University of Bremen

Chair of Logistics

PhD Thesis

# Operative Planning with Exchangeable and Mandatory Tasks

## Applications to Lot Size Planning and Transportation Planning

*Author:*

Mario Ziebuhr

Kulenkampffallee 117

28213 Bremen

*Submitted to:*

Prüfungsausschuss Dr. rer. pol.

University of Bremen

*Reviewers:*

Prof. Dr.-Ing. Herbert Kopfer

Prof. Dr. Tobias Buer

*Date of Colloquium:*

January 30, 2017

# Contents

# List of Figures

# List of Tables

## Abbreviations

| | |
|---|---|
| ACC-MA | Alternating and Converging Constraint Memetic Algorithm |
| ACO | Ant Colony Optimization |
| ALNS | Adaptive Large Neighborhood Search |
| CACM | Collaborative Ant Colony Metaheuristic |
| CC | Common Carrier |
| CGB-heuristic | Column Generation-based Heuristic |
| CGB-HER | CGB-heuristic with a Heterogeneous Vehicle Fleet |
| CGB-HOM | CGB-heuristic with a Homogenous Vehicle Fleet |
| CIOTP | Collaborative Integrated Operational Transportation Planning |
| CIOTPP | Collaborative Integrated Operational Transportation Planning Problem |
| CIOTPP-FL | Collaborative Integrated Operational Transportation Planning Problem with Forwarding Limitations |
| CP | Centralized Planning |
| CSP | Collaborative Local Search with Side Payments |
| CTP | Collaborative Transportation Planning |
| CTPP | Collaborative Transportation Planning Problem |
| CTPP-FL | Collaborative Transportation Planning Problem with Forwarding Limitations |
| DB | Daily Basis |
| DMLULSP | Distributed Multi-Level Uncapacitated Lot-Sizing Problem |
| DULR | Distributed Lot-Sizing Problem with Rivaling Agents |
| DULR-PL | Distributed Lot-Sizing Problem with Rivaling Agents with Production Limitations |
| ERP | Enterprise Resource Planning |
| ES | Evolutionary Strategy |
| HACM | Hierarchical Ant Colony Metaheuristic |
| MIP | Mixed Integer Programming |
| MLLS | Multi-Level Lot-Sizing |
| MLULSP | Multi-Level Uncapacitated Lot-Sizing Problem |
| MP | Master Problem |
| MRP | Material Requirements Planning |
| MRPII | Material Resource Planning |
| NBM | Negotiation Based Mechanism |
| NBM-2 | Negotiation Based Mechanism of Section 4.2 |
| NBM-3 | Negotiation Based Mechanism of Section 4.3 |
| NBM-H | Negotiation Based Mechanism of Homberger (2010) |
| NBM-PWR | Negotiation Based Mechanism with Part-Way Resets |

| | |
|---|---|
| NBM-PWR-1 | Negotiation Based Mechanism with Part-Way Resets of Section 4.1 |
| NBM-PWR-B | Best Solution of Negotiation Based Mechanism with Part-Way Resets |
| NBM-PWR-W | Worst Solution of Negotiation Based Mechanism with Part-Way Resets |
| NBM-PWR-$\varnothing$ | Mean Solution of Negotiation Based Mechanism with Part-Way Resets |
| INS | Insertion Attempts |
| IOTP | Integrated Operational Transportation Planning |
| IOTPP | Integrated Operational Transportation Planning Problem |
| IOTPP-FL | Integrated Operational Transportation Planning Problem with Forwarding Limitations |
| IP | Isolated Planning |
| LNS | Large Neighborhood Search |
| PDP | Pickup and Delivery Problem |
| PD-pair | Pickup and Delivery Pair |
| PDSP | Pickup and Delivery Selection Problem |
| PDSP-CR | Pickup and Delivery Selection Problem with Compulsory Requests |
| PDPTW | Pickup and Delivery Problem with Time Windows |
| PWR | Part-Way Resets |
| RB | Route Basis |
| SA | Simulated Annealing |
| SC1 | Scenario 1 |
| SC2 | Scenario 2 |
| SC3 | Scenario 3 |
| SCP | Set Covering Problem |
| SIO | Successive Insertion Operator |
| SPP | Set Partitioning Problem |
| SPP-LP | Set Partitioning Problem linear relaxed |
| TPP | Transportation Planning Problem |
| TSP | Traveling Salesman Problem |
| VRP | Vehicle Routing Problem |
| VRPTW | Vehicle Routing Problem with Time Windows |

# 1. Introduction

In modern supply chains, companies are forced to improve their profitability and flexibility in highly dynamic and competitive consumer markets. In recent years, where the value of international exports rose significantly, the management of supply chains has been evolved as a major management issue. A manual planning is not sufficient under these circumstances. Due to improved information technology and reduced computational effort, companies are able to optimize their processes by using sophisticated optimization methods corresponding to operations research. Real world problems can be transferred into mathematical models by using operations research, which can be solved by different solution approaches.

Most of the optimization methods focus on the optimization of the internal resources of companies. However, companies also use external resources for fulfilling their tasks (e.g., production of items or transportation of goods). By using external resources, companies are able to reduce their costs by increasing the complexity of their planning. This issue is addressed in literature by developing new solution approaches which optimize the management of internal and external resources simultaneously. It is usually assumed in these solution approaches that a task can be fulfilled by internal or external resources (referred to as exchangeable task). However, there are some tasks in practice where the application of certain external resources is prohibited due to contractual obligations (referred to as mandatory tasks). A lack of research is identified in terms of analyzing mandatory tasks within supply chains. That is why new mathematical models and solution approaches have to be developed. The integration of mandatory tasks enables the application of computational studies which determine the impact of mandatory tasks on the operational costs of a company.

## 1.1. Scope of the research

In supply chain networks "hundreds and thousands of individual decisions have to be made and coordinated every minute". Thereby, "planning supports decision-making by identifying alternatives of future activities and selecting some good ones or even the best one". (Stadtler et al., 2015, p. 81) An overview of different planning tasks within supply chains is given in Figure 1.1. This figure is also known as supply chain planning matrix where planning tasks are ordered within two dimensions: supply chain process (procurement, production, distribution, and sales) and planning horizon (long-term, medium-term, and short-term planning). In this figure, each module (i.e., planning task) is linked by information flows where higher planning modules limit the plans of lower modules and the results of lower modules are used as a feedback system for higher modules. (Stadtler et al., 2015, pp. 81-90)

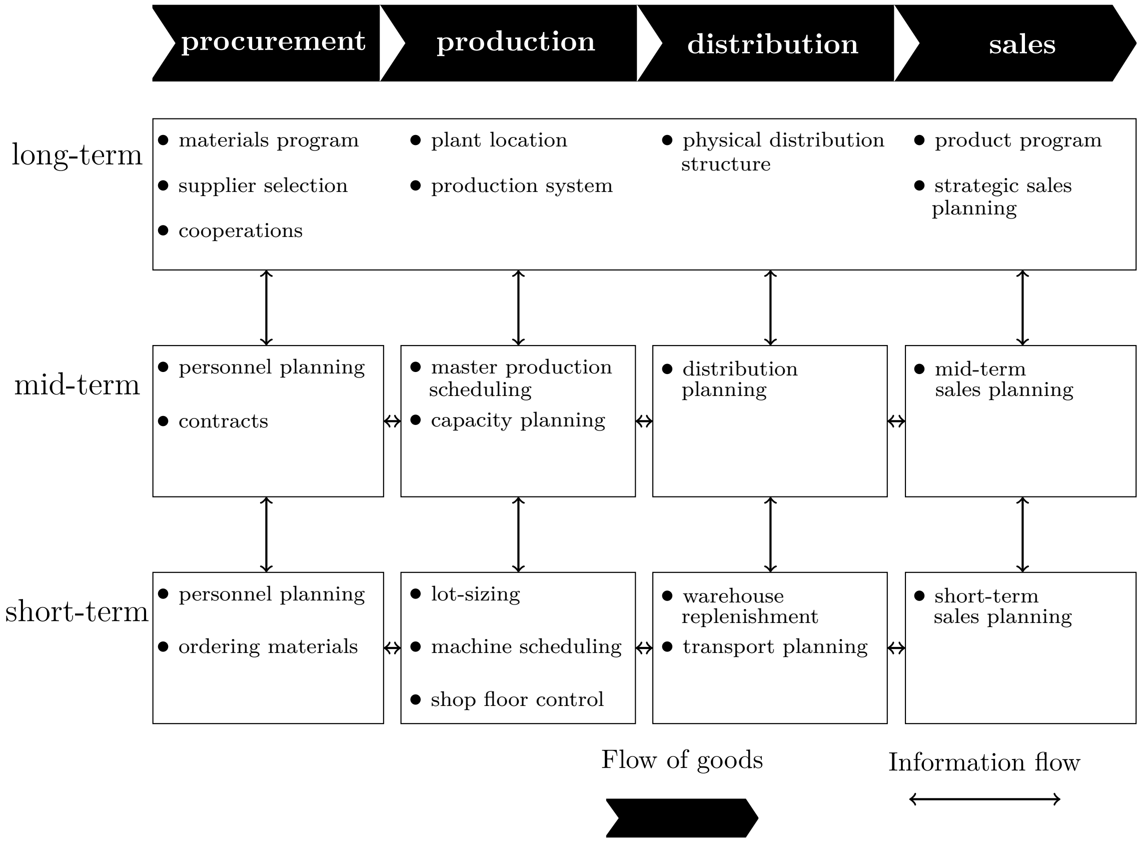


***Figure 1.1.:*** *Supply chain planning matrix, cf. Stadtler et al. (2015, p. 87)*

As can be seen in this figure, different self-interested companies align their planning with the goal to fulfill a task. Different kinds of companies are involved within supply chains (e.g., manufacturers, suppliers, transportation companies, and retailers). One goal of a supply chain is to offer a service or product at minimum costs. The main cost drivers within supply chains are related to the production, transportation, inventory, and material handling. Out of the listing production and transportation costs usually represent the highest cost components within supply chains. That is why optimization methods often focus on production and transportation planning problems.

Production planning is responsible for the manufacturing process within a supply chain where companies (referred to as manufacturers) align their production plans in order to fulfill the demand of customers. A common research field is represented by lot size planning problems (referred to lot-sizing problem), which belong to the short-term production planning. In a lot-sizing problem, the required output is usually given for a subset of tasks (e.g., final goods). A task is fulfilled by producing raw materials and components. Production occurs in batches, which are also denoted as lot sizes. The question is how large these lot sizes should be, taking into account that larger lot sizes reduce the total setup costs of the machines but increase the total inventory holding costs whereas smaller lot sizes have a contrary effect. In addition, there are usually input-output relationships between different tasks along with other constraints. These relationships complicate lot-sizing problems. A lot-sizing problem is already computationally challenging when all tasks belong to one decision maker (i.e., one manufacturer) but the complexity increases when tasks belong to different decision makers. It is usually assumed that each task can only be produced by one manufacturer. This assumption is not sufficient and represents a simplification in practice.

To move goods within a supply chain, an efficient transportation planning is crucial. Transportation planning is usually assigned to the short-term distribution planning where a transportation company (referred to as freight forwarder) can fulfill a task of a customer by using their own fleet (referred to as self-fulfillment), external carriers (referred to as subcontracting) or cooperating freight forwarders within a horizontal coalition (referred to as collaboration). The difference between subcontracting and collaboration is the relationship among the freight forwarders and carriers which is either a hierarchically related partnership in terms of subcontracting or an equal partnership in terms of collaboration. In the literature, the topic of transportation planning in combination with different fulfillment modes is addressed recently. It is usually assumed that each task is independent regarding the fulfillment mode. It means that a task can either be fulfilled by self-fulfillment, subcontracting or a member within a coalition. Based on the selection of a fulfillment mode, vehicle routing problems have to be solved, which deal with the determination of the best sequence for serving a given set of customers under different constraints (e.g., routing, capacity, and time constraints).

In the literature, which deals with optimizing lot-sizing problems and transportation planning problems (TPPs), it is assumed that a task can be fulfilled by internal or external resources. A task is outsourced in these scenarios when the private capacity is exceeded or when it is less costly. However, using appointed resources for fulfilling a task is sometimes prohibited due to contractual obligations. These contractual obligations are motivated by clients and contractors. On the one hand, when a client has safety concerns or premium goods he or she might be willing to pay more when the fulfillment remains by a certain fulfillment mode of a contractor instead of using external resources. On the other hand, a contractor might be willing to use internal resources when a task belongs to a strategic client with high margins where he or she does not want to share confidential information, like contact details, to possible competitors. The consideration of exchangeable and mandatory tasks represents a common issue within supply chains, which is rarely addressed. The goal of the thesis is to fill this research gap for optimization problems in lot size planning and transportation planning.

## 1.2. Objective and structure of the thesis

Companies are able to improve their profitability by using internal and external resources. However, companies have to be aware that some tasks cannot be fulfilled by certain resources. These mandatory tasks affect the planning processes within supply chains by an increased complexity where certain resources have to be reserved in advance. To deal with this situation, existing mathematical models and solution approaches have to be extended. This thesis focuses on optimization problems in lot size and transportation planning. New mathematical models and solution approaches allow the application of detailed computational studies. Companies are able to evaluate the impact of mandatory tasks for their planning scenarios due to these studies.

The thesis is organized into two parts considering the mentioned research fields: lot size planning with mandatory tasks (Part I) and transportation planning with mandatory tasks (Part II). In general, both parts give an overview about related existing mathematical models and solution approaches and how these have to be extended in terms of mandatory tasks. It is worth mentioning that different premises are given in both research fields. For example, there are solution approaches in transportation planning which consider internal and external resources simultaneously. However, it is not the case in lot size planning where tasks can only be produced by internal resources. The selection of the best resource for a task is computationally challenging and it is usually assumed that a task can be fulfilled by any resources. This thesis considers tasks which have to be fulfilled by predefined resources. These tasks are known as mandatory tasks. In the mentioned research fields, the impact of mandatory tasks is discussed separately from each other due to different premises.

In Part I, the topic of mandatory tasks in lot size planning is discussed. The term tasks stands for items. Chapter 2 begins with an introduction regarding the relevance of lot size planning in material resource planning. Afterward, the chapter focuses on multi-level lot-sizing problems. It is distinguished between single decision making and group decision making problems. In a single decision making problem, one manufacturer is in charge of the whole supply chain with all its items. There are several autonomous manufacturers in a group decision problem, where each manufacturer is in charge of a set of items. Obviously, the latter type is more common within supply chains. A negotiation based mechanism is presented to solve the corresponding lot-sizing problems. The mechanism is a heuristic, which uses an iterative procedure for proposing new solutions to the members within a supply chain. A solution is accepted as a mutually accepted solution when all members agree on it.

The applied negotiation based mechanism is known from literature and is rebuilt from scratch. In Chapter 3, the mechanism is extended by a part-way reset procedure. The applied part-way reset procedure allows to return to a previous solution by also modifying some parameters which influence the decisions of the members within a supply chain. Two computational studies are performed for evaluating the extended mechanism. In the first study, the impact of the part-way reset procedure is analyzed by comparing the mechanism with and without the procedure. Following a second study where the extended mechanism is evaluated by six state of the art heuristics.

Chapter 4 presents collaborative multi-level uncapacitated lot-sizing problems. There the assumption is dropped that each item can only be produced by one manufacturer. The chapter is divided into three sections. First, a mathematical model with a sole item-production is presented in Section 4.1. There manufacturers compete with each other regarding the production of certain items. Finally, the manufacturer with lowest costs gets the whole production volume of an item. Second, a mathematical model with a multiple item-production is introduced in Section 4.2 where each item can be produced by more than one manufacturer. In terms of both extensions, it is necessary to extend the presented negotiation based mechanism by a procedure which is able to identify a suitable shared production among the manufacturers. Both approaches are evaluated by computational studies. Third, a collaborative multi-level uncapacitated lot-sizing problem with exchangeable and mandatory items is introduced in Section 4.3. An exchangeable item can be produced by several manufacturers, while a mandatory item can only be produced by an appointed manufacturer. It is necessary to extend the negotiation based mechanism for the consideration of mandatory items such that it is ensured that the appointed manufacturer produces all units of a mandatory item. The impact of these mandatory items is analyzed in several computational studies.

Part II describes the consideration of mandatory tasks in different TPPs. There the term tasks stands for requests. In Chapter 5, an overview about transportation planning is presented. The thesis focuses on TPPs where goods have to be transported from a pickup node to a delivery node. First, single decision making problems are presented. Afterward, group decision making problems are discussed. Then, an overview about mandatory requests in TPPs is given, where it can be seen that there is still a lack of research. An existing column generation-based heuristic (CGB-heuristic) is applied to solve the considered TPPs, which divides a TPP into a subproblem and a master problem. In this thesis, the subproblem is solved by an adaptive large neighborhood search (ALNS) while the master problem is solved by the commercial solver CPLEX.

In Chapter 6, the CGB-heuristic is modified such that the heuristic is suitable for a TPP with exchangeable and mandatory requests. Due to the fact that the CGB-heuristic is already known, the chapter focuses on the modifications of this solution approach. Three solution strategies are proposed which are able to handle different types of requests. These solution strategies differ in terms of the consideration of mandatory requests within the CGB-heuristic and are denoted as: strict generation procedure, strict composition procedure, and repair procedure. Two computational studies are presented. One study determines the best basic solution approach by comparing the results of the ALNS with two different versions of the CGB-heuristic while the other one determines the performance of the best basic solution approach with the strict generation procedure compared to a state of the art heuristic.

Chapter 7 presents a TPP with self-fulfillment and subcontracting as fulfillment modes. This problem is denoted as integrated operational transportation planning (IOTP) problem. The IOTP problem is extended by two different types of mandatory requests and is solved by a CGB-heuristic. The CGB-heuristic is modified in terms of the mentioned solution strategies. In a computational study, the best solution strategy has to be determined by evaluating the performance of different solution strategies in terms of the solution quality and number of feasible solutions. The best solution strategy is used for a detailed computational study, where the impact of mandatory requests is analyzed on different transportation scenarios.

