunately, in its simple form this method provides only very loose bounds to the optimal objective value. However, this is only the case if it is applied to the original problem. If additional inequalities (see Section 4.3.2) are prematurely added, the lower bounds are much tighter and many non-integral solutions can be excluded.

Candidate List Cleaning

As mentioned above, whenever a new best solution is found in the branch-and-bound algorithm it can be used to shorten the list of candidates (line 20 in Figure 4.9). Normally, this is simply done by comparing the new upper bound provided by the new best solution to the lower bound that has to be calculated for each candidate.

So, this step is a trade-off between having to calculate the lower bounds earlier, or maintaining a longer list of candidates and thus possibly needing more effort to make the candidate selection. With the simple selection rules used in this algorithm it would not lead to a faster result. The same comparison needed here is also made whenever a subproblem is picked from the list of candidates. But possibly in the meantime an even better best solution can be found, so that a candidate can be eliminated more probably. Therefore removing candidates that violate the criterion is postponed to that time.

Nevertheless introducing the clean-up step could save memory. A shorter list of candidates would need less space.

Heuristic for Upper Bound

When no feasible solution is found for a subproblem with the relaxation algorithm, a heuristic can be used to find a new best solution (line 23 in Figure 4.9). This can drastically reduce the time necessary to find a good solution in a branch-and-bound
algorithm. If good feasible solutions are found early in the optimisation process, many branching iterations can be avoided because the solutions provide an upper bound that helps to screen the candidates in order to decide if they have to be analysed at all.

Therefore, several heuristics were tested for the problem under consideration but none of these could compare to simply using the linear programming relaxation and rounding up all non-integer variables. As shown in Section 4.3.3 this necessarily leads to a feasible solution to the integer optimisation problem. The main advantage of this method is, that the relaxation solution is already calculated before a heuristic solution is needed. So no additional optimisation process is needed here.

**Branching Rule**

A key element of the branch-and-bound algorithm is the branching rule (line 25 in Figure 4.9). It is the rule used to branch a problem into two or more subproblems. For integer programming, the usual approach is to choose a variable \( z \) which takes a non-integral value \( z^* \) in the relaxation solution and to exclude this result from the set of feasible solutions for the relaxation. If the variable must be an integer, the following condition can be used to create two subproblems and add a variable bound to each of them:

\[
[z^*_i] \leq z_i \leq [z^*_i] + 1
\]  \hspace{1cm} (4.56)

Often, more than one variable violates the integrality constraint in the relaxation solution. Then, a criterion is needed to choose the variable for branching. Simple criteria are to choose the variable with the smallest or largest fractional part. More sophisticated techniques are available, for instance based on the influence of the variable on the objective (Dakin 1965, p. 253). The goal is to use a branching rule which leads to a fast convergence of the upper and lower bounds, so that only a small number of subproblems needs to be evaluated. This can be achieved by choosing the fractional variable, which is expected to change the objective value most, if forced to be integer. Often, simple criteria are chosen, which only
heuristically estimate the impact on the objective value but can be applied more quickly (Linderoth and Savelsbergh 1999, p. 174).

For the container transport model, branching means choosing a non-integral vehicle amount \( f(i, j, t)^* \) from the relaxation solution and creating two subproblems with the following restrictions:

\[
\begin{align*}
  f(i, j, t) &\leq \lfloor f(i, j, t)^* \rfloor \\
  f(i, j, t) &\geq \lceil f(i, j, t)^* \rceil 
\end{align*}
\]  

(4.57)

The following selection criteria have been tried:

- Most fractional variable
- Transport with best utilisation
- Transport with worst utilisation
- Most expensive transport
- Transport with maximum cost change

The first three of these criteria aim at the utilisation of the transport vehicles. It is a classic generic rule to branch upon the most fractional variable, this means that the variable with the fractional part nearest to 0.5 would be chosen. This is equivalent to a half-full vehicle. The idea is, that a change in this variable value would result in a relatively large change to the total costs, so that the lower bound can be increased. Alternatively, the transport with the lowest or highest utilisation can be selected.

The remaining two branching criteria are more problem specific. Branching upon the most expensive transport in this context means selecting the fractional variable which is assigned the highest per-vehicle transportation costs \( c_{\text{transport}}(i, j) \). This is equivalent to choosing the variable with the largest objective function value.

The last rule selects the transport with the maximum cost change. This means that for all fractional variables the minimal change to the costs is determined which the
4 Planning under Fixed-Step Transport Costs

rounding of this variable would induce. This is the minimum of:

\[(f(i, j, t)^* - \lfloor f(i, j, t)^* \rfloor) \cdot c_{i,j}^{\text{transport}} \quad \text{and} \quad \lfloor f(i, j, t)^* \rfloor - f(i, j, t)^* \cdot c_{i,j}^{\text{transport}}\]

The variable which in the current solution yields the highest value according to this formula is chosen for branching. This value expresses only the direct impact of a variable on the objective function value, not the total change. If the flow of containers is limited on a connection by the branching process, it needs to be routed via a different connection where additional costs can arise. Nevertheless, for this particular problem it will often be the case that the flow is added to another vehicle which has a remaining capacity, so that its utilisation will increase without additional costs. Then this branching rule would determine the exact total change in costs.

The choice of the branching technique has a strong influence on the calculation time of the algorithm. The best results can be attained with the last two rules of choosing either the most expensive transport or the one with the maximum cost change for branching. This is demonstrated on numerical examples below in Section 4.4.

4.3.5 A Branch-and-Cut Algorithm

In the last section, the branch-and-bound algorithm was introduced as a method to reliably find the optimum solution. The performance of this algorithm can be improved further, if it is combined with cutting plane methods. The result is called a branch-and-cut algorithm then (for an introduction into this method see Wolsey 1998, pp. 157 ff.). This approach was also tested within the scope of this thesis. Its structure is shown in Figure 4.10.

This method is an extension of the branch-and-bound algorithm; in principle only one step is added (lines 23–26). Once a non-integer solution is found for a subproblem with the relaxation algorithm, an attempt is made to add inequalities (‘cuts’) to the optimisation problem, which cut off the non-integer solution from the set of feasible solutions for the relaxed problem. The subproblem is then reoptimised in order to find a new relaxation solution. If it is non-integral once again, the steps
4.3 Solution Algorithms

Initialisation

\[ C := \{ \text{Basic Problem} \} \]
\[ \text{upperBound} := \infty \]
\[ \text{bestSolution} := \text{None} \]

Optimisation

while \(|C| > 0\) do

Choose one \(p \in C\)
Delete \(p\) from \(C\)
Optimise a relaxation of \(p\)
if \(p\) has no feasible relaxation solution then
Quit this iteration (and drop candidate \(p\))
s := Relaxed solution of \(p\)
z := Objective value of \(s\)
if \(z \geq \text{upperBound}\) then
Quit this iteration (and drop candidate \(p\))
else if \(s\) is integral then
\[ \text{bestSolution} := s \]
\[ \text{upperBound} := z \]
Possibly:
Remove all candidates \(p \in C\) with lower bound \(\geq \text{upperBound}\)
else
if any known cutting planes are violated in solution \(s\) then
Add the cutting planes to the current subproblem \(p\)
Repeat from line 9
Possibly:
Use a heuristic to find a new best solution within \(p\)
Branch \(p\) into subproblems and insert them into \(C\)
end
end
end

Figure 4.10: The branch-and-cut algorithm.
are repeated as long as possible cuts can be identified. This additional loop helps to either find an integral solution before branching any further, or at least to provide a tighter lower bound than without the additional inequalities.

The branch-and-cut method relies on classes of inequalities which are guaranteed to be fulfilled by any feasible solution of the mixed integer problem, but are violated by some solutions of the continuous relaxation of the problem. These inequalities are stored in a ‘cut pool’. Whenever a non-integer solution is evaluated, inequalities are chosen from this cut pool that are violated by the current solution. Therefore it is a flexible method that allows adding or removing inequalities without changing the algorithmic structure.

Only standard cuts were tried for the container transport model. The specialised inequalities shown in Section 4.3.2 were applied prematurely, i.e. already on the main problem. So they were not used to generate cutting planes in the branch-and-cut algorithm. For this, only the following cuts were tested used:

- Mixed integer rounding (MIR) cuts (see Nemhauser and Wolsey 1988, pp. 244–246)
- Gomory’s mixed integer cuts (see Nemhauser and Wolsey 1988, pp. 227–229)

These are sets of inequalities which can be used for general (mixed) integer optimisation programs (for a more detailed explanation of these rules see Wolsey 1998, Chapter 8). For the container transport planning problem they showed good results.

### 4.4 Comparison of the Algorithms and Numerical Tests

Throughout the previous section, a branch-and-bound and a branch-and-cut approach were developed to solve the mixed integer problem induced by the fixed-step transport costs. As explained above, the algorithms can be tuned by adding or changing several steps. The experimental results presented in this section show the
4.4 Comparison of the Algorithms and Numerical Tests

effect of different variations for the problem with fixed-step transport costs. The aim of this section is to show how the solution methods discussed before perform in terms of speed and solution quality. This way it should be possible to identify the best approach for solving this class of problems.

4.4.1 Test Setup

For the tests, the algorithms were implemented with C++ based on the GNU Linear Programming Kit library (http://www.gnu.org/software/glpk/). It provides an interface for the standard simplex methods, as well as a framework for mixed integer programming. This setup has shown to calculate faster compared to a custom implementation of the branch-and-bound algorithm from scratch.

The system used for testing was not very powerful. It was equipped with 4 GB of memory and an Intel Celeron G1610 CPU with 2.60 GHz clock rate. Therefore rather small test examples were used. The optimisation process is expected to be much faster either using a more rapid processor, or if the implementation employs several CPU cores for parallel calculations. However, the main focus of the following tests is to analyse the performance of the optimisation methods in order to identify the best one. This result is independent of problem size and execution speed.

<table>
<thead>
<tr>
<th>Network</th>
<th>Periods (T)</th>
<th>Nodes (N)</th>
<th>Vehicle size (k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>3</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 4.4: General parameters for the example networks.

Four different container systems were used as benchmark problems, one of each of the types outlined in Section 1.2.1. The dimensions of these randomly generated problems were chosen in a manner to obtain comparable calculation times. The system had three to five network nodes, and a planning horizon ranging from six to twelve periods needed to be optimised. The general parameters of the benchmark
4 Planning under Fixed-Step Transport Costs

problems are listed in Table 4.4. The detailed data is summarised in Appendix A. Also some supplementary results are shown there.