In Chapter 8, a TPP with self-fulfillment and collaboration as fulfillment modes is considered. This problem is denoted as collaborative transportation planning (CTP) problem.

The CTP problem is extended by two different types of mandatory requests. Two of the three solution strategies of Chapter 7 are applied to solve this problem: strict generation and strict composition procedure. The repair procedure is skipped due to the results of the previous chapter. The impact of mandatory requests is analyzed in a computational section.

Chapter 9 presents a TPP with self-fulfillment, subcontracting, and collaboration as fulfillment modes. This problem is denoted as collaborative integrated operational transportation planning (CIOTP) problem. The CIOTP problem is extended by four different types of mandatory requests. The problem represents a combination of the presented IOTP and CTP problem. As in the previous chapter, the strict generation and strict composition procedure are applied. Two different computational studies are presented. One study analyzes the impact of mandatory requests separately from each other as in the previous chapters, while the other study evaluates the impact of various types of mandatory requests simultaneously. The consideration of various types of mandatory requests seems to be more realistic.

Finally, the main findings of the thesis are summarized in Chapter 10. Furthermore, an overview about promising future research regarding mandatory tasks is presented.

# Part I.

# Lot Size Planning with Mandatory Tasks

# 2. Lot Size Planning

The lot-sizing problem represents a basic optimization problem in material requirements planning (MRP). By solving lot-sizing problems, the goal is to determine the "best" (i.e., cost efficient) lot sizes for one or several items over several planning periods. First, lot size planning was primarily applied within a manufacturing company, but recently it is applied within supply chains with different manufacturers. The goal is to align the lot sizes over all manufacturers in order to reduce production, inventory holding, and setup costs. There are several different mathematical models and solution approaches which try to optimize the lot sizes within supply chains. However, two issues are not addressed. First, it is always assumed that each item can only be produced by one manufacturer. Second, it is assumed that there are no items which have to be produced by an appointed manufacturer. Both subjects will be discussed in the following chapters.

Section 2.1 classifies the MRP into the field of production planning. Lot size planning represents one planning phase in MRP. In lot size planning, different kinds of lot-sizing problems are known. These lot-sizing problems differ by the number of end products, level of the production structure, and planning periods. In this section, a scheme is presented which classifies lot-sizing problems.

Methods of operations research can be applied for determining suitable lot sizes. In the literature, multi-level lot-sizing (MLLS) problems are often applied, where an overview is given in Section 2.2. In MLLS problems, items (i.e., raw materials, components, and end products) have to be produced over several planning periods. The goal is to identify a production plan which reduces the setup and inventory holding costs. In this case, it is distinguished between two planning scenarios: single decision making and group decision making. Single decision making means that one decision maker is assumed to be in charge of all relevant planning data. In contrast, group decision making means that several decision makers are in charge of relevant planning data where each decision maker is aware of its individual data. Corresponding to MLLS problems, a single decision making problem is presented in Subsection 2.2.1 and a group decision making problem is presented in Subsection 2.2.2. In both subsections, the mathematical model as well as state of the art solution approaches are presented and discussed.

Closing this chapter, a certain solution approach is described in Section 2.3, which is able to solve MLLS problems with single and multiple decision makers. This solution approach is denoted as negotiation based mechanism (NBM) and was introduced by Klein et al. (2003a); Fink (2004); Homberger (2010). The NBM is a heuristic which uses an indirect problem representation and a simulated annealing (SA) as an acceptance criterion.

## 2.1. Material requirements planning

Manufacturers face larger and more complex product structures due to increasing customer needs (Dellaert and Jeunet, 2000). To deal with this situation, manufacturers use advanced information systems like enterprise resource planning (ERP) systems. In ERP, the concept of MRP represents an essential system, which is used in manufacturing and inventory management (Steinberg and Napier, 1980; Homberger and Gehring, 2009).

MRP belongs to production planning. Corresponding to Karimi et al. (2003), "production planning is an activity that considers the use of production resources in order to satisfy production goals over a certain period named the planning horizon". In production planning, three different time ranges are considered corresponding to Karimi et al. (2003):

- Long-term decisions focus on strategic decisions as product and process choices.
- Medium-term decisions focus on material requirements planning.
- Short-term decisions focus on day-to-day scheduling of operations.

Since the introduction of MRP in late 1960's, the relevance of the system has grown at a rapid rate in production planning (Yelle, 1979). MRP is responsible for generating a "production plan for each item over a given planning horizon" (Pitakaso et al., 2007). To be precise, it means that the task is "to calculate time-phased plans of secondary demands for components and parts based on a time series of primary demands (usually end products)" (Stadtler et al., 2015). Lot sizes have to be determined for all components and parts in all periods. That is why lot-sizing decisions represent one of the most important and difficult decision problems (Karimi et al., 2003).

In terms of MRP, there is the issue that it is unknown if a generated production plan can be used for the production because a capacity planning is not involved in MRP. To solve this issue, MRP is extended by a capacity planning and feedback system. A feedback system ensures that whenever there is any issue with the production plan it is possible to return to a previous planning step. (Wight, 1984, p. 49ff.) By also integrating top management, MRP is extended to material resource planning (MRP II) in the mid 1970's. MRP II is a hierarchically related system with a capacity planning and feedback system. (Voß and Woodruff, 2006, p. 200) Till today, MRP II is a common system in production planning. In Figure 2.1, an overview of MRP II is presented.

MRP II implies a top-down approach where the length of the planning horizon is reduced and the level of detail is increased from business planning to shop floor control. The system is initialized by a manufacturer, who defines his or her goals, investments, and product developments. The demand of a production line and the primary demand (i.e., demand of end products) have to be determined based on these decisions. Afterward, MRP is executed which calculates the secondary demand (i.e., demand of raw materials and components) based on the primary demand. In case that a manufacturer is able to execute the generated production plan, the jobs are released and scheduled on the machines. (Wight, 1984; Holzbaur and Hachtel, 2010, p. 49ff., p. 88ff.)

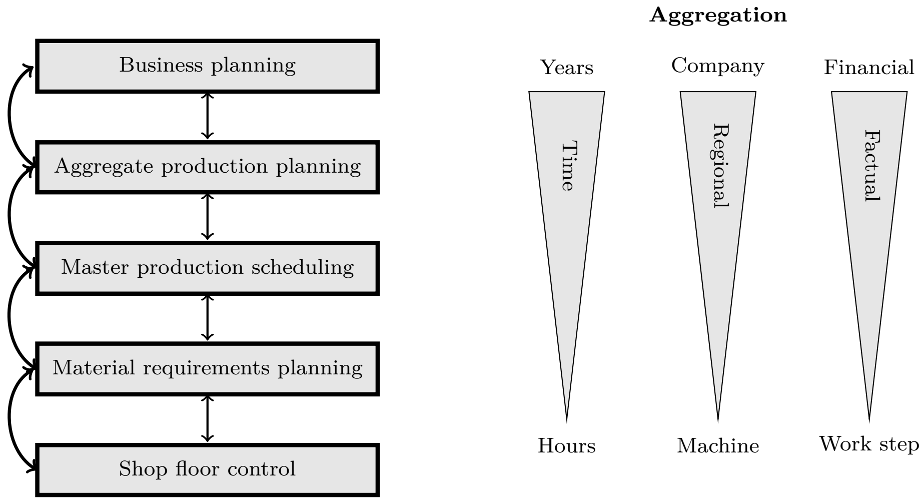


***Figure 2.1.:*** *Overview of MRP II, cf. Wight (1984); Thonemann (2005, p. 54, p. 56))*

MRP represents one phase in MRP II. The following initial data are required for the execution of MRP: primary demand, inventory levels, lead time of the items, and product structure. In Figure 2.2, common product structures are presented.

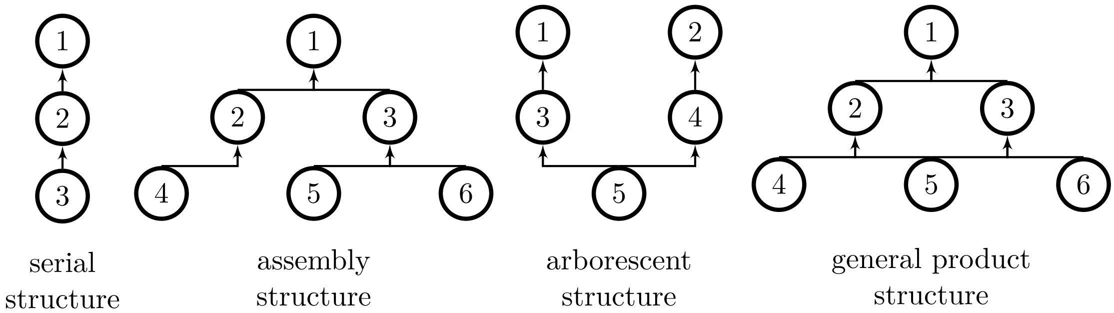


***Figure 2.2.:*** *Product structures, cf. Dellaert and Jeunet (2000), Sahling (2010, p. 12)*

A product structure can be visualized by a bill of materials or directed acyclic graph $D(N, E)$. In terms of the graph, the set of nodes is defined by $N$ and the set of edges is defined by $E$. Each node represents an item. These items are numbered in an ascending order corresponding to their production level. It means that items on the lowest level are raw materials and components, while end products are produced on the highest level. An edge from node $i$ to node $j$ indicates that item $i$ is necessary for producing item $j$. In this case, item $i$ is the predecessor of item $j$; and item $j$ is the successor of item $i$, respectively. (Dellaert and Jeunet, 2000; Homberger, 2006) Depending on the number of predecessors and successors of an item, it can be distinguished between four product structures: serial structure, assembly structure, arborescent structure, and general structure. All of these product structure are given in Figure 2.2. In a serial structure, each item has at most one direct predecessor and one direct successor. In the given serial structure, item three is a raw material (e.g., tree), item two is a component (e.g., wooden slats), and item one is an end product (e.g., wooden table). Each item may have more than one predecessor and at most one direct successor in an assembly structure. The opposite structure can be observed in an arborescent structure. A general product structure is given when each item might have more than one predecessor and successor. (Pitakaso et al., 2007)

By using a product structure and the remaining input data, the material requirements can be calculated. A sequential approach of MRP is divided into four phases. These phases are executed separately for each item of the product structure beginning by end products and ending by raw materials. In phase one, the demand of item $i$ is calculated by adding the lot sizes of the successors. Afterward, the demand is reduced by the available inventory level of item $i$ in phase two. The available inventory level is defined by the current inventory level minus a safety stock. In phase three, the lead times of the predecessors of item $i$ are taken into account by modifying the demand of item $i$. For example, if the mentioned predecessors have a lead time of one period, the demand of item $i$ will be shifted to a previous period in order that there is enough time for producing the requested demand of item $i$. Finally, the "best" lot sizes of item $i$ over a planning horizon have to be determined by taking different cost rates into account. The mentioned phases are repeated until each item of the product structure is analyzed. (Thonemann, 2005; Tempelmeier, 2006, pp. 312-315, p. 116)

As mentioned, the determination of the type, the quantity, and the due dates of the required factors of production is subject to MRP. This also involves lot-sizing decisions. Corresponding to Zangwill (1969), the idea of lot-sizing is that "a product is produced (or purchased) in batch quantities and then placed in stock. After the inventory has been sufficiently depleted, another batch is produced. The entire cycle just described is then repeated as often as necessary". In a lot-sizing problem, the required output per item and per time period is given for a subset of items to produce (end products). The question is how large these lot sizes should be by taking the mentioned contrary effect of inventory holding and setup costs into account.

Lot-sizing decisions affect directly the performance of a supply chain, which means it affects the ability to compete with the market (Karimi et al., 2003). In MRP, several lot-sizing problems are known. A lot-sizing problem of a manufacturer is defined by their type of organization and production process (Sahling, 2010). An overview about different lot-sizing models can be found in Tempelmeier (2006, p. 131ff.). A classification scheme is presented for categorizing lot-sizing models. In general, lot-sizing models can be classified corresponding to their degree of information, time structure, products, and machines (Sahling, 2010, p. 10). A classification scheme for lot-sizing models is presented in Figure 2.3.

On the first level of the scheme, it is distinguished between a deterministic and probabilistic data set. In lot-sizing models with a deterministic data set, it is assumed that the demand values are known in advance, while the demand values are uncertain to a certain degree in a probabilistic data set. It means that the occurrence of demand values depend on a probability. By considering a deterministic data set, two demand types are common. A static demand means that the given external demand for end products do

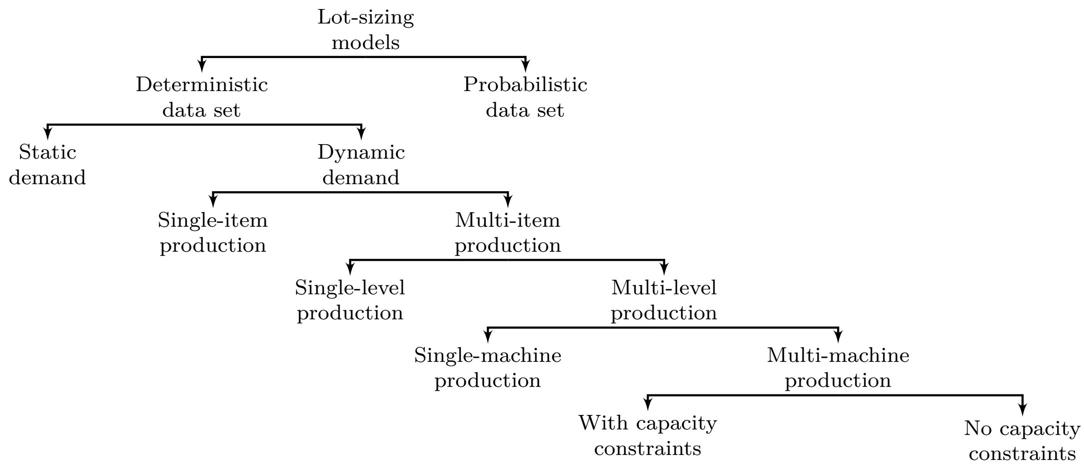


***Figure 2.3.:*** *Classification of lot-sizing models, cf. Karimi et al. (2003), Sahling (2010, p. 11)*

not change over the planning horizon. A dynamic demand means that the given external demand for end products changes over the planning horizon. In terms of the external demand, it can be distinguished between a single-item and multi-item production. In a single-item production, there is one end product, while in a multi-item production several end products are considered. Each end product is defined by a certain product structure. In a single-level production, an end product is produced directly from raw materials or purchased material. A multi-level production means that an end product is assembled out of different components and raw materials. In the latter scenario, different raw materials and components are assembled to end products by several operations. The raw materials, components, and end products can be produced either by a single or several machines. These machines might be restricted regarding their resources (e.g., manpower, equipment or budget) or unlimited. The former case is known as a capacitated problem, while the other one is known as an uncapacitated problem. (Dellaert and Jeunet, 2000; Karimi et al., 2003)

This thesis focus on uncapacitated MLLS problems, which are common in MRP (Pitakaso et al., 2007). MLLS problems have a deterministic data set, dynamic demand, and multi-level production. Uncapacitated MLLS problems are considered due to the fact that in capacitated scenarios sensitive and detailed data have to be known (Dellaert and Jeunet, 2003). Since some data are often missing, lot-sizing problems without capacity constraints are more common (Pitakaso et al., 2007).

## 2.2. Multi-level lot-sizing problems

This section is divided into two subsections. In Subsection 2.2.1, the multi-level uncapacitated lot-sizing problem (MLULSP) is presented, which is a single decision making problem. There a solution has to be calculated which minimizes the total production costs (referred to as total costs). In Subsection 2.2.2, the distributed multi-level uncapacitated lot-sizing problem (DMLULSP) is given, which is a group decision making problem.

Thereby, a solution has to be determined which minimizes the individual costs of each member within a coalition (referred to as local costs). Both lot-sizing problems are MLLS problems without capacity constraints. In this section, the mathematical formulations of both problems are presented and discussed. Both subsections consider a mixed integer programming (MIP) problem, which can be solved by different exact and heuristic approaches.

### 2.2.1. Single decision making

One of the first MLLS problems was introduced by Yelle (1979), which is known as the MLULSP. Corresponding to Xiao et al. (2012), the MLULSP "plays an important role in the efficient operation of modern manufacturing and assembly processes". The application of the MLULSP covers mainly three situations in practice: (1) some of the relevant planning data regarding the capacity are missing, (2) a successive production planning with a capacity planning is attached, and (3) a purchasing department needs an estimation for ordering raw materials (Pitakaso et al., 2007; Homberger and Gehring, 2009). Furthermore, the MLULSP is a well-known standard problem within lot size planning due to existing benchmark instances and a mathematical model with few constraints (Homberger and Gehring, 2009).

The MLULSP features some relevant properties of supply chains like a multi-level product structure and a trade-off between inventory holding and setup costs. That is why it is often used in MRP systems (Pitakaso et al., 2007). In Figure 2.4, a multi-level product structure with five production levels is presented. To be precise, a general product structure with 40 different items is given. These items are sorted in increasing level codes in accordance to their value (end products are listed on the highest level and raw materials on the lowest level). Each of these items is a standard item, which can be produced by a central decision maker. It means that one decision maker is in charge of all items and the corresponding relevant planning data. Based on the external demand for the end products (item one to item three), the decision maker tries to compute these lot sizes and inventories of each item and all production periods which minimize the total costs. In terms of the calculation of lot sizes, the following assumptions have to be taken into account (Homberger, 2006):

- There are no capacity restrictions. It is assumed that there is no lack of resources.
- Setup and inventory holding costs do not change over the time period.
- Unit costs can be ignored.
- There are no defective items.
- There are no inventories in the first period.