The calculation time was limited to 10 minutes for each of the optimisation runs. The time needed to find the optimal solution as well as the time until the algorithm terminated with the proof of optimality were measured. If a calculation was stopped before it was finished, only the elapsed time of 600 seconds was considered.

Tables 4.5 and 4.6 show the different branch-and-bound strategies as well as additional improvements which were evaluated.

<table>
<thead>
<tr>
<th>Candidate selection</th>
<th>Branching rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth-first</td>
<td>Most fractional variable</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most expensive transport</td>
</tr>
<tr>
<td></td>
<td>Transport with best utilisation</td>
</tr>
<tr>
<td></td>
<td>Transport with worst utilisation</td>
</tr>
<tr>
<td></td>
<td>Maximum cost change</td>
</tr>
</tbody>
</table>

Table 4.5: Branching and candidate selection strategies of the branch-and-bound algorithm.

<table>
<thead>
<tr>
<th>Additional constraints</th>
<th>Heuristic</th>
<th>Cuts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasibility 1</td>
<td>No</td>
<td>Mixed integer rounding</td>
</tr>
<tr>
<td>Feasibility 2</td>
<td>Yes</td>
<td>Gomory’s</td>
</tr>
<tr>
<td>Feasibility 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimality</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6: Optional improvements of the branch-and-bound algorithm.

4.4.2 Candidate Selection and Branching Rules

First, a basic branch-and-bound without heuristics or additional constraints was applied to the data. So, only rules for the candidate selection and for the branching had to be chosen in order to define a search strategy. The experiment helps to determine the best choice.
4.4 Comparison of the Algorithms and Numerical Tests

All combinations of the rules listed in Table 4.5 were tested. For the candidate selection only a classic 'depth-first' search and a 'best-first' search were taken into consideration. Best-first means, that the subproblem with the minimal relaxation solution is examined next. For the branching rules, more options were tested. First, the most fractional variable was chosen for branching. The next rule evaluated was the 'most expensive transport', which means to choose the variable with the maximum objective function coefficient. The remaining rules focused on the last vehicle on a connection, so only vehicles which are only partially loaded. It was tested to branch upon the connection with the best or worst utilised vehicle. The last rule applied, is to use the variable for branching which causes the maximum expected change to the objective value if it is forced to be integral. These rules are explained in more detail in Section 4.3.4.

Table 4.7 lists the results of this test. For each of the combinations of the candidate selection and branching rules, the average time to finding the optimal solution as well as to proving its optimality were determined. The algorithm did not finish within ten minutes with all of the search strategies. This can be seen in the last column which shows the percentage of problems that were completely solved.

<table>
<thead>
<tr>
<th>Candidate selection</th>
<th>Branching rule</th>
<th>Calculation time (s)</th>
<th>Examples solved to optimality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Optimal solution</td>
<td>Proof of optimality</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Most expensive</td>
<td>53</td>
<td>88</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Max. cost change</td>
<td>120</td>
<td>143</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most expensive</td>
<td>205</td>
<td>205</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Best utilisation</td>
<td>95</td>
<td>207</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Most fractional</td>
<td>148</td>
<td>216</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Worst utilisation</td>
<td>217</td>
<td>261</td>
</tr>
<tr>
<td>Best-first</td>
<td>Max. cost change</td>
<td>298</td>
<td>311</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most fractional</td>
<td>205</td>
<td>320</td>
</tr>
<tr>
<td>Best-first</td>
<td>Best utilisation</td>
<td>305</td>
<td>334</td>
</tr>
<tr>
<td>Best-first</td>
<td>Worst utilisation</td>
<td>228</td>
<td>363</td>
</tr>
</tbody>
</table>

Table 4.7: Results of the test of the candidate selection and branching rules.
4 Planning under Fixed-Step Transport Costs

The strategies are ordered ascendingly by the total calculation time until the proof of optimality. It turned out, that the best-first search often did not terminate in time, and it was generally slower than the depth-first search. The depth-first search with the ‘most expensive’ or ‘maximum cost change’ branching rules performed best. It is not surprising that these branching rules succeeded, as they aim at the costs which directly influence the objective value. It is a general recommendation to branch upon the fractional variable which leads to the largest change in the objective value when it is constrained to be an integer (Linderoth and Savelsbergh 1999, p. 174). Although the two best rules used here only heuristically determine this variable, this is enough to perform better than if a completely different pattern is followed.

4.4.3 Effect of the Additional Contraints

In a next experiment it was to be examined, how the additional bounds explained in Section 4.3.2 can be used to speed up the calculations. These conditions were added prematurely to the optimisation model and then the branch-and-bound algorithm was used on it. The total number of constraints which were added to the model formulation for the example networks is listed in Table 4.8. It can be seen, that the conditions lead to different numbers of additional constraints. The most are added by the optimality condition.

<table>
<thead>
<tr>
<th>Model</th>
<th>Network 1</th>
<th>Network 2</th>
<th>Network 3</th>
<th>Network 4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Feasibility 1</td>
<td>9</td>
<td>20</td>
<td>37</td>
<td>1</td>
<td>16.75</td>
</tr>
<tr>
<td>Feasibility 2</td>
<td>44</td>
<td>26</td>
<td>15</td>
<td>14</td>
<td>24.75</td>
</tr>
<tr>
<td>Feasibility 3</td>
<td>10</td>
<td>20</td>
<td>41</td>
<td>2</td>
<td>18.25</td>
</tr>
<tr>
<td>Optimality</td>
<td>68</td>
<td>150</td>
<td>80</td>
<td>32</td>
<td>59.75</td>
</tr>
<tr>
<td>Feasibility 1–3</td>
<td>63</td>
<td>66</td>
<td>93</td>
<td>17</td>
<td>59.75</td>
</tr>
<tr>
<td>All</td>
<td>131</td>
<td>190</td>
<td>216</td>
<td>49</td>
<td>142.25</td>
</tr>
</tbody>
</table>

Table 4.8: Number of constraints added to the optimisation model.

Figure 4.11 shows the average calculation time needed until the optimal solution was found and until the optimality was proved. The test did not produce a clear
result. For network 2, the first feasibility condition slowed down the calculation. The other feasibility conditions do not seem to have a strong influence. This might lead to the conclusion that the first feasibility condition has a negative impact on the runtime. However, during all tests performed for this thesis, the feasibility conditions turned out to be very beneficial in general. They shorten the optimisation process by cutting off regions of the search space for the relaxation algorithm, that are infeasible for the mixed integer problem.
4 Planning under Fixed-Step Transport Costs

Depending on the data, the optimality condition can be inefficient if it is added at the beginning of the branch-and-bound procedure. The additional constraints slow down the simplex optimisations and are sometimes too weak to speed up the convergence of the branch-and-bound. In addition, the ‘big M’ method used for these constraints might lead to noticeable rounding errors in some cases.

4.4.4 Effect of the Rounding Heuristic

As a next step, it is now demonstrated how a simple rounding heuristic as explained in Section 4.3.3 helps to improve the solution algorithm. The solution speed of the default branch-and-bound is compared to one which rounds all relaxation solutions up to obtain an integer solution. Figure 4.12 shows the time needed to find the optimal solution of the four examples and to prove the optimality.

![Figure 4.12: Calculation time with and without the rounding heuristic (with depth-first search and branching on the most expensive transport).](image)

The program cannot be proved to be faster at finding the optimum in this setting with the heuristic. On the other hand, the heuristic also had no negative impact on
the time needed. Whether with or without the rounding, the branch-and-bound found the optimal solution in a comparable time. The main advantage of the heuristic is that a feasible integer solution is available right from the beginning.

The heuristic neither helps to prove the optimality faster, nor to find the optimal solution faster. But it is very useful when solving large problems, because with the heuristic feasible solutions are already available very early. One should consider using it, if finding the optimal solution is not the only aim, but also providing a solution within a limited time frame.

4.4.5 Effect of the Cutting Techniques

In the previous sections, variations of the branch-and-bound algorithm were examined. On top of this, it is possible to integrate cutting techniques, which results in a branch-and-cut algorithm as outlined in Section 4.3.5.

Two standard cutting techniques (abbreviated as ‘cuts’) introduced in Section 4.3.5 were tested: ‘Mixed integer rounding’ cuts and ‘Gomory’s’ cuts.

With both of these techniques the branch-and-cut has shown to be much faster in all examples calculated for this thesis compared to a branch-and-bound algorithm. Table 4.9 on the following page shows, that this also holds for the benchmark problems. The calculation time is strongly reduced although only standard cutting techniques are used.

Therefore it can clearly be regarded as a general recommendation to solve this type of model with a branch-and-cut approach.

4.4.6 Summary and Recommendations

In this section the alternative settings for the branch-and-bound and branch-and-cut algorithms were analysed. It was shown that a depth-first strategy is best, branching upon the transport connection with either the highest costs-per-item or with the highest impact on the objective when the quantity is rounded. The additional
4 Planning under Fixed-Step Transport Costs

<table>
<thead>
<tr>
<th>Cuts</th>
<th>Network example</th>
<th>Calculation time (s)</th>
<th>Optimality gap after 10 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Network 1</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>None</td>
<td>Network 2</td>
<td>120</td>
<td>137</td>
</tr>
<tr>
<td>None</td>
<td>Network 3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>None</td>
<td>Network 4</td>
<td>85</td>
<td>195</td>
</tr>
<tr>
<td>Mixed integer rounding</td>
<td>Network 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mixed integer rounding</td>
<td>Network 2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mixed integer rounding</td>
<td>Network 3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mixed integer rounding</td>
<td>Network 4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Gomory's</td>
<td>Network 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Gomory's</td>
<td>Network 2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Gomory's</td>
<td>Network 3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Gomory's</td>
<td>Network 4</td>
<td>40</td>
<td>105</td>
</tr>
</tbody>
</table>

Table 4.9: Results of the test of the cutting rules (with depth-first search and branching on the most expensive transport).

feasibility conditions help to reduce the calculation time a bit further; this effect could not be shown clearly with the used examples, however. The also applied optimality condition instead extended the total time. It should not be used to preliminarily add constraints to the optimisation model.

To round all fractional values up, immediately provides a feasible solution to the mixed-integer program. At the same time this heuristic has no influence on the time needed to solve the problem. It cannot shorten the calculation because no tight lower bound is available for this problem. So, the upper bound provided by the heuristic cannot exclude many subproblems as the gap between the two bounds is too large. The heuristic should therefore be used if it is useful to have a feasible solution quickly. Without the heuristic, it is possible that it takes very long for large problems until the first feasible solution is obtained.