This subsection focuses on the MLULSP formulation of Dellaert and Jeunet (2000), which represents an extension of the MLULSP formulation of Steinberg and Napier (1980). Steinberg and Napier (1980) developed a MIP model based on the formulation of Yelle (1979). In a MLULSP, a single member (referred to as agent) is assumed, who acts as

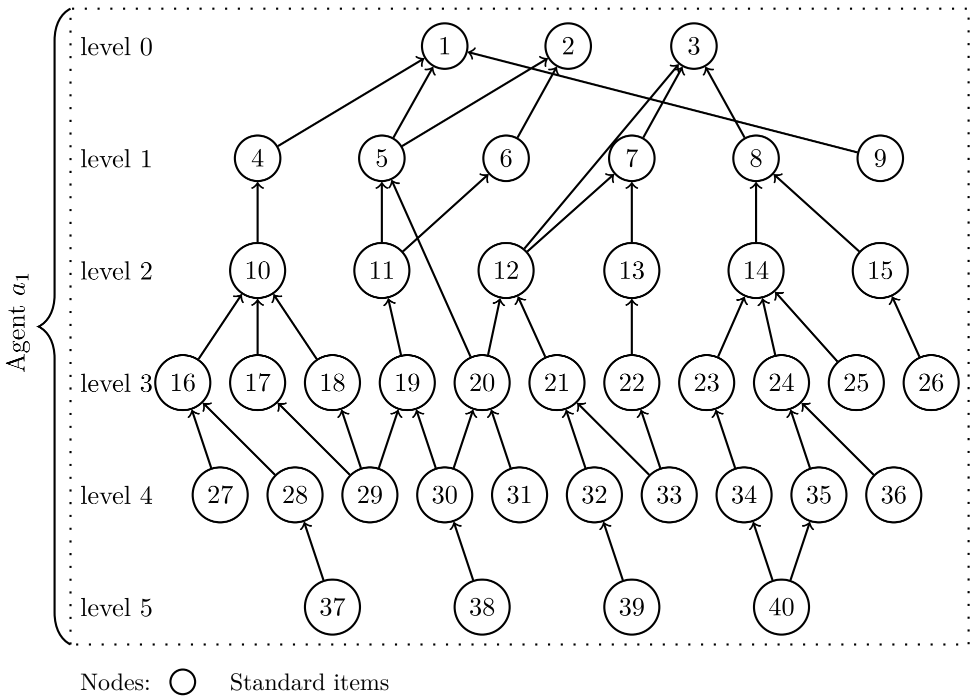


***Figure 2.4.:*** *Product structure of a MLULSP, cf. Buer et al. (2013)*

central decision maker. This agent is aware of all relevant planning data. In a MLULSP, a set $T$ of possible production periods, $T = \{1, \ldots, m\}$, and a set $I$ of items are given, $I = \{1, \ldots, n\}$. The relations among the items are defined by the corresponding product structure. Final items (i.e., end products) are assembled of one or more components, while components may themselves consist of other components or raw materials. For each item $i \in I$, a set $\Gamma^+(i) \subset I$ of all direct successors and a set $\Gamma^-(i) \subset I$ of all direct predecessors are given. Final items are characterized by $\Gamma^+(i) = \emptyset$ and raw materials are characterized by $\Gamma^-(i) = \emptyset$. In the product structure, the production coefficient $p_{ij}$ indicates the required units of item $i$ to produce one unit of item $j$. Without loss of generality, $p_{ij} = 1$ is assumed (Fink, 1997). Let $s_i, h_i$, and $t_i$ be the setup cost, inventory holding cost, and lead time, respectively, per item $i \in I$. An exogenous demand $d_{it}$ ($i \in I|\Gamma^+(i) = \emptyset$ and $t \in T$) is given for each final item. It means that it is assumed that there is just an internal demand for components and raw materials. Components and raw materials cannot be sold to customers.

In a MLULSP, the central decision making agent has to decide the lot size $x_{it}$ for each item $i \in I$ with $\Gamma^+(i) \neq \emptyset$ and each period $t \in T$. Therefore, the endogenous demand $d_{it}$ as well as the inventory $l_{it}$ for each item $i \in I$ and each period $t \in T$ have to be determined. Finally, this lot size decision also includes the setup decision, i.e., if a production of item $i$ takes place in period $t$ at all ($y_{it} = 1$) or not ($y_{it} = 0$).

To solve a MLULSP, a production plan $y$ has to be given. A production plan contains the setup decision ($y_{it}$) for each item $i \in I$ and each period $t \in T$. The mathematical model of the considered MLULSP can be formulated as follows (Ziebuhr et al., 2013):

$$\min f^{nd}(y) = \sum_{i\in I}\sum_{t\in T}(s_i y_{it} + h_i l_{it}) \tag{2.1}$$

$$\text{s. t.} \quad l_{it} = l_{i,t-1} + x_{it} - d_{it}\,, \qquad \forall i \in I,\, \forall t \in T, \tag{2.2}$$

$$l_{i,0} = 0\,, \qquad \forall i \in I, \tag{2.3}$$

$$l_{it} \geq 0\,, \qquad \forall i \in I,\, \forall t \in T \setminus \{0\}, \tag{2.4}$$

$$d_{it} = \sum_{j\in\Gamma^+(i)} p_{ij} x_{j,t+t_i}\,, \qquad \forall i \in \{j \in I \mid \Gamma^+(j) \neq \emptyset\},\, \forall t \in T, \tag{2.5}$$

$$0 \geq x_{it} - M y_{it} \qquad \forall i \in I,\, \forall t \in T, \tag{2.6}$$

$$x_{it} \geq 0\,, \qquad \forall i \in I,\, \forall t \in T, \tag{2.7}$$

$$y_{it} \in \{0,1\}\,, \qquad \forall i \in I,\, \forall t \in T. \tag{2.8}$$

In a MLULSP, the goal of a central decision making agent is to minimize his or her total costs $f^{nd}(y)$ by considering a given production plan $y$. The total costs include setup and inventory holding costs for all items $i \in I$ over all periods $t \in T$. By minimizing the total costs, constraints (2.2)–(2.8) have to be observed. The inventory balance constraints (2.2) ensure that the inventory $l_{it}$ of item $i$ at the end of the current period $t$ is determined by the inventory of the previous period $t-1$ and the amount $x_{it}$ produced in the current period minus the demand for item $i$ in the current period. For all items, the inventory of the first period (i.e., $t = 0$) is zero (constraints (2.3)); and for remaining periods the inventory is non-negative (constraints (2.4)). The endogenous demands for components and raw materials are determined by constraints (2.5). These constraints ensure that the production of item $j$ in period $t + t_i$ triggers a corresponding demand $d_{it}$ for all $i \in \Gamma^-(j)$, that is, a demand for each item $i$ preceding item $j$ in the multi-level item structure. The lot size $x_{it}$ is non-negative (constraints (2.7)). In case that item $i$ is produced in period $t$ ($x_{it} > 0$), the setup decision is set to one ($y_{it} = 1$), otherwise to zero ($y_{it} = 0$). This procedure is linearized by introducing a big-M. A big-M is a large number and the constraints are given by (2.6). As usual in literature, it is assumed that the setup decision is binary (constraints (2.8)). Thereby, it is assumed that it is sufficient when the solution approach considers only complete period demands which reduces the computational effort significantly (Kuik and Salomon, 1990).

Several exact and heuristic approaches are known for MLLS problems. Some exact approaches, which are able to solve small instances, can be found by Zangwill (1969); Crowston and Wagner (1973); Steinberg and Napier (1980); Afentakis et al. (1984); Afentakis and Gavish (1986). Zangwill (1969); Crowston and Wagner (1973) present dynamic programming approaches, while Steinberg and Napier (1980); Afentakis et al. (1984); Afentakis and Gavish (1986) propose branch and bound algorithms. Similar to exact approaches, some simple heuristics are applied, which either consist of the sequential application of single-level lot-sizing models to each item of the product structure (Yelle, 1979; Bookbinder and Koch, 1990) or the simultaneous application of multi-level lot-sizing models (Burn and Millen, 1985; Coleman and McKnew, 1991). In recent ERP systems, such heuristics are still used to solve MLLS problems (Pitakaso et al., 2007).

Till today, there are no exact approaches which are able to solve large instances optimally in reasonable time (Xiao et al., 2012) due to the fact that the MLULSP is NP-hard for general product structures (Bussieck et al., 1998). That is why heuristics are preferred. To be precise, metaheuristics are more common compared to simple heuristics. A metaheuristic is a heuristic which uses a strategy that guides the search by using several sub heuristics. Different metaheuristics are known for solving a MLULSP. Most of them use a neighborhood search for identifying better solutions and differ in terms of the problem representation. In Kuik and Salomon (1990); Tang (2004), an SA is used to solve a MLULSP. Both publications are applied on different instances. Kuik and Salomon (1990) solve instances with an assembly and general product structure, while Tang (2004) solves only small instances with an assembly product structure. Evolutionary algorithms are applied by Dellaert and Jeunet (2000); Homberger (2006, 2008). The idea of these algorithms is to generate new solutions based on existing solutions by using mutation and crossover operators. Dellaert and Jeunet (2000) use a hybrid approach with several heuristics for constructing and improving solutions, while Homberger (2006, 2008) use an evolutionary algorithm in combination with a binary coded production plan. Ant colony systems are presented by Pitakaso et al. (2007); Homberger and Gehring (2009), where artificial ants analyze the solution space. In Pitakaso et al. (2007), a two step approach is applied where an ant colony algorithm determines the sequence of items and a heuristic determines the lot sizes for each item. In contrast, Homberger and Gehring (2009) use an ant colony algorithm with a pheromone matrix combined with a binary coded production plan. A variable neighborhood search for a MLULSP is presented by Xiao et al. (2011). The idea of this solution approach is that a change of the neighborhood during the local search is allowed. An extended approach with different neighborhood search techniques is presented by Xiao et al. (2012). It can be concluded that metaheuristics are able to obtain satisfying solutions, but there is still some research left for improving the solution quality (Xiao et al., 2011).

### 2.2.2. Group decision making

The non-distributed or centralized multi-level uncapacitated lot-sizing problem can be considered as a special case of the (distributed) DMLUSLP. In a DMLULSP, the assumption is dropped that the MLULSP is solved by one agent. Instead, it is assumed that the set of items $I$ is partitioned and the responsibility to jointly produce items is assigned to a group of agents $A$ (referred to as coalition). In this case, each agent might represent an organizational unit with his or her individual goals and sensitive data. That is why the goal in the DMLULSP is to reduce the local costs of each agent instead of minimizing the total costs. The extended model is denoted as DMLULSP and was introduced by Homberger (2010).

In Figure 2.5, the product structure of a DMLULSP is presented, where two agents are in charge of a disjoint set of items. In this case, agent $a_1$ produces item 1 to 20 and agent $a_2$ produces item 21 to 40. Consequently, each item can only be produced by one agent. For example, agent $a_1$ cannot produce item five, while agent $a_2$ cannot produce item 33. With a given external demand for the end products one to three, agent $a_1$ tries to compute these lot sizes and inventories which minimize his or her local costs and afterward, agent $a_2$ does the same for the remaining items. It means that based on the decisions of agent $a_1$, the decisions of agent $a_2$ are influenced.

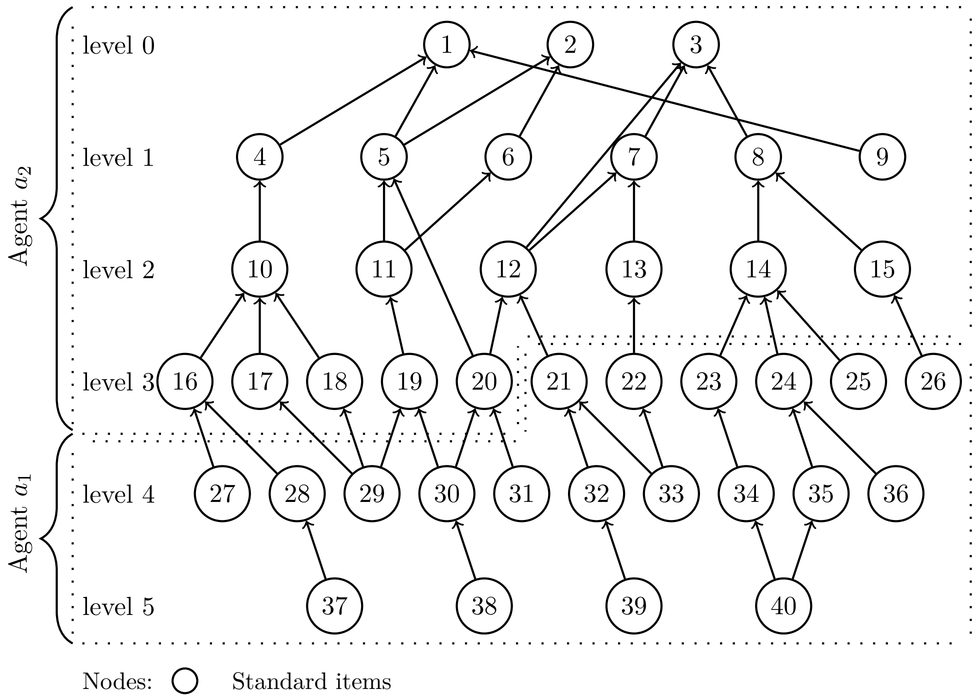


***Figure 2.5.:*** *Product structure of a DMLULSP, cf. Buer et al. (2013)*

In a DMLULSP, a coalition $A = (1, ..., k)$ with $k$ agents is assumed. Thereby, each agent $a \in A$ is in charge of a disjoint set of items $I_a$ with $\bigcup_{a \in A} I_a = I$ and $\bigcap_{a \in A} I_a = \emptyset$. A DMLULSP assumes asymmetric information regarding the cost parameters $s_i$ and $h_i$. That is, agent $a$ knows the values of $s_i$ and $h_i$ for all items he or she produces $(i \in I_a)$ but not for those items produced by the remaining agents $a'$ $(a' \in A, a' \neq a)$ and vice versa. These parameters are considered as private information of agent $a$ because the negotiation power of agent $a$ might be negatively affected when other agents knew about them (asymmetric information). For this reason, agent $a$ does not want to reveal private cost parameters to other agents. However, it is assumed that the production structure represents public information due to some kind of common industry knowledge. That is why the product structure is available to all agents (symmetric information). Although the agents have to cooperate for fulfilling the overall goals related to the production within the supply chain, each agent is still autonomous and self-interested. The individual objective function $f_a$ of agent $a \in A$ is to minimize his or her local costs for producing items $I_a$.

A DMLULSP consists of constraints (2.2)–(2.8) and objective function (2.10). The goal is to minimize the total costs, i.e., the sum of the agent's local costs. In the following, objective function (2.10) is referred to as the total cost function of the coalition and objective function (2.9) is denoted as local cost function of an individual agent $a$ $(a \in A)$. Due to the interdependencies between the items, a change in the production plan $(y)$ that reduces the total costs will usually decrease the local costs of some agents, while the local costs of other agents are increased. In that sense, the local cost functions of the agents are usually conflicting, which complicates the goal of minimizing the total costs (Homberger, 2010).

$$\min f_a(y) = \sum_{i \in I_a} \sum_{t \in T} (s_i y_{it} + h_i l_{it}) \tag{2.9}$$

$$\min f(y) = \sum_{a \in A} f_a(y) \tag{2.10}$$

As can be seen by the formulation, a DMLULSP is a coordination problem, where the production is spread across different agents. Three types of coordination approaches are known corresponding to Homberger (2010): (1) coordinated by a central decision maker, (2) coordinated by a single contract, and (3) coordinated by automated negotiation. The first approach is a centralized approach. There decision makers share their sensitive data (e.g., cost rates) and accept the decision of one decision maker. The remaining approaches are decentralized approaches where decision makers try to align their planning by using contracts or negotiations. The difference between the second and third approach is that software agents can be used easily in automated negotiations, who can be applied more directly in MRP systems. In automated negotiations, metaheuristic are often used for simulating the decentralized coordination of software agents.

Five publications are known which use this kind of negotiation process: an evolutionary strategy is used by Homberger (2011), ant colony algorithms are applied by Homberger and Gehring (2010); Buer et al. (2013), an SA is used by Homberger (2010), and a hill-climbing approach with side payments is applied by Homberger et al. (2015). These solution approaches differ in terms of the negotiation process and objective function. The evolutionary strategy (ES) was introduced by Homberger (2011) and the ant colony optimization (ACO) was introduced by Homberger and Gehring (2010). Similarly to the ACO, the collaborative ant colony metaheuristic (CACM) and the hierarchical ant colony metaheuristic (HACM) from Buer et al. (2013) are based on ant colony optimization, however, both heuristics use a more sophisticated encoding strategy on which the ant search graph is based. In contrast to the other heuristics, HACM is not applicable to general DMLULSP instances but to instances with a specific multi-level item structure that enables hierarchical planning. The NBM of Homberger (2010), which is denoted as NBM-H, is based on an SA. The collaborative local search with side payments (CSP) from Homberger et al. (2015) is a local search, which uses encoded solutions and side payments. Thereby, ”side payments compensate one or more agents for accepting solutions with higher local costs in order to find a solution with lower total costs” (Homberger et al., 2015).

## 2.3. Negotiation based mechanism

An efficient automated negotiation mechanism for determining production plans with several software agents was introduced by Klein et al. (2002, 2003a,b,c). This mechanism uses a decentralized SA for determining a suitable production plan. In Homberger (2010), the NBM is used to solve the described MLULSP and DMLULSP. In this section, the mechanism is explained in detail because an extended version of the NBM is applied in this thesis.

First, the idea of the NBM is described in Subsection 2.3.1, where a binary coded production plan (referred to as contract) is used during the negotiation between the agents. In Subsection 2.3.2, it is explained how a production plan can be transferred into a contract and vice versa. The generation of an initial contract and the corresponding updating procedure are addressed in Subsection 2.3.3. In each negotiation round, agents accept a new contract corresponding to the metropolis acceptance criterion of the SA, which is explained in Subsection 2.3.4. A cooling schedule has to be defined for monitoring the acceptance criterion, which is addressed in Subsection 2.3.5. The NBM is explained based on a DMLULSP. In case that the NBM is applied to solve a MLULSP, the set of agents just contains one agent.