On top of these results, the branch-and-cut strategy was tested. It was shown that the algorithm terminates much faster when cutting techniques are used.
In general, one could therefore recommend to use a branch-and-cut approach and also to integrate the heuristic if it is possible that a solution is needed before the algorithm finishes. One could also consider to integrate the feasibility conditions as additional model constraints. The optimality condition should in contrast to be used this way as this slows down the calculation. Here it might be worthwhile to invest further research. Probably it is possible to use the optimality conditions and also the considerations which lead to the feasibility conditions to develop new problem specific cutting techniques. The cuts could dynamically be added to the branch-and-cut instead of including additional constraints. This could lead to a much faster optimisation process.
5 Summary and Outlook

The aim of this thesis was to provide mathematical models for planning transports and capacity adaptations in a container system. The main characteristics of this type of system and the planning tasks were analysed in Section 1.2. This helped to render the objective more precisely. The connection of capacity with transport planning makes most sense within a short planning horizon, when enough detailed data is available to schedule transports. Therefore, this is what the research focused on.

In addition, a key characteristic of container transports was identified: containers can often be moved in larger quantities on the same vehicle, so that transport costs should not be regarded as linear but increasing stepwise with the number of vehicles employed. Hence, the costs represent a fixed-step function of the quantity.

It was shown that extensive literature is available concerning related topics. Mathematical models already exist, which allow for some of the facets of transport and capacity planning that are important for reusable containers. However, these approaches never cover all relevant aspects. Some are missing completely; among them the challenge to model fixed-step transportation costs.

To get a more flexible, yet mathematically tractable model, a basic model was developed. This is very simple to solve as it can be reduced to the well investigated minimum cost flow problem and it can also be extended flexibly. In Chapter 3 model add-ons were presented to include the aspects of lost orders, delayed deliveries and take-along transports as well as two capacity measures, namely purchase and disposal of containers as well as leasing. All of these extensions still preserve the favourable characteristics of the basic model in terms of complexity.

To consider fixed-step transport costs, another model was proposed that is used instead of the basic model and can also be extended by the add-ons. It requires
more computational effort to solve this optimisation problem, as the minimum cost flow problem then changes the cost structure and becomes non-linear. Therefore, the model was reformulated to achieve a mixed integer program. This can be solved by a tailored branch-and-cut approach, which combines the positive characteristics of a heuristic and an exact solution algorithm: it provides a reasonable and feasible solution quickly and is guaranteed to discover the optimum solution in the end.

Certain aspects have been left open. From the perspective of planning container systems, this thesis focused on only one type of container. If substitute types exist, multicommodity flows should be regarded. Also other interesting points have been treated in related research areas, for example deliveries within time windows or network nodes with a limited handling capacity could be worth analysing.

From the view of operations research, intriguing questions arise. It was surprising that no research seemed to be available for a minimum cost flow problem with stepwise costs. Such a model was developed here, and the discovered feasibility and optimality conditions as well as the solution approach can be generalised. Nevertheless, there is need for further research. The branch-and-cut algorithm still has enormous potential. Based on optimality conditions such as the one presented in Section 4.3.2 as well as on existing ideas for the fixed charge network flow problem or other integer network problems, more sophisticated branching rules, cutting planes and heuristics can be identified that would allow to apply the model to much larger use cases.
Appendix A

Data and Results of the Calculations in Section 4.4

This appendix shows supplementary data and results for the calculations in Section 4.4. The example problems used for the calculations are listed in Section A.1. The corresponding optimal solutions are shown in Section A.2. In the remaining sections, the results of the benchmark calculations can be found in more detail than in the text of the thesis.

A.1 Data

This section shows the example networks used for the benchmark calculations. These represent the four different types of container systems outlined in Section 1.2.1.

A.1.1 Network 1: Distribution Network

The first example corresponds to the container system in a distribution network. This means that the goods are routed from a central location to the remaining sites. As the container flows move in the opposite direction, the containers are released at the satellite nodes while the central node shows an excess demand of containers. The example covers 11 periods and 5 nodes.
Appendix A  Data and Results of the Calculations in Section 4.4

Distances between network nodes:

\[
(m_{i,j}) = \begin{pmatrix}
1000 & 1 & 2 & 3 & 1 \\
1 & 1000 & 1000 & 1000 & 1000 \\
2 & 1000 & 1000 & 1000 & 1000 \\
3 & 1000 & 1000 & 1000 & 1000 \\
1 & 1000 & 1000 & 1000 & 1000 \\
\end{pmatrix}
\]

Supply at the network nodes:

\[
(s_{i,t}) = \begin{pmatrix}
53 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 5 & 8 & 4 & 6 & 5 & 2 & 4 & 3 & 3 \\
3 & 6 & 9 & 7 & 8 & 4 & 6 & 3 & 2 & 2 \\
4 & 4 & 8 & 8 & 5 & 7 & 6 & 5 & 3 & 1 \\
7 & 5 & 6 & 9 & 4 & 3 & 4 & 4 & 0 & 1 \\
\end{pmatrix}
\]

Demand at the network nodes:

\[
(d_{i,t}) = \begin{pmatrix}
28 & 23 & 20 & 16 & 19 & 9 & 7 & 10 & 11 & 20 & 18 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

Storage costs:

\[
(c_{i}^{\text{stock}}) = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]
A.1 Data

Per-vehicle transport costs:

\[
(c_{t,j}^{\text{transport}}) = \begin{pmatrix}
1000 & 1 & 2 & 3 & 1 \\
1 & 1000 & 1000 & 1000 & 1000 \\
2 & 1000 & 1000 & 1000 & 1000 \\
3 & 1000 & 1000 & 1000 & 1000 \\
1 & 1000 & 1000 & 1000 & 1000 \\
\end{pmatrix}
\]

A.1.2 Network 2: Hub-and-Spoke Network

The second network follows a hub-and-spoke structure. At some of the 5 locations empty containers are needed, other locations constantly provide a supply. Both types of nodes are only connected via a central transshipment node. 12 periods were to be analysed.

Distances between network nodes:

\[
(m_{i,j}) = \begin{pmatrix}
1000 & 1 & 2 & 3 & 1 \\
1 & 1000 & 1000 & 1000 & 1000 \\
2 & 1000 & 1000 & 1000 & 1000 \\
3 & 1000 & 1000 & 1000 & 1000 \\
1 & 1000 & 1000 & 1000 & 1000 \\
\end{pmatrix}
\]

Supply at the network nodes:

\[
(s_{i,t}) = \begin{pmatrix}
18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 5 & 7 & 8 & 6 & 4 & 5 & 6 & 3 & 4 & 2 & 4 \\
6 & 4 & 5 & 7 & 7 & 6 & 2 & 4 & 1 & 5 & 0 & 5 \\
21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]
Appendix A Data and Results of the Calculations in Section 4.4

Demand at the network nodes:

\[
(d_{i,t}) = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 7 & 8 & 5 & 4 & 5 & 2 & 4 & 2 & 0 & 1 & 2 \\
4 & 5 & 6 & 5 & 7 & 3 & 7 & 5 & 3 & 1 & 0 & 4
\end{pmatrix}
\]

Storage costs:

\[
(c_{i}^{stock}) = \begin{pmatrix}
0 & 1 & 1 & 1 & 1
\end{pmatrix}
\]

Per-vehicle transport costs:

\[
(c_{i,j}^{transport}) = \begin{pmatrix}
5000 & 5 & 10 & 15 & 5 \\
5 & 5000 & 5000 & 5000 & 5000 \\
10 & 5000 & 5000 & 5000 & 5000 \\
15 & 5000 & 5000 & 5000 & 5000 \\
5 & 5000 & 5000 & 5000 & 5000
\end{pmatrix}
\]

A.1.3 Network 3: Procurement Network

Network 3 is the equivalent to a procurement system as it is needed for instance in the automotive industry. One central node periodically supplies the system with empty containers. This would be the production plant in an automotive company. The other nodes (which would be the suppliers of the plant) show a demand of empties. The optimal flows for a planning horizon of 12 periods were to be determined.
A.1 Data

Distances between network nodes:

\[
(m_{i,j}) = \begin{pmatrix}
1000 & 1 & 2 & 3 & 1 \\
1 & 1000 & 1000 & 1000 & 1000 \\
2 & 1000 & 1000 & 1000 & 1000 \\
3 & 1000 & 1000 & 1000 & 1000 \\
1 & 1000 & 1000 & 1000 & 1000 \\
\end{pmatrix}
\]

Supply at the network nodes:

\[
(s_{i,t}) = \begin{pmatrix}
30 & 16 & 18 & 18 & 18 & 24 & 30 & 23 & 23 & 16 & 14 & 8 \\
12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

Demand at the network nodes:

\[
(d_{i,t}) = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 7 & 8 & 9 & 6 & 7 & 2 & 1 & 3 & 3 & 1 & 2 \\
0 & 0 & 6 & 8 & 7 & 3 & 5 & 3 & 0 & 5 & 3 & 4 \\
3 & 3 & 5 & 1 & 6 & 5 & 4 & 6 & 4 & 2 & 2 & 1 \\
2 & 0 & 5 & 9 & 4 & 8 & 5 & 4 & 1 & 1 & 1 & 5 \\
\end{pmatrix}
\]

Storage costs:

\[
(c_{i,stock}^t) = \begin{pmatrix}
0 & 0 & 0 & 0 \\
\end{pmatrix}
\]
Appendix A  Data and Results of the Calculations in Section 4.4

Per-vehicle transport costs:

\[
(c_{i,j}^{\text{transport}}) = \begin{pmatrix}
1000 & 1 & 2 & 3 & 1 \\
1 & 1000 & 1000 & 1000 & 1000 \\
2 & 1000 & 1000 & 1000 & 1000 \\
3 & 1000 & 1000 & 1000 & 1000 \\
1 & 1000 & 1000 & 1000 & 1000 \\
\end{pmatrix}
\]

A.1.4 Network 4: Complex Network

As a fourth type of network, a complex structure was chosen. That means that supplies and demands arise at all nodes without a specific pattern. As it turned out that this type of problem is harder to optimise, a network with only three nodes was used. Also the planning horizon was limited to only 6 periods.