### 2.3.1. General overview

The NBM of Klein et al. (2002); Fink (2004); Homberger (2010) can be used to solve MLLS problems. The mechanism is able to solve coordination problems by using contracts. In this case, the term contract can be understood as a mutually accepted production plan of a supply chain. The key elements of the mechanism are the application of a neutral agent (referred to as mediator) and several individualistic agents with an SA acceptance behavior. The mediator supports the negotiation process by generating new contracts and by monitoring the negotiation. During the negotiation, a contract is forwarded to all agents of a supply chain. Then, a voting is performed. In terms of the voting, Klein et al. (2002) identified that agents have an opportunistic behavior. It means that they only accept contracts which reduce their local costs. This would lead to a local optimum. That is why an SA acceptance criterion is used, which enables the possibility that agents accept worse contracts by a specific probability. The goal is the identification of a global optimum.

In Figure 2.6, the procedure of the negotiation protocol is visualized. The approach is initialized by a mediator who generates an initial contract randomly and defines it as best contract $c$. This contract is forwarded to the agents $a_k$ of the supply chain with $k$ agents, who evaluate the contract. The initial contract is accepted by all agents independently from the solution quality. Afterward, the negotiation procedure begins, which is repeated until a certain stop criterion is met. In this procedure, the mediator generates a new contract $c'$ (referred to as proposal) based on the existing contract $c$. Then, the proposal $c'$ is forwarded to all agents, who evaluate the proposal $c'$ corresponding to their local costs. Thereby, each agent performs a voting corresponding to the proposal $c'$. An agent accepts the proposal $c'$ if it leads to lower local costs or if it leads to higher local costs corresponding to a specific probability. If all agents accept a proposal, the proposal $c'$ will be updated as new contract $c$. One negotiation round is preformed after this step. The procedure is repeated until a certain number of negotiation rounds are performed.

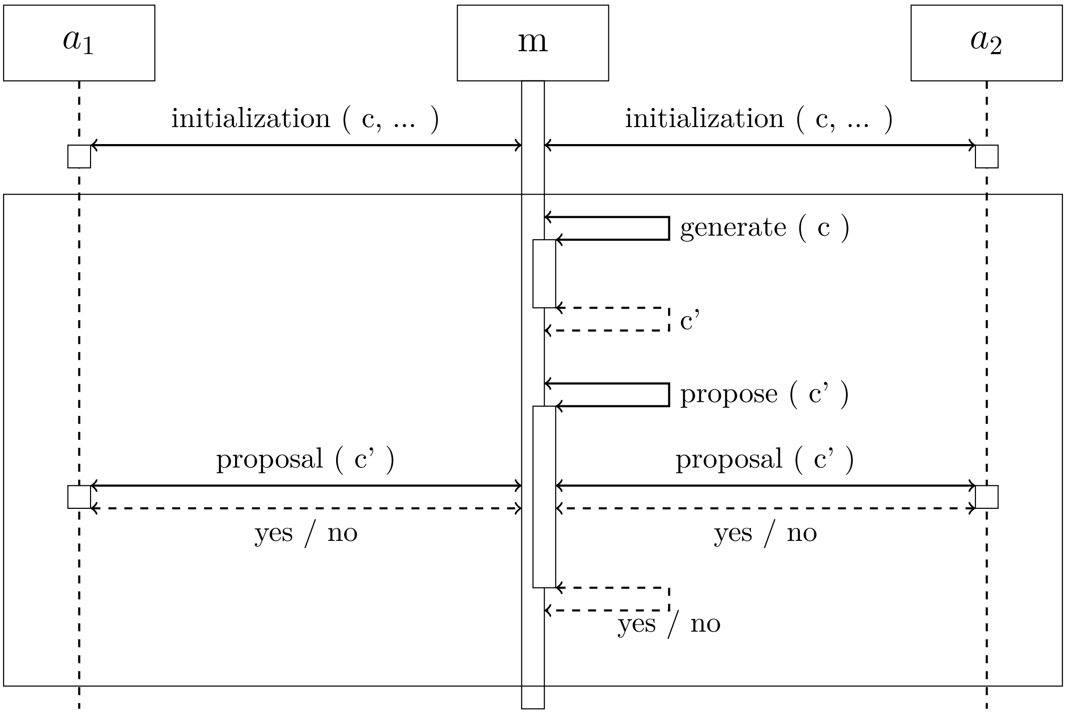


***Figure 2.6.:*** *Negotiation protocol, cf. Fink (2004)*

Klein et al. (2002) identified that the NBM has a good performance in case that all agents act cooperatively by accepting worse proposals. The efficiency of this solution approach depends on the generation of new contracts and the probability of the SA. One disadvantage of this solution approach is that agents, who do not act cooperatively in terms of accepting worse proposals, can benefit from cooperative agents, which could lead to the prisoner's dilemma of game theory. Fink (2004) proposes the integration of the mandatory acceptance criterion for avoiding the prisoner's dilemma, which "forces" agents to behave cooperatively. Furthermore, Fink (2004) suggests to transfer the manual approach of Klein et al. (2002) into an automatic negotiation approach, where software agents adopt the behavior of real agents. In literature, the usage of software agents (referred to as agents) is preferred because it is often cheaper and faster than a manual approach.

In Homberger (2010), the solution approach of Klein et al. (2002) and its extensions described by Fink (2004) is applied to the MLULSP and DMLULSP. In terms of the adaption of this solution approach, Homberger (2010) proposes that a binary coded production plan is used as a contract. The idea of the binary encryption was introduced by Homberger (2006, 2008).

A detailed overview about the NBM is given by Alg. 1. In general, this algorithm is similar to the negotiation protocol in Figure 2.6. The initial contract $c$ is generated randomly by the mediator. Then, each agent $a \in A$ evaluates the contract $c$ with his or her local objective function $f_a(c)$. Therefore, the contract $c$ has to be decoded. The procedure of coding and decoding is discussed in Subsection 2.3.2. As soon as the contract $c$ is evaluated, each agent $a \in A$ executes a trial run for determining agent specific initial temperatures $T_a^0$ and cooling rates $\tau_a$. The concept of the trial run is explained in Subsection 2.3.4.

**Algorithm 1:** NBM, cf. Ziebuhr et al. (2013)

**Data**: problem data, set of agents $A$, number of negotiation rounds $r_{\max}$, mandatory acceptance ratio $P_{\text{init}}$, end temperature $T^{\text{end}}$

1 mediator — negotiation round counter $r \leftarrow 0$
2 mediator — generate initial contract $c$ randomly
3 each $a \in A$ — evaluate contract $c$ with local objective function $f_a(c)$
4 each $a \in A$ — compute initial temperature $T_a^0$ and cooling rate $\tau_a \leftarrow \sqrt[(r_{\max}-1)]{T^{\text{end}}/T_a^0}$
5 **while** $r \leq r_{max}$ **do**
6     mediator — update counter $r \leftarrow r + 1$
7     mediator — generate proposal $c' \leftarrow N(c)$
8     each $a \in A$ — evaluate proposal $c'$ with local objective function $f_a(c')$
9     each $a \in A$ — accept $c'$ with probability $P_a(c, c', r)$
10     **if** *all agents accept proposal* $c'$ **then**
11         mediator — update mutually accepted contract $c \leftarrow c'$
12         each $a \in A$ — update temperature $T_a^r \leftarrow \tau_a \cdot T_a^{r-1}$
13     **end**
14 **end**
15 **return** $c$ mutually accepted contract

Based on the initial contract $c$ and determined cooling schedules, the mediator tries to improve the initial contract $c$ in several negotiation rounds (line 5–line 14 of Alg. 1). In each negotiation round $r$, the mediator is responsible for generating the proposal $c'$ corresponding to the contract $c$. The generation of contracts is addressed in Subsection 2.3.3. Then, the mediator submits the proposal $c'$ to each agent of the coalition $A$. Each agent $a \in A$ evaluates the proposal $c'$ based on his or her local objective function $f_a(c')$. As acceptance criterion the metropolis acceptance criterion is used, which is explained in Subsection 2.3.4. If all agents accept the proposal $c'$, the proposal $c'$ will be used as new mutually accepted contract $c$. At the end of the negotiation round $r$, each agent $a \in A$ updates his or her specific temperatures $T_a^r$ corresponding to the cooling schedule. The cooling schedule is addressed in Subsection 2.3.5. The negotiation procedure is repeated until a certain amount of negotiation rounds $r_{\max}$ are analyzed. Finally, the best contract $c$ is returned.

### 2.3.2. Encoding and decoding of solutions

A solution for the DMLULSP covers the setup decision ($y_{ita}$), the lot size ($x_{ita}$), and the amount of inventory ($l_{ita}$) for each item, each period, and each agent. One issue is that it is often difficult to derive directly feasible solutions for the MLULSP (Dellaert and Jeunet, 2000). That is why Homberger (2010) proposes to use an encoded solution within the NBM. The idea of an encoded solution was originally introduced for a genetic algorithm by Homberger (2006, 2008).

In Homberger (2010), it is proposed to use an encoded solution as production plan, which is defined as contract. The contract $c$ is a $|I| \times |T| \times |A|$ binary matrix, where $c_{ita} = 1$ means that item $i$ might be produced in period $t$ by agent $a$ (i.e., $y_{ita} = 1$) or not (i.e., $y_{ita} = 0$). In contrast, $c_{ita} = 0$ means that a production is not possible (i.e., $y_{ita} = 0$).

To identify a feasible solution for the DMLULSP, a contract has to be decoded with the goal of determining the setup decisions, lot sizes, and inventories. Three decoding steps are performed. These decoding steps are executed separately for each item of the product structure. Thereby, the items are listed in an ascending order of the line numbers. By considering the item $i$ and the responsible agent $a$, the decoding procedure runs through the following steps (Homberger, 2010):

- In Step 1, the demand $d_{ita}$ is calculated for each period $t \in T$ and each agent $a \in A$ according to constraints (2.5). For final items the demand is given endogenously, while the remaining ones have to be determined.
- In Step 2, the setup decision $y_{ita}$ and lot size $x_{ita}$ is calculated for each period $t \in T$ and each agent $a \in A$ corresponding to two scenarios:
  - In Scenario 1, if there is a demand ($d_{ita} \geq 0$) and a production is possible ($c_{ita} = 1$), the setup decision will set to one ($y_{ita} = 1$) and the lot size will set to the demand ($x_{ita} := d_{ita}$).
  - In Scenario 2, if there is a demand ($d_{ita} \geq 0$) and a production is not possible ($c_{ita} = 0$) in period $t$ for agent $a$, the setup decision will set to zero ($y_{ita} = 0$) in period $t$ for agent $a$. In this case, the demand is transferred to a previous period $t'$, where a production is possible ($c_{it'a} = 1$). In period $t'$, where a production is possible, the setup decision is set to one ($y_{it'a} = 1$) and the lot size in period $t'$ is modified by the demand ($x_{it'a} := x_{it'a} + d_{ita}$).
- In Step 3, the amount of inventory $l_{ita}$ is calculated for each period $t \in T$ and each agent $a \in A$ according to constraints (2.2).

The described decoding procedure is repeated until each item $i \in I$ is determined. Finally, each agent $a \in A$ calculates his or her local costs based on the setup decision ($y_{ita}$), lot size ($x_{ita}$), and amount of inventory ($l_{ita}$) for each item and each period. Agent specific setup and inventory holding costs remain private due to this procedure. The decoding procedure is applied in line 3 and line 8 of Alg. 1.

### 2.3.3. Generation of contracts

The NBM uses two procedures in terms of the modification of a contract. The procedure in line 2 of Alg. 1 is responsible for generating an initial contract, while the second one in line 7 of Alg. 1 is responsible for generating a proposal by slightly updating a contract during the negotiation. In both cases, the mediator is in charge of this procedure with the goal to ensure a "fair" negotiation.

An initial contract $c$ is generated randomly. Thereby, with a probability of 50 % the bits of the contract $c$ are set to one and to zero otherwise. In terms of this procedure, there are two exceptions. One exception is that in these periods, before the first demand occurs, a production is not possible. A second exception is that in the first period with a demand, a production is always possible for all items and cannot be changed during the negotiation. This ensures that a contract can always be decoded to a feasible solution.

A new proposal $c'$ is generated by flipping a randomly chosen element from the contract $c$. That is, the neighborhood $N(c)$ of the currently accepted contract $c$ is defined as the set of contracts that can be obtained by flipping a single element of $c$, i.e., if $c_{ita} = 1$ then $c'_{ita} = 0$ and if $c_{ita} = 0$ then $c'_{ita} = 1$. The given graph is a directed graph which is weakly connected. In terms of the contract, it is always ensured that an agent is selected who is able to produce the selected item as well as that a production is always possible in the first period with a demand.

### 2.3.4. Acceptance criterion

Corresponding to Fink (2004), it cannot be recommended to use an acceptance criterion which only accepts improvements of a solution. Otherwise the local search would get stuck in a local optimum. That is why an SA is used for the local search where the metropolis acceptance criterion ensures that improvements and deterioration are accepted.

An SA is a metaheuristic based on local search, which is frequently used to solve discrete optimization problems (Kirkpatrick et al., 1983). In order to escape from a local optimum, SA also allows moves that increase the objective function value (in a minimization problem). Whether an increasing move is performed depends on a probability measure which in turn depends on the objective function value of the solution and in particular the corresponding temperature value. At first, the temperature is set to a high value which decreases (anneals) as the search advances.

In line 9 of Alg. 1, the metropolis acceptance criterion takes place. According to this criterion, each agent $a \in A$ always prefers a proposal $c'$ to the current, mutually accepted contract $c$ if the agent's local costs decrease ($f_a(c') \leq f_a(c)$). Nevertheless, if $c'$ leads to higher local costs ($f_a(c') > f_a(c)$), agent $a$ will prefer $c'$ compared to $c$ with probability $P_a(c, c', r)$ where $r$ is the current negotiation round (Homberger, 2010):

$$P_a(c, c', r) = \begin{cases} 1 & \text{if } f_a(c') \leq f_a(c), \\ \exp\left(\frac{f_a(c) - f_a(c')}{T_a^r}\right) & \text{otherwise.} \end{cases} \tag{2.11}$$

Corresponding to formula (2.11), the probability of accepting deterioration depends on the gap between the local costs with contract $c$ compared to the local costs with proposal $c'$ and the individual temperature value of the agent $a$ in round $r$ ($T_a^r$). Obviously, higher gaps between the local costs and higher temperature values cause a higher probability of accepting deteriorating proposals.

One issue in terms of an SA is the determination of the initial temperature values ($T_a^0$) for each agent $a \in A$ within the coalition. Fink (2004) proposes to use a mandatory acceptation ratio $P_{\text{init}}$, which defines "the number of candidate contracts that shall be accepted by an agent for the beginning phase and final phase of the negotiation process, respectively". The advantage of this approach is that a "fair" acceptance behavior can be ensured, where agents accept worse proposals on the same level independently from their objective function. To determine these initial temperature values, each agent $a \in A$ performs a trial run based on the initial contract $c$. Fink (2004) suggests that one trial run covers 10 % of the maximal number of negotiation rounds ($r_{\max}$). In a DMLULSP with five agents, this would double the number of negotiation rounds ($2 \cdot 10\,\% \cdot 5 = 100\,\%$). That is why Homberger (2010) suggests to use this concept of trial runs just for the determination of the initial temperature values. The end temperature values are determined based on a computational study.

In a trial run, a negotiation of agent $a$ is simulated and the success ($m_1$) and failure ($m_2$) of proposals and the difference between the contract $c$ and the proposal $c'$ for each failure ($\delta^a_{m_2}$) are measured by the mediator. The negotiation process is the same as presented by Alg. 1 except that a proposal is accepted in case a proposal leads to lower local costs and by a specific probability $P^{(n-1)}_{\text{init}}$. The acceptance decision of the remaining agents is simulated by using $P^{(n-1)}_{\text{init}}$. Finally, the initial temperature value $T^0_a$ of agent $a$ is computed by using the following formula (Fink, 2004):

$$P_{\text{init}} = \frac{m_1 + \sum_{i=i}^{m_2} e^{\frac{\delta^a_i}{T^a_0}}}{m_1 + m_2} \tag{2.12}$$

The described procedure is repeated for each agent $a \in A$. Based on the initial temperature values, the agent's cooling schedules can be determined.

### 2.3.5. Agent's cooling schedule

Different cooling schedules are known for updating the individual temperature values ($T^r_a$) of each agent $a \in A$ within an SA (e.g., linear, geometric, and logarithmic cooling schedule). These schedules are necessary to decrease the individual temperature values $T^r_a$ of each agent $a \in A$. Decreasing temperature values imply that the probability to accept a non-improving proposal declines as well during the negotiations. In this mechanism, a geometric cooling schedule is used in line 4 and line 12 of Alg. 1.

In line 12 of Alg. 1, the geometric cooling schedule of (2.13) is used. In that schedule, $\tau_a$ is a constant factor depending on the end temperature $T^{\text{end}}$ to which the agent's individual temperatures $T^r_a$ should anneal (Homberger, 2010):

$$T^r_a = \tau_a \cdot T^{r-1}_a \text{ with } \tau_a = \sqrt[(r_{max}-1)]{\frac{T^{\text{end}}}{T^0_a}} \tag{2.13}$$

In this schedule, $T^{\text{end}}$ and $T^0_a$ are given parameters of the search. To compute appropriate $T^0_a$ values of each agent $a \in A$, the mentioned trial runs are executed. The end temperature values $T^{end}$ are determined based on computational studies.

# 3. Negotiation based Mechanism with Part-Way Resets

In the previous chapter, the NBM of Klein et al. (2002); Fink (2004); Homberger (2010) has been presented. The mechanism is rebuilt from scratch in this chapter. Thereby, it is identified that the solution quality can be improved for the MLULSP and DMLULSP by integrating a part-way reset (PWR) procedure. The extended solution approach is denoted as NBM with part-way resets (NBM-PWR).

Section 3.1 gives an overview about the detailed algorithm of the PWR procedure. The idea of this procedure is to overcome disagreements between agents more easily by resetting the local search once in a while to earlier solutions (i.e., proposals) and discriminating some agents randomly during the negotiation process. A discriminated agent has a higher probability of accepting a proposal which does not decrease his or her local costs. As soon as the idea is described, the algorithm of the PWR procedure is presented, which is divided into two phases: store and reset phase.

Section 3.2 analyzes the performance of the NBM-PWR by several computational studies. In Subsection 3.2.1, the setup of the applied computational studies is presented. There the structure of the instances, evaluation criterion, and parameter setting are discussed. In Subsection 3.2.2, the rebuilt NBM is compared with and without the PWR procedure on a subset of the DMLULSP instances. In Subsection 3.2.3, the NBM-PWR is compared with six state of the art heuristics. This chapter is based on Ziebuhr et al. (2013).

## 3.1. Solution approach

The DMLULSP is a group decision problem, which has to be solved by a set of self-interested and autonomous agents. In this group decision problem, it is usually difficult to negotiate a joint production plan that minimizes the total costs because the agents intend to minimize their local costs (i.e., individual costs). The mentioned NBM can be applied for this problem. There a negotiation is simulated by agents (i.e., software agents), who use the metropolis acceptance criterion for accepting or declining a proposal. By considering a proposal, which leads to higher local costs, the criterion defines the probability of accepting this proposal based on the increase of costs and agent specific temperature values. At the beginning, these temperature values are set to a high value. Afterward, these temperature values are annealed by a cooling schedule during the negotiation. It means that the probability of accepting increasing proposals is much higher at the beginning than at the end of the negotiation.

There is one major issue in terms of the existing NBM with the metropolis acceptance criterion. To fully accept a proposal on the level of the group decision problem, each agent has to accept the proposal. In general, this represents a ”fair” approach because no agent is favored compared to another agent within a coalition. This approach is even supported by using the same mandatory acceptance ratio for each agent within a coalition in order that each agent has the same probability of accepting a worse proposal at the beginning of the negotiation. By rebuilding the NBM and doing some experiments, it is observed that this approach is "fair" but it also leads to worse solutions in accordance to the total costs. This might be explained by the fact that some agents have a much higher financial impact on the total costs than other agents because, e.g., they have more valuable items. However, each agent has the same voting power in the NBM. This voting power is not an issue at the beginning of the NBM because the initial temperature values are pretty high. However, this mechanism often gets stuck in a local optimum after some negotiation rounds because single minor agents veto a globally improving solution. Furthermore, it is observed that the best contract is often found after about 50 %–70 % of the common negotiation rounds in the literature, which means that many negotiation rounds are performed without identifying a better contract.

The PWR procedure is proposed to solve these issues. The idea of the PWR procedure is to change the voting power of the agents within a coalition. A simple solution approach in accordance to this issue would be the identification of the minor agents and reducing their voting power by higher temperature values. However, this leads to a scenario where some agents are favored compared to minor agents. This means a "fair" approach cannot be achieved and minor agents would probably leave the coalition. That is why the PWR procedure is proposed, which weakens the voting power of each agent. Furthermore, it is proposed that instead of one extensive local search the local search is divided into smaller ones with higher annealing rates. Therefore, reset points are installed, where the local search is reset to an earlier stage. A reset means that an earlier stored proposal is used. This procedure allows the local search to explore different areas in the contract space and therefore to use the remaining negotiation rounds more effectively. In addition to a reset of the contract, the temperature values of some agents $A'$ ($A' \subset A$) are raised, which are selected randomly. This implies for all agents in $A'$ that the probability to accept a non-improving proposal increases. From the point of view of an agent in $A'$ this is a clear handicap because his or her veto power is weakened. That is why it is ensured that a rise of a temperature value occurs for all agents in $A$ exactly once during the whole negotiation.

The PWR procedure maintains a certain number of proposals during the local search together with the involved temperature values of each agent $a \in A$. If the local search does not progress in a satisfactory way, the search will be reset to an earlier proposal and continues from there with some modifications of the agents' temperature values. Unlike restart metaheuristics like GRASP (greedy randomized restart procedure, e.g., Resende and Ribeiro (2010)) a new solution is not constructed from scratch but the local search is set back to a previous one. That is why this procedure is denoted as PWR procedure. An overview of the NBM-PWR is given by Alg. 2, which extends the known algorithm of the NBM. Several parameter values are added for the PWR procedure: number of reset points $\gamma$, storage frequency $\rho$, cardinality $k$, temperature increment $\Delta$, and candidate set $C$. These parameter values are determined by a computational study. The mechanism is initialized by activating the store phase. Then, the mechanism runs as known by determining an initial contract and individual cooling schedules. Afterward, the negotiation phase takes place, where the mediator is responsible for generating a proposal and the agents decide on their own if they want to accept or decline a proposal. The PWR procedure with the corresponding phases is implemented within the negotiation phase. First, $\gamma$ reset points are generated in the store phase. Second, $\gamma$ restarts are performed in the reset phase. It is worth mentioning that the store phase has to be completed before the reset phase takes place. Finally, the best mutually accepted contract $c$ is returned.

**Algorithm 2:** NBM-PWR, cf. Ziebuhr et al. (2013)

**Data**: problem data, number of reset points $\gamma$, storage frequencies $\rho$, $\rho_s$, and $\rho_r$, cardinality $k$, temperature increment $\Delta$, candidate set $C$

mediator — **phase** $\leftarrow$ *store*, $\rho \leftarrow \rho_s$
mediator — negotiation round counter $r \leftarrow 0$
mediator — generate initial contract $c$ randomly
each $a \in A$ — evaluate contract $c$ with local objective function $f_a(c)$
each $a \in A$ — compute initial temperature $T_a^0$ and cooling rate $\tau_a \leftarrow \sqrt[(r_{max}-1)]{T^{\text{end}}/T_a^0}$
**while** $r \leq r_{max}$ **do**
  mediator — $r \leftarrow r + 1$
  mediator — generate proposal $c' \leftarrow N(c)$
  each $a \in A$ — evaluate proposal $c'$ with local objective function $f_a(c')$
  each $a \in A$ — accept $c'$ with probability $P_a(c, c', r)$
  **if** *all agents accept proposal* $c'$ **then**
    mediator — update mutually accepted contract $c \leftarrow c'$
    each $a \in A$ — update temperature $T_a^r \leftarrow \tau_a T_a^{r-1}$
  **end**
  **if** $r \mod (\rho \cdot r_{max}) = 0$ **then**
    **if** $|C| = \gamma \wedge$ **phase** $=$ *store* $\wedge$ *deadlock* **then**
      mediator — **phase** $\leftarrow$ *reset*, $\rho \leftarrow \rho_r$
    **end**
    mediator — $c \leftarrow$ `PartwayReset`$(c', T, \Delta, k,$ **phase**$)$
  **end**
**end**
**return** $c$ mutually accepted contract

At the beginning of Alg. 2, an empty candidate set $C$ is initialized, which contains $\gamma$ potential reset points ($\gamma$ is set to five reset points). A reset point $R_j$ ($1 \leq j \leq \gamma$) consists of an encoded solution $c'$ found in negotiation round $r$ together with the values of the temperature parameters used by each agent to reach this solution, i.e., $R_j := (r, c', T_1^r, \ldots, T_{|A|}^r)$. In short, the candidate set of reset points is defined as follows:

$$C = \{R_j\} \text{ with } 1 \leq j \leq \gamma \quad \text{and} \quad R_j = (r, c', T_1^r, \ldots, T_{|A|}^r) \tag{3.1}$$

The PWR procedure consists of the mentioned phases: store and reset. In the store phase (lines 2–4 of Alg. 3), potential reset points are generated and added to the candidate set $C$. According to Alg. 3, every time when $\rho_s \cdot r_{\max}$ negotiation rounds are performed, a reset point $R$ is added to $C$. The maximum number of negotiation rounds $r_{\max}$ and the fraction

$\rho_s$ are external parameters. As soon as $\gamma$ reset points are generated, the store phase is completed and the phase is set to reset and the storage frequency $\rho$ is set to $\rho_r$.

**Algorithm 3:** PWR procedure, cf. Ziebuhr et al. (2013)

**Data**: problem data, candidate set $C$, proposal $c'$, temperatures $T$, temperature increment $\Delta$, cardinality $k$, **phase**

1 **if phase** $=$ *store* **then**
2 | mediator | generate reset point $R \leftarrow (r, c', T_1, \ldots, T)$
3 | mediator | $C \leftarrow C \cup \{R\}$
4 | mediator | **if** $|C| = \gamma$ **then phase** $\leftarrow$ *reset*
5 **else if phase** $=$ *reset* **then**
6 | mediator | select reset point $R \in C$ randomly and update $C \leftarrow C \setminus \{R\}$
7 | mediator | replace current contract $c \leftarrow c(R)$
8 | mediator | select subset $A' \subset A$ with cardinality $k$ randomly
9 | each $a \in A$ | update temperature $T_a \leftarrow T_a(R)$
10 | each $a \in A'$ | increase temperature $T_a \leftarrow T_a \cdot \Delta$
11 | **return** $c$ contract and $T_a$ temperature
12 **end**

In the reset phase (lines 6–11 of Alg. 2), the usage of a reset point is allowed. This phase may only start after $\gamma$ reset points have been generated and the negotiation reaches a deadlock. A deadlock is reached when it is not possible to find a new mutually accepted contract after a defined number of negotiation rounds. Then, the reset phase takes place. First, a reset point $R$ is selected randomly from $C$ and the current contract $c$ is replaced by the proposal of the reset point $c(R)$ and the temperature parameter $T_a^r$ of each agent $a \in A$ in the current round $r$ is reset to $T_a(R)$. Second, the temperature values of $k$ agents in $A$ (set is defined by $A'$) are raised by temperature increment $\Delta$. The probability of these $k$ agents to accept an inferior proposal increases and the chance to escape a local optimum originated by a minority of agents increases as well. In terms of the selection of the reset points and agents, it is always ensured that both selections are taken into account by further selections. It means that a reset point and an agent cannot be selected a second time. After the second phase is started, the search cannot return to the first phase. First, the candidate set $C$ is filled until $\gamma$ reset points are achieved in phase one and then, emptied in phase two as long as the maximum number of negotiation rounds $r_{\max}$ are not completed.

Note, with the applied test instances from the literature, a mutual accepted contract is always found before a reset of the search is performed. In the store phase, the NBM-PWR tries to improve a jointly accepted contract. In the reset phase, resets of the search are allowed and the NBM-PWR may also try to improve a contract by using temporarily not jointly accepted proposals. The application of proposals in this context represents a disadvantage of this approach, which could not be solved. However, it is assumed that this effect can be neglected due to the fact that just five out of 100,000 or 400,000 negotiation rounds are used for generating reset points.

## 3.2. Computational studies

The performance of the PWR procedure is evaluated by using existing MLULSP and DMLULSP instances. In total, three instance classes ($s$, $m$, and $l$) with 408 instances are solved in a benchmark study: 136 MLULSP and 272 DMLULSP instances.

In Subsection 3.2.1, the structure of the instances and setup of the computational studies are described. A rebuilt NBM is compared with the NBM-PWR in Subsection 3.2.2. The idea is to identify the impact of the PWR procedure based on the medium instances ($m$). In Subsection 3.2.3, a benchmark study is performed, where the NBM-PWR is compared with six state of the art heuristics.

### 3.2.1. Setup of computational studies

The MLULSP instance set includes three instance groups, which are denoted as $s1$, $m1$, and $l1$ with a total of 176 instances. Instance group $s1$ ($m1$, $l1$, respectively) contains 96 small (40 medium, 40 large, respectively) instances, in which one agent has to find a production plan. These instance sets can be distinguished regarding the number of items $N$, end items $R$, planning periods $T$, product structure, primary demand, and setup and inventory holding costs. The number of end items $R$ is included in the number of items $N$. An overview of the MLULSP instances is given by Table 3.1. A common feature of these instances is that the parameter $p_{ij}$ is always set to one, while the parameter $t_i$ is set to zero for the instance group $s$ and $m$.

***Table 3.1.:*** *MLULSP instances, cf. Homberger (2010)*

| group | instances | N | R | T | $t_i$ | $p_{i,j}$ | product structure |
|---|---|---|---|---|---|---|---|
| s1 | 96 | 5 | 1 | 12 | 0 | 1 | assembly |
| m1 | 40 | 40, 50 | 1, 3 | 12, 24 | 0 | 1 | assembly and general |
| l1 | 40 | 500 | 1, 5 | 36, 52 | 1 | 1 | assembly and general |

The 96 small instances with five items and twelve planning periods were introduced by Benton and Srivastava (1985); Veral and LaForge (1985); Coleman and McKnew (1991), which contain four product structures, six kinds of primary demands as well as four kinds of setup and inventory holding costs. The 40 medium instances were introduced by Dellaert and Jeunet (2000); Afentakis et al. (1984); Afentakis and Gavish (1986). There one agent produces 40 items with one or three end items over twelve planning periods as well as 50 items with one or three end items over 24 planning periods. This instance set contains four product structures and five kinds of primary demands. The 40 large instances with 500 items (including one to five end items) and up to 52 planning periods were introduced by Dellaert and Jeunet (2000). It is worth mentioning that it is identified in preliminary tests that the NBM-PWR is not powerful enough to compute solutions of very good quality for large instances. That is why the large instance set is skipped in this thesis. A second reason is the high computational effort.

The DMLULSP instances were introduced by Homberger (2010), which are based on the MLULSP instances. The DMLULSP instance set includes six instance groups denoted as

$s2$, $s5$, $m2$, $m5$, $l2$, and $l5$ with a total of 352 instances. Instance group s2 ($m2$, $l2$, respectively) contains 96 small (40 medium, 40 large, respectively) instances, in which two agents have to find a joint production plan. The instance groups $s5$, $m5$, and $l5$ have to be solved by five agents. The difference between the MLULSP and DMLULSP instances is that the set of items $I$ is divided into disjoint item sets. Each of these disjoint sets is assigned to one agent. In 50 % of the DMLULSP instances, the items were assigned uniformly while the other half were assigned non-uniformly to the agents. The remaining data were adopted and not modified.

As evaluation criterion, the gap of the distributed solution $y$ computed by the NBM-PWR for a given instance to the best-known non-distributed solution $y^{\text{bk}}$ from the literature is considered. It means that the solutions of the MLULSP are used as a reference scenario. This criterion was suggested by Dudek and Stadtler (2005, 2007). The percentage gap $\mathcal{G}(y)$ is calculated as follows:

$$\mathcal{G}(y) = \frac{f(y) - f^{\text{nd}}(y^{\text{bk}})}{f^{\text{nd}}(y^{\text{bk}})} \cdot 100\,\% \tag{3.2}$$

The best-known solutions for the MLULSP $f^{\text{nd}}(y^{\text{bk}})$ have been gathered from the literature (Dellaert and Jeunet, 2000; Pitakaso et al., 2007; Homberger, 2008, 2010; Xiao et al., 2011). The idea of using this criterion is that the total costs of an optimal MLULSP solution is a lower bound for the total costs of an optimal solution of the DMLULSP. However, it is unknown if the reported solutions for the medium instances are optimal. It is expected that $\mathcal{G}(y) \geq 100\,\%$, which is however not guaranteed.

Computational studies are executed for determining appropriate parameter values for the number of negotiation rounds $r_{\max}$, mandatory acceptance ratio $P_{\text{init}}$, temperature increment $\Delta$, end temperature $T^{\text{end}}$, and storage frequencies $\rho_s$ and $\rho_r$. Ten instances (id 15-24) are solved for each instance group. The applied parameter values are listed in Table 3.2. It is worth mentioning that better solutions can be generated by higher numbers of negotiation rounds, but the existing heuristics use the same number of negotiation rounds. That is why the number of negotiation rounds is not changed in order to be comparable with the existing heuristics.

***Table 3.2.:*** *Parameter setting of NBM-PWR, cf. Ziebuhr et al. (2013)*

| group | instances | $r_{\max}$ | $P_{\text{init}}$ (%) | $\Delta$ | $T^{\text{end}}$ (Â°C) | $\rho_s$ (%) | $\rho_r$ (%) |
|---|---|---|---|---|---|---|---|
| s1 | 15-24 | $5 \cdot 10^4$ | 45 | 5 | 0.01 | 5 | 0.25 |
| s2 | 15-24 | $5 \cdot 10^4$ | 90 | 7.5 | 0.01 | 5 | 0.25 |
| s5 | 15-24 | $5 \cdot 10^4$ | 98 | 2.5 | 0.01 | 5 | 0.25 |
| m1 | 15-24 | $4 \cdot 10^5$ | 60 | 10 | 0.01 | 5 | 0.25 |
| m2 | 15-24 | $4 \cdot 10^5$ | 45 | 2.5 | 0.01 | 5 | 0.25 |
| m5 | 15-24 | $4 \cdot 10^5$ | 90 | 5 | 0.01 | 5 | 0.25 |

The NBM-PWR was implemented in JAVA (JDK 1.7) and the computational studies were executed on a Windows 7 personal computer with Intel Core i7-2600 processor (3.4 GHz and 16 GB of main memory).

### 3.2.2. Impact of part-way resets

In the first computational study, a rebuilt NBM is compared with the NBM-PWR in order to evaluate the effectiveness of the PWR procedure. The results of this comparison are listed in Table 3.3. The table shows the results for three instance groups $m1$, $m2$, and $m5$. For each instance group, only a subset of ten instances is used and each instance is solved once.