Distances between network nodes:

\[
(m_{i,j}) = \begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{pmatrix}
\]

Supply at the network nodes:

\[
(s_{i,t}) = \begin{pmatrix}
28 & 34 & 29 & 29 & 29 & 20 \\
26 & 28 & 38 & 35 & 26 & 39 \\
29 & 0 & 19 & 0 & 35 & 0 \\
\end{pmatrix}
\]

Demand at the network nodes:

\[
(d_{i,t}) = \begin{pmatrix}
12 & 0 & 14 & 0 & 22 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
22 & 0 & 24 & 0 & 16 & 29 \\
\end{pmatrix}
\]
A.2 Optimal Solutions

Storage costs:

\[(c_{i}^{stock}) = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}\]

Per-vehicle transport costs:

\[(c_{i,j}^{transport}) = \begin{pmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{pmatrix}\]

A.2 Optimal Solutions

For the planning problems raised with the examples in the previous section, the optimal solutions are listed here.

A.2.1 Network 1

The optimal solution (with an objective value of 20) for the distribution network example is:

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>t</th>
<th>x_{i,j,t}</th>
<th>f_{i,j,t}</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>7</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>9</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>

Table A.1: Optimal transports in the distribution network example.
Appendix A  Data and Results of the Calculations in Section 4.4

A.2.2 Network 2

The optimal solution (with an objective value of 209) for the hub-and-spoke network example is:

Table A.2: Optimal stock holding in the distribution network example.

<table>
<thead>
<tr>
<th>i</th>
<th>t</th>
<th>$y_{i,t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i</th>
<th>t</th>
<th>$y_{i,t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>29</td>
</tr>
</tbody>
</table>

Table A.3: Optimal transports in the hub-and-spoke network example.

<table>
<thead>
<tr>
<th>i</th>
<th>t</th>
<th>$x_{i,j,t}$</th>
<th>$f_{i,j,t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i</th>
<th>t</th>
<th>$x_{i,j,t}$</th>
<th>$f_{i,j,t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>
### A.2 Optimal Solutions

#### A.2.3 Network 3

The optimal solution (with an objective value of 33) for the procurement network example is:

Table A.4: Optimal stock holding in the hub-and-spoke network example.

<table>
<thead>
<tr>
<th>i</th>
<th>t</th>
<th>$y_{i,t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>27</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>27</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>42</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>48</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i</th>
<th>t</th>
<th>$y_{i,t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>54</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>61</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

Table A.5: Optimal transports in the procurement network example.

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>t</th>
<th>$x_{i,j,t}$</th>
<th>$f_{i,j,t}$</th>
<th>i</th>
<th>j</th>
<th>t</th>
<th>$x_{i,j,t}$</th>
<th>$f_{i,j,t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>17</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix A  Data and Results of the Calculations in Section 4.4

<table>
<thead>
<tr>
<th>i</th>
<th>t</th>
<th>y_{i,t}</th>
<th>i</th>
<th>t</th>
<th>y_{i,t}</th>
<th>i</th>
<th>t</th>
<th>y_{i,t}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>29</td>
<td>2</td>
<td>10</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>11</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>23</td>
<td>3</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>46</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>62</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>76</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>84</td>
<td>3</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>9</td>
<td>3</td>
<td>10</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>9</td>
<td>5</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>7</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.6: Optimal stock holding in the procurement network example.

A.2.4 Network 4

The optimal solution (with an objective value of 33) for the complex network example is:

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>t</th>
<th>x_{i,j,t}</th>
<th>f_{i,j,t}</th>
<th>i</th>
<th>j</th>
<th>t</th>
<th>x_{i,j,t}</th>
<th>f_{i,j,t}</th>
<th>i</th>
<th>t</th>
<th>y_{i,t}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>15</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>43</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>61</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>99</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>58</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>93</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>68</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>94</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>39</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>81</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>165</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>26</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.7: Optimal transports and stock holding in the complex network example.
A.3 Test of Candidate Selection and Branching Rules

When testing all combinations of candidate selection and branching rules in Section 4.4.2 all of the networks were optimised each time. The detailed results for the different networks are listed in the following table:

<table>
<thead>
<tr>
<th>Candidate Selection</th>
<th>Branching Rule</th>
<th>Network example</th>
<th>Calculation time (s)</th>
<th>Optimality gap after 10 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth-first</td>
<td>Most fractional</td>
<td>Network 1</td>
<td>40</td>
<td>126</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Most fractional</td>
<td>Network 2</td>
<td>75</td>
<td>217</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Most fractional</td>
<td>Network 3</td>
<td>15</td>
<td>42</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Most fractional</td>
<td>Network 4</td>
<td>460</td>
<td>480</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Most expensive</td>
<td>Network 1</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Most expensive</td>
<td>Network 2</td>
<td>120</td>
<td>137</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Most expensive</td>
<td>Network 3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Most expensive</td>
<td>Network 4</td>
<td>85</td>
<td>195</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Best utilisation</td>
<td>Network 1</td>
<td>5</td>
<td>55</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Best utilisation</td>
<td>Network 2</td>
<td>10</td>
<td>191</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Best utilisation</td>
<td>Network 3</td>
<td>25</td>
<td>176</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Best utilisation</td>
<td>Network 4</td>
<td>340</td>
<td>407</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Worst utilisation</td>
<td>Network 1</td>
<td>10</td>
<td>133</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Worst utilisation</td>
<td>Network 2</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Worst utilisation</td>
<td>Network 3</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Worst utilisation</td>
<td>Network 4</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Max. cost change</td>
<td>Network 1</td>
<td>10</td>
<td>45</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Max. cost change</td>
<td>Network 2</td>
<td>25</td>
<td>43</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Max. cost change</td>
<td>Network 3</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Depth-first</td>
<td>Max. cost change</td>
<td>Network 4</td>
<td>435</td>
<td>470</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most fractional</td>
<td>Network 1</td>
<td>5</td>
<td>249</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most fractional</td>
<td>Network 2</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most fractional</td>
<td>Network 3</td>
<td>10</td>
<td>111</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most fractional</td>
<td>Network 4</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most expensive</td>
<td>Network 1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most expensive</td>
<td>Network 2</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most expensive</td>
<td>Network 3</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Best-first</td>
<td>Most expensive</td>
<td>Network 4</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Best-first</td>
<td>Best utilisation</td>
<td>Network 1</td>
<td>10</td>
<td>68</td>
</tr>
<tr>
<td>Best-first</td>
<td>Best utilisation</td>
<td>Network 2</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Best-first</td>
<td>Best utilisation</td>
<td>Network 3</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Best-first</td>
<td>Best utilisation</td>
<td>Network 4</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Best-first</td>
<td>Worst utilisation</td>
<td>Network 1</td>
<td>45</td>
<td>430</td>
</tr>
<tr>
<td>Best-first</td>
<td>Worst utilisation</td>
<td>Network 2</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Best-first</td>
<td>Worst utilisation</td>
<td>Network 3</td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>Best-first</td>
<td>Worst utilisation</td>
<td>Network 4</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
<tr>
<td>Best-first</td>
<td>Max. cost change</td>
<td>Network 1</td>
<td>10</td>
<td>59</td>
</tr>
<tr>
<td>Best-first</td>
<td>Max. cost change</td>
<td>Network 2</td>
<td>575</td>
<td>575</td>
</tr>
<tr>
<td>Best-first</td>
<td>Max. cost change</td>
<td>Network 3</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Best-first</td>
<td>Max. cost change</td>
<td>Network 4</td>
<td>&gt;600</td>
<td>&gt;600</td>
</tr>
</tbody>
</table>

Table A.8: Detailed results of the test of the candidate selection and branching rules.
Appendix A Data and Results of the Calculations in Section 4.4

A.4 Results for the Test of the Additional Constraints

The table below refers to Section 4.4.3. It shows the detailed results for each of the networks.

![Table A.9: Detailed results of the test of the additional constraints (with depth-first search and branching on the most expensive transport).](Image)

Table A.9: Detailed results of the test of the additional constraints (with depth-first search and branching on the most expensive transport).


Bibliography


Dong, Jing-Xin and Dong-Ping Song (2009). ‘Container fleet sizing and empty repositioning in liner shipping systems’. In: Transportation Research Part E: Logistics and Transportation Review 45 (6), pp. 860–877.


Bibliography


Bibliography


Bibliography


Bibliography


Zusammenfassung: Transport- und Kapazitätsplanung für Mehrwegbehälter unter Berücksichtigung sprungfixer Transportkosten


Die mathematischen Methoden, die für diese Arbeit entwickelt wurden, bilden einen flexiblen Baukasten aus zwei Grundmodellen und mehreren Erweiterungen, die alle miteinander kombiniert werden können. Das erste Grundmodell unterstellt lineare Transportkosten. Für diesen Fall konnte gezeigt werden, dass das Modell in das gut untersuchte Minimalkostenflussproblem überführt werden kann, für das sehr effiziente Lösungsverfahren verfügbar sind. Das zweite Grundmodell hingegen ersetzt die linearen Transportkosten durch sprungfixe Kosten. Da es sich hierbei

Der Forschungsbeitrag besteht im Wesentlichen aus zwei Bereichen. Auf der einen Seite wurden die besonderen Eigenarten von Behältersystemen und die hierzu verfügbare Literatur katalogisiert. Hieraus wurden dann mathematische Planungsmodelle abgeleitet und Lösungsverfahren aufgezeigt. Auf der anderen Seite wurde dabei das in der Literatur bereits wohlbekannte Minimalkostenflussproblem um den Aspekten nicht-linearer, stufenweise ansteigender Kosten erweitert. Das entstandene Modell ist auch für die allgemeine Transportplanung nützlich und die bei der Entwicklung aufgedeckten Ansätze und offenen Fragen bilden einen guten Ausgangspunkt für die gezielte weitere Forschung in diesem Bereich.
Erklärung

Die vorliegende Arbeit wurde selbständig und ohne unerlaubte fremde Hilfe angefertigt. Es wurden dabei keine anderen als die in der Arbeit angegebenen Schriften und Hilfsmittel benutzt.Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Es wurde auch nicht die Hilfe von Vermittlungs- oder Beratungsdiensten wie beispielsweise Promotionsberatern in Anspruch genommen.

Ich habe bisher keine Promotionsversuche unternommen, keine Promotionen abgeschlossen und auch die vorliegende Dissertation weder in gleicher noch anderer Form in einem anderen Versuch oder Prüfungsverfahren vorgelegt.