***Table 3.3.:*** *Comparison of NBM with and without PWR procedure, cf. Ziebuhr et al. (2013)*

| group | instances | mean $\mathcal{G}$ (%) | |
|---|---|---|---|
| | | NBM | NBM-PWR |
| m1 | 15-24 | 0.845 | 0.700 |
| m2 | 15-24 | 2.421 | 1.774 |
| m5 | 15-24 | 21.862 | 15.392 |

As can be seen from the two rightmost columns, the PWR extension is able to reduce the gap to the central best-known solution for all three instance groups under the given test conditions. For example, it can be observed that the mean solution values in case of instance group $m1$ are 0.145 % less by using the PWR procedure. The effect reinforces, at least for these instances, if the distributed nature of the problem increases. The achieved cost reduction for instance group $m1$, $m2$, and $m5$ is significant with about 21 %, 27 %, and 30 %, respectively. Several findings can be conducted by analyzing the solutions. First, the idea of using stored proposals and modified temperature values leads to lower total costs. Second, the impact of the temperature increment has a higher impact on the total costs, which is identified by an additional computational study. Third, it is important which reset point $R_j \in C$ and which $k$ agents are selected in the reset phase. For example, it is always preferable if the minor agent is chosen in the first reset round of the NBM-PWR because there are more negotiations rounds left for the improvement of a contract.

Due to the third finding, it is observed that the solution quality fluctuates, because in a "fair" approach these decisions have to be done randomly. It is proposed to use an additional computational study, where each instance is solved four times instead of once. Then, the fluctuation range should be measurable. In Table 3.4, best solutions (NBM-PWR-B), worse solutions (NBM-PWR-W), and mean solutions (NBM-PWR-∅) of the NBM-PWR are presented for small and medium instances.

As can be seen by the results, the fluctuation range is measurable. For example, in terms of the instance group $s1$, the best solutions have a mean $\mathcal{G}$ of 0.021 %, while the worst solutions have a mean $\mathcal{G}$ of 0.087 %. This effect reinforces if the distributed nature

**Table 3.4.:** *Fluctuation range of NBM-PWR*

| group | instances | mean $\mathcal{G}$ (%) | | |
|---|---|---|---|---|
| | | NBM-PWR-B | NBM-PWR-W | NBM-PWR-∅ |
| s1 | 1-96 | 0.021 | 0.087 | 0.053 |
| s2 | 1-96 | 0.156 | 1.165 | 0.583 |
| s5 | 1-96 | 0.663 | 2.788 | 1.679 |
| m1 | 1-40 | 0.530 | 1.171 | 0.865 |
| m2 | 1-40 | 1.129 | 2.326 | 1.688 |
| m5 | 1-40 | 10.973 | 24.927 | 17.828 |

of the problem or the number of items increases. It might be explained by the fact that it is more difficult to identify the minor agent in a scenario with five agents than with three agents. Furthermore, this effect increases in medium instances because there is a much higher unequal distribution of the items compared to small instances. Agents have almost a similar impact on the total costs in case of the small instances.

### 3.2.3. Comparison with state of the art heuristics

In a second computational study, the performance of the NBM-PWR is compared with the mentioned six state of the art heuristics from Subsection 2.2.2. It means that the NBM-PWR is compared with the ES, ACO, CACM, HACM, NBM-H, and CSP.

Table 3.5 shows the mean $\mathcal{G}$ over the 96 instances of group $s2$ and $s5$ as well as 40 instances of group $m2$ and $m5$, respectively. The NBM-H results are extracted from Homberger (2010). In this table, the results of the NBM-PWR-∅ and NBM-PWR-B are presented, which are generated by solving each instance four times. In case of this study, it is worth mentioning that all heuristics use the same number of negotiation rounds; except for the CSP which uses 50 % of the common negotiation rounds for the medium instances.

**Table 3.5.:** *Mean gap $\mathcal{G}$ for NBM-PWR and six state of the art heuristics, cf. Ziebuhr et al. (2013); Homberger et al. (2015)*

| group | $\|A\|$ | mean $\mathcal{G}$ (%) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | ES | NBM-H | ACO | CACM | HACM | CSP | NBM-PWR-B | NBM-PWR-∅ |
| s2 | 2 | 5.11 | 1.42 | - | 2.11 | 4.35 | 0.7 | **0.16** | 0.58 |
| s5 | 5 | 19.54 | 2.32 | - | 9.17 | 9.21 | 4.6 | **0.66** | 1.68 |
| m2 | 2 | 47.23 | 7.11 | 5.73 | 1.98 | 3.29 | 1.5* | **1.13** | 1.69 |
| m5 | 5 | 96.15 | 12.13 | 8.15 | 8.06 | 8.06 | **6.5*** | 10.97 | 17.97 |
| ∅$\mathcal{G}$ | | 42.01 | 5.75 | 6.94 | 5.33 | 6.97 | 3.33 | 3.23 | 5.48 |

*Results are determined by using 200,000 instead of 400,000 negotiation rounds ($r_{max}$)

By considering the results of the last row, where the mean gaps over all instances are listed, it can be observed that the NBM-PWR-B has the best overall performance, while a slightly worse performance is archived by the CSP. It means that the NBM-PWR-B outperforms all other heuristics for instance group $s2$, $s5$, and $m2$. However, the NBM-PWR-B cannot outperform CSP and heuristics based on ant colony optimization on the instance group $m5$ with five agents. Nevertheless, the results indicate that the PWR procedure increases the mean solution quality of the NBM-H because the NBM-PWR-B outperforms the NBM-H on all instance groups ($s2$, $s5$, $m2$, and $m5$).

A second option for evaluating the performance of the NBM-PWR-B is given by analyzing the number of new benchmark solutions. That is why the objective function values, which are computed by the NBM-PWR for each of the 80 medium instances ($m2$ and $m5$), are given in Table 3.6. The second column displays the best-known solutions for the non-distributed MLULSP ($m1$) according to the literature. Columns three to five display the results for the instance group $m2$ and columns six to eight display results for the instance group $m5$. The results were computed by the heuristics CACM, HACM, and NBM-PWR, respectively. The objective function values of the best-known solutions are highlighted in bold. In terms of this computational study, it is worth mentioning that the solutions of the CSP are not published. That is why the solutions of the CACM and HACM are used for this study.

It can be seen from Table 3.6 that the NBM-PWR improves 30 out of 40 best-known solutions for instance group $m2$ and 17 out of 40 best-known solutions for instance group $m5$. In a direct comparison with heuristics CACM and HACM, NBM-PWR outperforms CACM on 30 out of 40 $m2$-instances and on 17 out of 40 $m5$-instances; the NBM-PWR outperforms HACM on 38 out of 40 $m2$-instances and on 19 out of 40 $m5$-instances. All in all, the NBM-PWR seems to be competitive with respect to the obtained solution quality to the state of the art heuristics. While the NBM-PWR seems to be superior on two agent instances, CACM appears to be superior for five agent instances. It is assumed that this is caused by the random selection of the subset of agents whose temperature values are increased (line 8 of Alg. 3) because the chance to increase the temperature of an agent that is actually causing a deadlock decreases with a higher number of involved agents.

With respect to the computational effort of the different heuristics it has to be mentioned that the NBM-PWR is clearly slower than the NBM-H, CACM, and HACM. On average, the NBM-H needs 42 seconds per DMLULSP instance, while the NBM-PWR requires 296 seconds (including multiple solutions per instance). The higher computational effort can mainly be explained by a less efficient implementation of the NBM-PWR. In any case, the PWR procedure itself is undemanding with respect to implementation requirements. However, it should be noted that for each instance the compared heuristics generate the same number of solutions, i.e., the opportunity for identifying a good solution is equal for the applied heuristics.

In this chapter, the known NBM was extended by a PWR procedure. The idea of this procedure is to overcome disagreements between agents more easily by resetting the search once in a while to earlier solutions and discriminating some agents randomly during the negotiation. A discriminated agent has a higher probability of accepting a contract. The extended solution approach outperforms the state of the art heuristics on 30 out of 40 medium instances with two agents and on 17 out of 40 sized instances with five agents. Future research should focus on developing an intelligent approach to adaptively parametrize the PWR procedure – in particular the selection of agents whose temperature values are increased – so that it computes competitive results for larger lot-sizing instances.

***Table 3.6.:*** *Computational results for CACM, HACM, and NBM-PWR, cf. Ziebuhr et al. (2013)*

| | central | m2, $\lvert A\rvert = 2$ agents | | | m5, $\lvert A\rvert = 5$ agents | | |
|---|---|---|---|---|---|---|---|
| no. | best-known | CACM | HACM | NBM-PWR | CACM | HACM | NBM-PWR |
| 1 | 194,571 | 198,654.45 | 201,111.70 | **198,057.95** | 207,655.95 | 208,247.35 | **205,296.95** |
| 2 | 179,762 | **180,144.25** | 187,453.30 | 181,203.85 | **198,732.35** | 201,047.75 | 217,802.15 |
| 3 | 165,110 | **167,159.30** | 171,486.90 | 167,226.90 | **174,637.30** | 174,803.05 | 178,324.55 |
| 4 | 155,938 | **156,130.95** | 157,343.20 | 156,184.20 | **168,125.95** | 169,405.35 | 186,271.70 |
| 5 | 201,226 | **201,436.75** | 205,828.80 | 207,900.75 | **215,672.95** | 217,148.80 | 215,828.85 |
| 6 | 183,219 | **183,316.80** | **183,316.80** | 184,874.00 | **203,307.80** | **203,307.80** | 206,855.15 |
| 7 | 187,790 | 189,432.95 | 192,141.90 | **189,181.85** | 204,519.15 | 206,384.90 | **193,221.55** |
| 8 | 136,462 | 140,267.00 | 143,273.10 | **139,788.15** | 154,127.50 | 155,005.80 | **144,373.50** |
| 9 | 161,304 | 168,115.90 | 171,232.00 | **163,590.55** | **168,236.50** | 171,153.90 | 168,651.05 |
| 10 | 186,597 | 188,307.30 | 188,832.50 | **187,654.10** | 202,756.50 | 204,165.30 | **198,237.60** |
| 11 | 342,916 | 347,533.80 | 357,774.10 | **346,877.90** | 371,034.90 | 371,768.95 | **356,741.00** |
| 12 | 340,686 | 350,363.75 | 356,713.70 | **347,745.95** | 374,680.00 | 377,264.80 | **361,902.80** |
| 13 | 292,908 | 296,308.90 | 297,279.35 | **294,518.80** | 312,728.90 | 312,516.70 | **310,901.15** |
| 14 | 378,845 | **383,953.00** | 393,565.40 | 391,933.55 | **418,806.15** | 421,202.30 | 436,125.55 |
| 15 | 354,919 | 363,276.35 | 365,881.15 | **360,608.60** | **374,513.25** | 374,964.25 | 378,290.50 |
| 16 | 346,313 | 355,522.90 | 362,010.00 | **350,535.25** | **391,312.00** | 392,701.40 | 402,815.15 |
| 17 | 325,212 | **333,256.15** | 340,158.25 | 334,694.05 | 360,186.00 | 362,002.10 | **340,470.85** |
| 18 | 411,997 | 421,956.45 | 422,673.80 | **415,172.85** | 467,870.25 | 470,057.20 | **449,555.65** |
| 19 | 385,939 | 397,906.50 | 403,863.50 | **390,126.90** | 427,922.75 | 429,155.40 | **411,996.60** |
| 20 | 390,194 | **398,145.05** | 406,909.50 | 399,311.90 | 447,712.55 | 450,545.85 | **445,091.95** |
| 21 | 148,004 | 148,697.75 | 153,633.55 | **148,126.10** | **153,985.00** | 154,732.70 | 177,796.00 |
| 22 | 185,161 | 192,901.15 | 195,513.15 | **185,469.25** | **198,846.05** | 200,145.65 | 200,172.70 |
| 23 | 197,695 | 201,034.60 | 202,926.20 | **199,737.70** | **204,575.00** | 204,627.00 | 233,723.85 |
| 24 | 185,542 | 188,634.85 | 190,227.60 | **188,060.20** | 197,505.30 | 198,291.65 | **192,770.20** |
| 25 | 160,693 | 160,924.90 | 161,290.90 | **160,692.90** | 170,716.95 | 172,166.70 | **168,013.80** |
| 26 | 192,157 | 197,132.00 | 197,589.55 | **194,746.90** | 203,624.90 | 205,281.35 | **197,233.05** |
| 27 | 184,358 | 186,187.90 | 187,520.35 | **184,578.25** | **190,854.45** | 191,636.40 | 195,043.70 |
| 28 | 136,757 | 138,717.25 | 139,620.70 | **137,555.90** | **142,564.05** | 143,993.00 | 151,239.40 |
| 29 | 161,457 | 161,967.00 | 161,967.00 | **161,549.00** | **169,077.40** | 169,112.70 | 177858.00 |
| 30 | 166,041 | 169,387.00 | 172,662.30 | **167,803.60** | 181,454.00 | 183,848.25 | **175,226.70** |
| 31 | 344,970 | 353,513.20 | 357,377.20 | **347,226.10** | **373,222.00** | 376,412.35 | 415,677.45 |
| 32 | 289,846 | 294,543.90 | 295,794.25 | **290,675.05** | 309,364.45 | **309,271.30** | 310,046.50 |
| 33 | 352,634 | 369,629.50 | 369,877.00 | **352,902.55** | **384,679.85** | 388,249.85 | 391,213.50 |
| 34 | 337,913 | **344,460.85** | 345,452.15 | 344,659.65 | **362,379.50** | 363,750.50 | 385,450.10 |
| 35 | 356,323 | 363,559.75 | 364,774.25 | **359,197.10** | **385,051.35** | 385,597.30 | 487,519.55 |
| 36 | 319,905 | 327,621.70 | 331,926.50 | **323,251.30** | **342,949.55** | 345,287.60 | 349,326.85 |
| 37 | 411,338 | 433,067.10 | 431,887.60 | **413,189.95** | 459,952.45 | 460,670.15 | **446,384.10** |
| 38 | 366,848 | 377,363.70 | 379,710.05 | **371,951.85** | 396,312.90 | 397,393.40 | **390,989.10** |
| 39 | 401,732 | 416,743.90 | 418,036.95 | **406,274.85** | **430,166.95** | 430,335.75 | 578,394.75 |
| 40 | 305,011 | **306,785.75** | 308,950.90 | 307,842.50 | **318,313.65** | 319,545.80 | 332,210.90 |
| mean | | 268,851.56 | 271,889.68 | **266,316.97** | **285,503.36** | 286,829.96 | 294,126.11 |
| median | | **247,990.33** | 250,811.53 | 249,287.90 | **262,518.70** | 263,210.05 | 271,885.18 |

# 4. Collaborative Multi-Level Uncapacitated Lot-Sizing Problems

In this chapter, the solution approach of Chapter 3 is used to solve collaborative MLULSPs. A collaborative MLULSP generalizes the known DMLULSP by considering competition between manufacturers with respect to the production of some items. That is why it is closer to some requirements of real world supply chains where multiple manufacturers are able to produce the same item (e.g., raw materials and components).

Section 4.1 introduces a distributed uncapacitated lot-sizing problem with rivaling agents (DULR), which fits into the category of collaborative MLULSP. The DULR fits to the category because there some items might be produced by more than one agent (i.e., manufacturer). These items are denoted as exchangeable items. It is proposed to consider a DULR with sole item-production where agents compete for the production of an exchangeable item and the favorable agent gets the whole production volume. In terms of these exchangeable items, it is not decided in advance which agent gets the whole production volume. This decision has to be done by an assignment procedure of the solution approach. Due to the consideration of exchangeable items, additional frictions within coalitions are observed. A side payment procedure is introduced to enhance the stability of a coalition, which distributes cost savings obtained through cooperation of the agents within the coalition. This section is based on Buer et al. (2015).

A DULR with multiple item-production is presented in Section 4.2, where each item is an exchangeable item which might be produced by at least one agent. A multiple item-production is ensured by extending the known formulation of the DULR in terms of new cost parameters. Due to these additional parameters, a multiple item-production is preferred instead of a sole item-production. A similar solution approach like in Section 4.1 is applied to solve a DULR with multiple item-production. The difference is that the known assignment and side payment procedure are updated as well as that the PWR procedure is deleted. The PWR procedure can be deleted, because the updated side payment procedure is used in order that minor agents do not veto against a globally improving solution. Besides the formulation of a DULR with multiple item-production, the goal of this section is to present three different ways of computing side payments and to analyze their impact on the solution quality and computational effort by experiments. This section is based on Eslikizi et al. (2015).

Section 4.3 introduces a DULR with production limitations. Thereby, the DULR of Section 4.2 is modified in order that some items have to be produced by an appointed agent due to contractual obligations. These items are denoted as mandatory items. The solution approach of Section 4.2 is applied to solve this optimization problem. However, as observed in the previous section, this solution approach has performance issues regarding the fluctuation range of the solution quality. That is why the solution approach is modified. In a computational study, it is verified that the modified solution approach is able to achieve better results for the DULR with multiple item-production than the previous one. A second study is applied at the end of this section, which analyzes the impact of mandatory items. This section is based on Ziebuhr et al. (2015).

## 4.1. Lot size planning with sole item-production

A DULR with sole item-production is introduced in this section, where agents compete for the production of an item and the favorable agent gets the whole production volume. The NBM-PWR is applied to solve this optimization problem and extended by two procedures. To evaluate the extended solution approach, two computational studies are carried out. It is identified in one study that the solution approach seems to be appropriate to solve a DULR with sole item-production.

To get a better idea of a DULR with sole item-production, the product structure of this DULR is presented in Figure 4.1. In contrast to the DMLULSP, where each item can be produced by exactly one agent, it is proposed that some items might be produced by more than one agent (i.e., exchangeable items). In this DULR, each agent is still in charge of a disjoint set of items (i.e., non-exchangeable items), but there is also a set of exchangeable items. In Figure 4.1, agent $a_1$ produces item 1 to 20 except for item ten and 18, while agent $a_2$ produces item 21-40 except for item 23, 31, and 36. The mentioned exceptions are exchangeable items. Each of these exchangeable items might be produced by agent $a_1$ or agent $a_2$, while a non-exchangeable item like item twelve can only be produced by agent $a_1$. In terms of an exchangeable item, it is not defined which agent gets the production

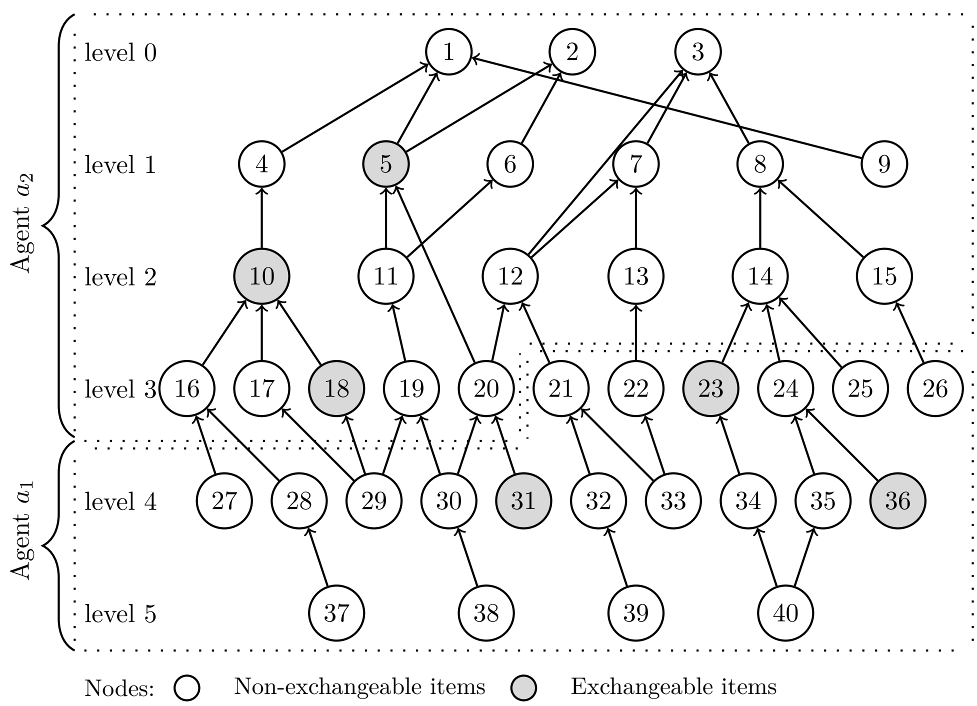


***Figure 4.1.:*** *Product structure of a DULR with sole item-production, cf. Buer et al. (2013)*

volume of this item. This decision has to be done by the solution approach. There are two different goals in terms of a DMLULSP. Obviously, one goal is the determination of these lot sizes and inventories for each non-exchangeable item which minimize the local costs of each agent. However, a cooperative behavior is assumed in case of exchangeable items where agents with the lowest additional costs get the production volume. A stable coalition is enhanced by using side payments. Side payments are used for distributing cost savings regarding exchangeable items.

With respect to an exchangeable item, the question arises how to split the production of an exchangeable item among the agents that are able to produce it? The production quantity of such an item may be split in a specific ratio between rivaling agents (i.e., one agent may substitute another agent) or one of the agents may produce it exclusively. Thanks to the missing capacity constraints of the DULR, the problem narrows down to the question which of the rivaling agents produces an exchangeable item at lowest additional costs over the planning period? It is supposed, that it is often more advantageous that an exchangeable item is produced by a single agent solely than to split the production quantity between multiple agents because in that case the setup costs incur multiple times as well. All in all, this presumption was approved by performing computational studies. That is why a sole-item production is presented in this section, where the agent with the lowest additional costs gets the whole production volume of an exchangeable item.

This section is organized as follows. In Subsection 4.1.1, the mathematical formulation of the DMLULSP is extended to a DULR with sole item-production. Therefore, it is necessary to add new parameters and constraints as well as to modify existing constraints. The NBM-PWR is extended to solve the DULR, which is discussed in Subsection 4.1.2. In Subsection 4.1.3, computational studies are performed, which verify the performance of the solution approach and analyze the side payments based on an example.

### 4.1.1. Mathematical formulation

The mathematical formulation of this DULR is based on a DMLULSP, which is defined in Subsection 2.2.2. In a DMLULSP, it is assumed that each agent $a \in A$ is in charge of a disjoint set of items $I_a$ with $\cup_{a \in A} I_a = I$ and $\cap_{a \in A} I_a = \varnothing$. In a DULR with sole item-production, the intersecting set is not empty $\cap_{a \in A} I_a \neq \varnothing$. It means that in a DMLULSP each item can be produced by exactly one agent, while in a DULR some items might be produced by more than one agent. To take this assumption into account, the set of items $I$ has to be modified for this DULR. Therefore, the set of items $I$ is divided into a set of exchangeable items $I^C$ and non-exchangeable items $I^N$, i.e., $I = I^C \cup I^N$. Non-exchangeable items are items which are produced by one agent only. Furthermore, setup cost $s_{ia}$ and inventory holding cost $h_{ia}$ are extended by a second dimension in order that the costs are defined per item $i \in I$ and agent $a \in A$. The last modification is necessary due to the consideration of exchangeable items for which agents should have different cost rates like in practice.

A new binary decision variable is introduced for the consideration of exchangeable items. The allocation parameter $al_{ia}$ defines whether agent $a$ is assigned to produce item $i$ ($al_{ia} =$

1) or not ($al_{ia} = 0, i \in I, a \in A$). The binary variable $al_{ia}$ is given as an input for a lot-sizing problem similar to the setup decisions otherwise an agent would not produce an exchangeable item in a scenario where the goal is to minimize local costs. Furthermore, similar to the costs let $l_{ita}, d_{ita}, x_{ita}$, and $y_{ita}$ be the inventory, demand, lot size, and setup decision for each item $i \in I$, each period $t \in T$, and each agent $a \in A$. In a DULR with sole item-production, each agent $a \in A$ has to solve the following mathematical model (Buer et al., 2015):

$$\min f_a(y, al) = \sum_{i \in I_a} \sum_{t \in T} (s_{ia} y_{ita} + h_{ia} l_{ita}) \tag{4.1}$$

$$\text{s. t.} \quad l_{ita} = l_{i,t-1,a} + x_{ita} - d_{ita}\,, \qquad \forall i \in I_a, \forall t \in T \tag{4.2}$$

$$l_{i0a} = 0\,, \qquad \forall i \in I_a, \tag{4.3}$$

$$l_{ita} \geq 0\,, \qquad \forall i \in I_a, \forall t \in T \setminus \{0\}, \tag{4.4}$$

$$d_{ita} = al_{ia} \sum_{j \in \tau^+(i)} p_{ij} x_{j,t+t_i,a}, \qquad \forall i \in \{j \in I_a | \tau^+(j) \neq \emptyset\}, \forall t \in T, \tag{4.5}$$

$$0 \geq x_{ita} - M y_{ita}\,, \qquad \forall i \in I_a, \forall t \in T, \tag{4.6}$$

$$x_{ita} \geq 0\,, \qquad \forall i \in I_a, \forall t \in T, \tag{4.7}$$

$$y_{ita} \in \{0, 1\}\,, \qquad \forall i \in I_a, \forall t \in T, \tag{4.8}$$

$$al_{ia} \in \{0, 1\}, \qquad \forall i \in I_a. \tag{4.9}$$

This optimization problem is defined by objective function (4.1) and constraints (4.2)–(4.9), where each agent $a \in A$ tries to minimize his or her local costs $f_a(y, al)$ for his or her set of items. In terms of the constraints, most of the known constraints of the MLULSP are just extended by a third dimension. That is why it is referred to Subsection 2.2.1 for a detailed explanation of these constraints. There are some exceptions. The known constraints (2.5) are modified to constraints (4.5) in order that the demand $d_{ita}$ of an item can be assigned to an agent who is able to produce this item. As mentioned, a sole item-production is considered; that is why the allocation variable is defined as a binary variable by constraints (4.9), which means that an agent produces the whole production volume or nothing. The allocation variable is given in advance and is generated by a mediator, who observes the feasibility of an allocation. An allocation is feasible in case that an agent gets the whole production who is able to produce this item. Thereby, it has to be ensured that $\sum_{a \in A} al_{ia} = 1 \quad \forall i \in I^C$ is fulfilled. In terms of the generation of the allocation variable, the mediator tries to minimize the total costs of the supply chain, which is defined by $f(y, al) = \sum_{a \in A} f_a(y, al)$. By relaxing constraints (4.9) to a continuous variable, it is possible that the production responsibility for an exchangeable item may be split between agents. However, this procedure would still lead to the same result with an exclusive assignment of exchangeable items to agents due to the setup costs.

### 4.1.2. Solution approach

An assignment and side payment procedure is introduced to solve a DULR with sole item-production. The extended solution approach is denoted as NBM-PWR-1 and is explained in this subsection. First, a general overview of the NBM-PWR-1 is presented. Second, the assignment procedure is discussed in detail, which assigns exchangeable items to agents. Third, the side payment procedure is explained, which determines side payments for each agent within a coalition.

**General overview:** In general, the NBM-PWR-1 works similar as the solution approach described in Chapter 3. An overview of the NBM-PWR-1, which is able to deal with exchangeable items, is given by Alg. 4. By initializing the mechanism, the contract $c$ is generated randomly. The contract $c$ is used as an input for the assignment procedure, which is responsible for determining the allocation variable $al_{ia}$ for each exchangeable item $i \in I^C$ and each agent $a \in A$. Based on the determined variable, each agent $a \in A$ evaluates the contract $c$ and determines his or her cooling schedule. Then, the negotiation is initiated, where the mediator presents a new proposal to all agents and each agent evaluates the proposal (line 3 of Alg. 4). In case that the local search reaches a deadlock, it is possible to use one of the mentioned reset points of the PWR procedure. As soon as each negotiation round is analyzed, the best mutually accepted contract $c$ is used as an input for the assignment procedure (line 4 of Alg. 4). The idea of the second application is to optimize the allocation for the final contract. Then, side payments are calculated (line 5 of Alg. 4).

**Algorithm 4:** NBM-PWR-1, cf. Buer et al. (2015)

**Data**: problem data

| | | |
|---|---|---|
| 1 | mediator | generate initial contract $c$ randomly |
| 2 | mediator | $al \leftarrow$ `assign`($c$) |
| 3 | mediator+$a \in A$ | joint contract $c \leftarrow$ `NBM-PWR`($al$) |
| 4 | mediator | $al \leftarrow$ `assign`($c$) |
| 5 | mediator+$a \in A$ | compute side payments $\leftarrow$ `shapley`($A, v$) |
| 6 | **return** $c$ mutually accepted contract and $al$ allocation variable | |

**Assignment procedure:** An assignment procedure is used in order to ensure that one agent gets the whole production volume of an exchangeable item. This procedure is applied twice by Alg. 4. Once it is applied at the beginning of the mechanism and a second time at the end of the mechanism. Before the assignment procedure is applied, it is assumed that the initial allocation variable $al_{ia}$ is set to one in case that agent $a$ can produce item $i$ or zero otherwise. It means that the assignment procedure has to be applied once in order to generate an initial solution and it is applied a second time for the final mutually accepted contract. The assignment procedure is outlined in Alg. 5, which assigns values to the allocation variables $al_{ia}$ $(i \in I^C, a \in A)$.

The idea of this procedure is that each agent, who is able to produce a selected exchangeable item, determines his or her local costs for item $i$. Then, the mediator assigns the production volume of an exchangeable item to this agent who has lower additional costs. For each exchangeable item $i \in I^C$, the procedure begins by identifying the rivaling agents $A^i$, who are able to produce item $i$. Then, one agent $a_1 \in A^i$ is selected randomly. Based on these identified parameters, for each agent $a \in A^i$ the local costs are computed for the case that another agent $a \in A^i, a \neq a_1$ is solely responsible to produce item $i$

**Algorithm 5:** assign - Assignment procedure, cf. Buer et al. (2015)

**Data**: problem data, contract $c$

1 **for** $i \in I^C$ **do**<br>
2     mediator    identify rivaling agents $A^i$<br>
3     mediator    select agent $a_1 \in A^i$ randomly<br>
4     **for** $a \in A^i$ **do**<br>
5         **if** $a \neq a_1$ **then**<br>
6             mediator    update allocation $al_{ia} \leftarrow 1$ and $al_{ia_1} \leftarrow 0$<br>
7             mediator    store costs $\Omega_{ia} \leftarrow f(c, al)$<br>
8             mediator    update allocation $al_{ia} \leftarrow 0$ and $al_{ia_1} \leftarrow 1$<br>
9         **else**<br>
10             mediator    store costs $\Omega_{ia} \leftarrow f(c, al)$<br>
11         **end**<br>
12     **end**<br>
13 **end**<br>
14 **for** $i \in I^C$ **do**<br>
15     mediator    identify agent with lowest costs $a_1 = \arg\min_{a \in A^i} \Omega_{ia}$<br>
16     mediator    update allocation $al_{ia_1} \leftarrow 1$ and $al_{ia} \leftarrow 0$ for all $a \in A^i \setminus \{a_1\}$<br>
17 **end**<br>
18 **return** $al_{ia}$ allocation variable per item and agent

(i.e., $al_{ia} = 1$ and $\sum_{a \in A} al_{ia} = 1$). The resulting total costs are maintained in the matrix $(\Omega_{ia})^{|I^C| \times |A|}$. Finally, the actual assignment is made (lines 14–17 of Alg. 5). Item $i$ is produced solely by that agent $a_1 \in A^i$ who caused the lowest local cost value according to $\Omega_{ia}$ for $i$, i.e., $al_{ia_1} \leftarrow 1$ and $al_{ia} \leftarrow 0$ for all other rivaling agents ($a \in A^i \setminus \{a_1\}$). As a result of this algorithm, the determined allocation parameter $al_{ia}$ per item and agent is returned.

It is obvious that a contract has to be given as input for the assignment procedure. In terms of the contract, there is one modification compared to Chapter 3. In each negotiation round $r$, the current contract $c$ is usually updated by randomly choosing one item $i'$ out of $c$ and flipping a randomly chosen period $t'$ for the given item $i'$. Here, a contract $c$ has three dimensions, where it is specified if an agent $a \in A$ is allowed to produce a specific item $i \in I$ in a given period $t \in T$ or not. In this case, there are many elements of the contract space which have no impact on the solution. Since it does not matter if the contract element of an agent, who is not able to produce a selected item, is set to one or zero. These elements are denoted as infeasible elements. Here, infeasible elements are skipped in terms of the updating procedure of a contract. Several negotiation rounds would be used insufficiently without this modification.

A drawback of the assignment procedure is that the results strongly depend on the initial contract. In general, it is suggested to execute this procedure in each negotiation round, but this leads to a much higher computational effort without a visible improvement. That is why it is just applied two times. An advantage, however, is that agents are not required to reveal their setup and inventory holding cost rates, which are usually considered as private information.

**Side payment procedure:** It is obvious that the decision of which agent should produce which exchangeable item may involve frictions in the DULR. To overcome frictions and to enhance the stability of a coalition of agents, it is proposed to distribute the achieved cost savings by using side payments such that each agent is motivated to share sensitive data regarding exchangeable items. The determination of side payments is part of the game theory. In general, this discipline is used to solve economic decision problems with conflicting interests and coordination issues. Thereby, two research fields are distinguished: cooperative and uncooperative game theory. In cooperative game theory, it is assumed that there is a cooperative behavior among the agents and the focus is on the outcome of a cooperation instead of the strategies among the agents like in uncooperative game theory. (Wiese, 2005, p. 8ff)

In cooperative game theory, an optimization problem is modeled as a game with transferable payoff, which is a pair $(A, v)$, where $A$ represents the set of agents and $v$ represents the characteristic function. A transferable payoff means that there is an exchange object, which can be exchanged from one agent to another without transaction costs, where it is assumed that the transferable payoff of one agent is directly transferred to another agent without any losses. In the terminology of cooperative game theory, the agents which are involved in the lot-sizing supply chain game at hand are the players. The set $A$ of all agents is called coalition and each subset $S$ of $A$ is denoted as sub coalition, while the characteristic function $v(S)$ specifies the payoff that the agents of the sub coalition $S$ can obtain by cooperating with each other. (Hiller, 2011; Rothe et al., 2012, p. 8ff, p. 95)

It seems natural to associate the characteristic function with the costs saved due to cooperating within a coalition. This (standard) interpretation may lead to drawbacks for this DULR. For practical applications of the lot-sizing game, it is usually only possible to find a feasible production plan if all agents cooperate. Only one missing agent may invalidate a production plan. Consider an example. Two agents $a_1$ and $a_2$ produce three items $i1$, $i2$, and $i3$; each item has a positive demand. Item $i1$ may be produced only by $a_1$ and item $i2$ only by $a_2$. Item $i3$ is an exchangeable item, which may be either produced by $a_1$ or $a_2$. Furthermore, item $i3$ requires item $i1$ and $i2$ as input. Then, only the coalition $\{a_1, a_2\}$ is able to generate a feasible production plan. The sub coalitions $\{\emptyset\}, \{a_1\}$, and $\{a_2\}$ cannot satisfy the demand on their own. Therefore, the cost saving function of every sub coalition will be zero, which leads to identical side payments for each player.

The lot-sizing game is interpreted differently to overcome this issue with infeasible production plans. Thorough game-theoretic analysis of lot-sizing and production games are given by Guardiola et al. (2009) or Drechsel and Kimms (2011). However, they do not focus on asymmetric information and mediator based negotiations to find feasible lot-sizing contracts. In this thesis, the players of the game are still the agents and it is focused on the set of exchangeable items $I^C$, where it is still possible to generate a feasible production plan if one agent is missing. Therefore, it is assumed that for each sub coalition $S \subseteq A$ it is always possible to produce all exchangeable $I^C$ and non-exchangeable items $I^N$ even when the respective agent of an item is not part of the sub coalition $S$. Let $S = \{a_1\}$ be a sub coalition from the example. Then, the sub coalition $S$ is able to produce item $i1$, $i2$, and $i3$, where item $i3$ as exchangeable item is produced by agent $a_1$ and item $i2$ as non-exchangeable item is produced by an external manufacturer with the same production costs as agent $a_2$. Hence, in this interpretation each sub coalition $S \subseteq A$ is always able to produce all items and in case that both agents are available, which can produce a certain exchangeable item, this agent who has lower additional costs gets the production volume.

To determine side payments, the well-known Shapley formula is used (Shapley, 1953). The formula ensures a unique distribution among the agents. Four assumptions have to be fulfilled in order to use this approach: (1) the number of agents is limited, (2) a cooperation among these agents is possible, (3) a transferable payoff exists, and (4) the characteristic function represents a suitable image of the reality. Based on these assumptions, the Shapley value determines the mean marginal contributions to all permutations in which the coalition can be formed. Thereby, it is assumed that each permutation has the same probability. A permutation of the agents in $A$ is described as follows: $(p_1, p_2, ..., p_b)$. The term $p_1$ represents the first agent of the permutation $p$ with in total $b!$ different permutations of the agents. The set of permutations is defined by $P$. (Hiller, 2011, p. 16ff.) Corresponding to the Shapley formula, it is defined that the marginal contribution $MC_a^p(v)$ of agent $a$ in permutation $p$ to sub coalition $K_a(p)$ can be calculated as follows (Hiller, 2011, p. 17):

$$MC_a^p(v) = v((K_a(p)) - v(K_a(p)\backslash\{a\}) \tag{4.10}$$

$K_a(p)\backslash\{a\}$ is defined as the set of agents proceeded by agent $a$ in the permutation $p$. By assuming that $A = \{a_1, a_2, a_3\}$ and $p = \{a_3, a_2, a_1\}$, it is obvious that $K_{a_2}(p) = \{a_3, a_2\}$ and $K_{a_2}(p)\backslash\{a_2\} = \{a_3\}$. Here, $v((K_a(p))$ is the minimum total costs that sub coalition $K_a(p)$ achieves. Several assumptions are considered in terms of the characteristic function. An additional agent may contribute additional exchangeable items and therefore reduces the total costs. It means that an additional agent will not increase the total costs, i.e., $v(K_a(p)\backslash\{a\}) \leq v(K_a(p))$. This holds, as long as the problem is solved to optimality. However, the applied approach for this DULR cannot ensure to find solutions that are optimal. Taking this into account, the side payments $Sh_a(A, v)$ for an agent $a \in A$ are calculated according to the Shapley formula (Hiller, 2011, p. 18):

$$Sh_a(A, v) = \frac{1}{b!} \sum_{p \varepsilon P} MC_a^p(v) \tag{4.11}$$

The mentioned formulas are embedded in Alg. 6. There the side payment procedure is presented, which is able to determine the Shapley values. This procedure is executed for each agent $a \in A$ separately.

As input of the side payment procedure, the contract $c$, the selected agent $a'$, and the allocation $al_{ia}$ have to be given. Before the Shapley value of a selected agent $a'$ can be calculated, all possible permutations $P$ of the set of agents $A$ have to be determined. For each permutation $p \in P$, the marginal contribution $MC_{a'}^p(v)$ of agent $a'$ in permutation $p$

**Algorithm 6:** shapley - Side payment procedure, cf. Buer et al. (2015)

**Data**: problem data, contract $c$, agent $a'$, allocation $al_{ia'}$

mediator  determine permutations $P$ of agent $a'$
**for** $p \in P$ **do**
  mediator  update allocation $al_{ia'}$ based on set $K_{a'}(p)$
  **for** $a \in K_{a'}(p)$ **do**
    agent $a \in A$  determine characteristic function $v((K_{a'}(p))$
  **end**
  mediator  update allocation $al_{ia'}$ based on set $K_{a'}(p) \backslash \{a'\}$
  **for** $a \in K_{a'}(p) \backslash \{a'\}$ **do**
    agent $a \in A$  determine characteristic function $v(K_{a'}(p) \backslash \{a'\})$
  **end**
  mediator  determine marginal contribution $MC^{p}_{a'}(v) = v((K_{a'}(p)) - v(K_{a'}(p) \backslash \{a'\})$
**end**
mediator  determine Shapley value $Sh_{a'}(A, v) = \frac{1}{b!} \sum_{p \varepsilon P} MC^{p}_{a'}(v)$;
**return** $Sh_{a'}(A, v)$ Shapley value for agent $a'$

to sub coalition $K_a(p)$ is determined by calculating the characteristic functions $v(K_{a'}(p))$ and $v(K_{a'}(p) \backslash \{a'\})$ and using the mentioned formula (4.10). For the determination of the characteristic function of both sub coalitions, it is proposed to update the allocation parameter $al_{ia}$ in order that for each exchangeable item $i \in I^C$ the agent $a$ with the lowest additional costs gets the whole production volume ($al_{ia} = 1$). As soon as each marginal contribution is determined, the Shapley value $Sh_{a'}(A, v)$ can be calculated by the mentioned formula (4.11). This procedure is repeated for each agent $a \in A$. Finally, the individual side payments are returned and transferred to the agents within the coalition.

### 4.1.3. Computational studies

The NBM-PWR-1 is used to solve a DULR with sole item-production. Since this DULR is not known in the literature, new test instances are generated. The instances are based on existing DMLULSP instances from Homberger (2010). First, the setup of the computational study and the generation of test instances are described. Second, the solution approach is evaluated against well-defined reference values. Third, some of the identified side payments are considered in detail.

To be suitable for this DULR, the DMLULSP instances of Subsection 3.2.1 are extended, where each item is originally assigned to exactly one agent. One main assumption of the generated DULR instances is that it is always ensured that two agents compete with each other regarding the production of an exchangeable item. The DMLULSP instances have to be extended in terms of the definition of exchangeable items and new setup and inventory holding costs. First, the number of exchangeable items $|I^C|$ per instance has to be defined, which is set to one exchangeable item per agent for instance group $s2$ and $s5$ and four exchangeable items per agent for instance group $m2$ and $m5$. Second, it is decided for each item $i \in I$ randomly if it is an exchangeable item ($i \in I^C$) or not ($i \in I^N$). Third, the rivaling agents are determined for each exchangeable item. The selection process is executed randomly and it is ensured that each agent $a \in A$ receives the same amount of exchangeable items. However, sometimes it occurs that one agent is in charge of so many items that the remaining agents of the sub coalitions do not have enough items to produce his or her proposed amount of exchangeable items. In this case, the restricted agent defines the amount of exchangeable items per agent. Fourth, new cost parameters have to be added. Thereby, it is ensured for each exchangeable item $i \in I^C$ that one rivaling agent gets the same cost parameters as in the DMLULSP. For the additional rivaling agent, the inventory holding cost $h_{ia}$ as well as the setup cost $s_{ia}$ are chosen randomly between 80 % and 120 % of the original value in the corresponding DMLULSP instance. It means that the original cost parameters are available for each item in any case. With this setting, it can be inferred that optimal solutions for a DULR with sole item-production have lower or at least equal objective function values compared to optimal solutions for the DMLULSP (i.e., without exchangeable items). Due to this procedure, solutions for DULR should have lower total costs than the corresponding solutions of DMLULSP which is why the objective function values of the best-known solutions for DMLULSP can be used as reference values.

The NBM-PWR-1 uses the same parameter values as proposed in Subsection 3.2.1. As evaluation criterion the percentaged gap $\mathcal{G}(y)$ is used, where the DULR solution $y$ computed by the NBM-PWR-1 is compared to the solution for the DMLULSP $y^{bk}$ computed by the NBM-PWR of Chapter 3. The NBM-PWR-1 was implemented in JAVA (JDK 1.7) and the computational studies were executed on a Windows 7 personal computer with Intel Core i7-2600 processor (3.4 GHz and 16 GB of main memory).

In the first computational study, the performance of the NBM-PWR-1 is evaluated by comparing DULR solutions with DMLULSP solutions. The aggregated results are listed in Table 4.1.

***Table 4.1.:*** *Comparison of mean total costs of DMLULSP and DULR solutions, cf. Buer et al. (2015)*

| group | instances | $\lvert I^C \rvert$ | DMLULSP | DULR | outperform (or equal) |
|---|---|---|---|---|---|
| s2 | 1-96 | 2 | 812.19 | 802.01 | 76 out of 96 |
| s5 | 1-96 | 5 | 816.08 | 788.06 | 89 out of 96 |
| m2 | 1-40 | 8 | 266,316.97 | 270,301.08 | 19 out of 40 |
| m5 | 1-40 | 20 | 294,126.11 | 292,704.11 | 22 out of 40 |

The results of the table indicate that the NBM-PWR-1 is equal or even outperforms the reference value in 206 out of 272 cases. Especially the results for the small instances ($s2$ and $s5$) are rather promising, which means that this solution approach is able to unlock the additional potential to decrease the total costs due to exchangeable items. The solution approach computes superior results for 76 out of 96 small instances with two agents ($s2$). The NBM-PWR-1 is even better for instances with five agents, i.e., the solution approach outperforms the reference value in 89 out of 96 instances. However, the performance of the mechanism is inferior for medium instances ($m2$ and $m5$). Although the mean total costs over the $m2$ and $m5$ instances are lower than the reference value, the mechanism is only able to unlock the cost savings potential for about half of the tested instances. In terms of the computational effort, both solution approaches perform similar.

The detailed results for the medium instances are listed in Table 4.2. The solution values with lower costs are marked in bold. As can be seen from the last two rows, the NBM-PWR-1 is able to identify DULR solutions with lower costs than DMLULSP solutions by about 50 % of the instances.

***Table 4.2.:*** *Detailed results per instance of group m2 and m5 by NBM-PWR-1, cf. Buer et al. (2015)*

| id | m2, \|A\| = 2 agents | | m5, \|A\| = 5 agents | |
|---|---|---|---|---|
| | DMLULSP | DULR | DMLULSP | DULR |
| 1 | 198,057.95 | **197,368.71** | 205,296.95 | **202,802.46** |
| 2 | 181,203.85 | **181,140.68** | **217,802.15** | 220,219.18 |
| 3 | 167,226.90 | **166,008.26** | **178,324.55** | 185,717.64 |
| 4 | 156,184.20 | **155,239.85** | 186,271.70 | **184,359.05** |
| 5 | 207,900.75 | **207,161.45** | 215,828.85 | **206,922.01** |
| 6 | **184,874.00** | 188,497.23 | 206,855.15 | **196,404.72** |
| 7 | **189,181.85** | 191,204.32 | 193,221.55 | **193,078.34** |
| 8 | 139,788.15 | **139,408.91** | 144,373.50 | **140,300.28** |
| 9 | **163,590.55** | 164,862.05 | **168,651.05** | 170,319.84 |
| 10 | **187,654.10** | 190,328.40 | **198,237.60** | 201,311.25 |
| 11 | 346,877.90 | **346,680.69** | **356,741.00** | 375,320.46 |
| 12 | **347,745.95** | 354,693.60 | **361,902.80** | 409,600.58 |
| 13 | **294,518.80** | 378,403.69 | 310,901.15 | **307,586.13** |
| 14 | **391,933.55** | 396,485.96 | **436,125.55** | 460,424.00 |
| 15 | 360,608.60 | **357,300.25** | 378,290.50 | **361,254.29** |
| 16 | **350,535.25** | 351,276.42 | **402,815.15** | 456,951.21 |
| 17 | 334,694.05 | **327,583.31** | **340,470.85** | 360,591.84 |
| 18 | **415,172.85** | 429,630.06 | **449,555.65** | 479,436.73 |
| 19 | 390,126.90 | **382,407.54** | **411,996.60** | 414,012.94 |
| 20 | **399,311.90** | 404,949.60 | 445,091.95 | **428,093.89** |
| 21 | 148,126.10 | **146,211.94** | 177,796.00 | **159,298.11** |
| 22 | 185,469.25 | **184,313.61** | 200,172.70 | **190,224.38** |
| 23 | **199,737.70** | 207,901.44 | 233,723.85 | **221,910.27** |
| 24 | 188,060.20 | **186,707.36** | 192,770.20 | **192,282.44** |
| 25 | 160,692.90 | **159,988.80** | **168,013.80** | 181,966.93 |
| 26 | 194,746.90 | **193,369.82** | **197,233.05** | 199,857.89 |
| 27 | **184,578.25** | 195,644.73 | **195,043.70** | 203,470.39 |
| 28 | 137,555.90 | **136,484.44** | 151,239.40 | **146,640.84** |
| 29 | **161,549.00** | 164,200.95 | 177,858.00 | **170,897.59** |
| 30 | 167,803.60 | **167,712.08** | **175,226.70** | 182,215.80 |
| 31 | **347,226.10** | 360,326.25 | **415,677.45** | 431,203.04 |
| 32 | **290,675.05** | 295,544.84 | 310,046.50 | **301,178.97** |
| 33 | **352,902.55** | 353,948.41 | 391,213.50 | **384,147.95** |
| 34 | **344,659.65** | 352,606.90 | 385,450.10 | **381,972.47** |
| 35 | **359,197.10** | 364,700.12 | 487,519.55 | **396,615.25** |
| 36 | 323,251.30 | **321,664.80** | 349,326.85 | **340,422.74** |
| 37 | **413,189.95** | 422,141.67 | **446,384.10** | 470,352.38 |
| 38 | 371,951.85 | **370,447.33** | 390,989.10 | **390,754.94** |
| 39 | **406,274.85** | 408,241.68 | 578,394.75 | **442,206.49** |
| 40 | **307,842.50** | 309,255.01 | **332,210.90** | 365,838.67 |
| mean | **266,316.97** | 270,301.08 | 294,126.11 | **292,704.11** |
| outperform | **21** | 19 | 18 | **22** |

In the absence of benchmark results for the problem at hand, the performance of the solution approach for small instances seems to be appropriate, while the performance for medium instances offers room for improvement. This means that it is not necessary to apply the assignment procedure more than twice for the small instances. However, in terms of the results for medium instances, the current process of assigning production responsibilities for items to agents (cf. Alg. 5) prior to the actual negotiation phase of Alg. 4 and prior to the determination of side payments (cf. Alg. 6) seems to be inefficient.

That is why it is proposed to analyze different points in time for reassigning the production responsibilities during the negotiation phase. The assignment of exchangeable items to agents is recomputed in: (1) every round, (2) every 500 rounds, and (3) every 10,000 rounds. In these experiments, it is identified that none of these changes could improve the solution quality significantly. Each time when the assignment is recomputed, the previous modifications of the contract are often useless due to the fact that the contract is modified for a different assignment. Another disadvantage is that the computational effort increases significantly with the determination of a new assignment, because this determination has a similar computational effort like one negotiation round.

The Shapley value and a modified characteristic function for this DULR are used to compute distributions of the saved costs to the agents. The goal of this procedure is to ensure a fair distribution of the saved costs. In a second computational study, the side payments are analyzed. Table 4.3 exemplary presents side payments for four instances (no. 25 of groups $s2$, $s5$, $m2$, and $m5$). The value $v(A)$ (2nd and 5th column) indicates the achieved cost savings due to the existence of exchangeable items within the coalition. The data in the marked cells can be interpreted as follows. If no side payments are calculated (first line), only agent $a_1$ will benefit from the cost savings of 29.5 monetary units achieved by the coalition. The reason is that agent $a_2$ produces the exchangeable items at lower costs than agent 1. Without the exchangeable items of agent $a_2$, the total costs would be 29.5 units higher. Due to the side payments calculated by the Shapley formula (second line), agent $a_2$ receives 14.8 units due to his or her ability to produce the exchangeable items more efficiently than other agents within the coalition. As can be seen, the side payments lead to a scenario where each agent receives the same portion of cost savings.

***Table 4.3.:*** *Savings distribution to agents with and without Shapley payments, cf. Buer et al. (2015)*

| type | \|A\| = 2 agents | | | \|A\| = 5 agents | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $v(A)$ | $Sh_{a_1}$ | $Sh_{a_2}$ | $v(A)$ | $Sh_{a_1}$ | $Sh_{a_2}$ | $Sh_{a_3}$ | $Sh_{a_4}$ | $Sh_{a_5}$ |
| s | 29.5 | 29.5 | 0.0 | 40.1 | 0.0 | 0.0 | 11.6 | 28.5 | 0.0 |
| s, (Shapley) | 29.5 | 14.8 | 14.8 | 40.1 | 1.3 | 14.3 | 5.8 | 14.3 | 4.5 |
| m | 2,941.0 | 1,251.0 | 1,704.0 | 7,695.7 | 1,822.4 | 2,197.4 | 430.0 | 1,343.3 | 1,902.6 |
| m, (Shapley) | 2,941.0 | 1,470.5 | 1,470.5 | 7,695.7 | 2,448.4 | 1,098.7 | 1,592.4 | 892.6 | 1,663.6 |

In conclusion, a DULR with sole item-production is presented in this section in which some agents are able to produce the same items and are therefore rivals. Previous distributed lot sizing models in this line of research assume that each agent produces different items, i.e., there is a disjoint allocation of items to agents. The DULR is solved by the NBM-PWR, which was introduced in Chapter 3. Therefore, the NBM-PWR is extended by an assignment and side payment procedure. Two computational studies are carried out. First, a study is performed which analyzes the performance of the NBM-PWR-1 for a DULR with sole item-production compared to the DMLULSP. The computational results indicate that the NBM-PWR-1 is able to outperform the reference value in almost all of the small instances and in about half of the medium instances. For future research, it appears promising to improve the performance of the negotiation mechanism by developing a more sophisticated assignment procedure and to develop an approach with a multiple item-production instead of a sole item-production.

## 4.2. Lot size planning with multiple item-production

In this section, a DULR with multiple item-production is introduced, where agents compete regarding the production of an item. The NBM-PWR-1 without the PWR procedure is applied to solve this optimization problem. The mechanism is modified in terms of the two procedures: assignment and side payment procedure.

To get a better idea of a DULR with multiple item-production, the corresponding product structure is presented in Figure 4.2.

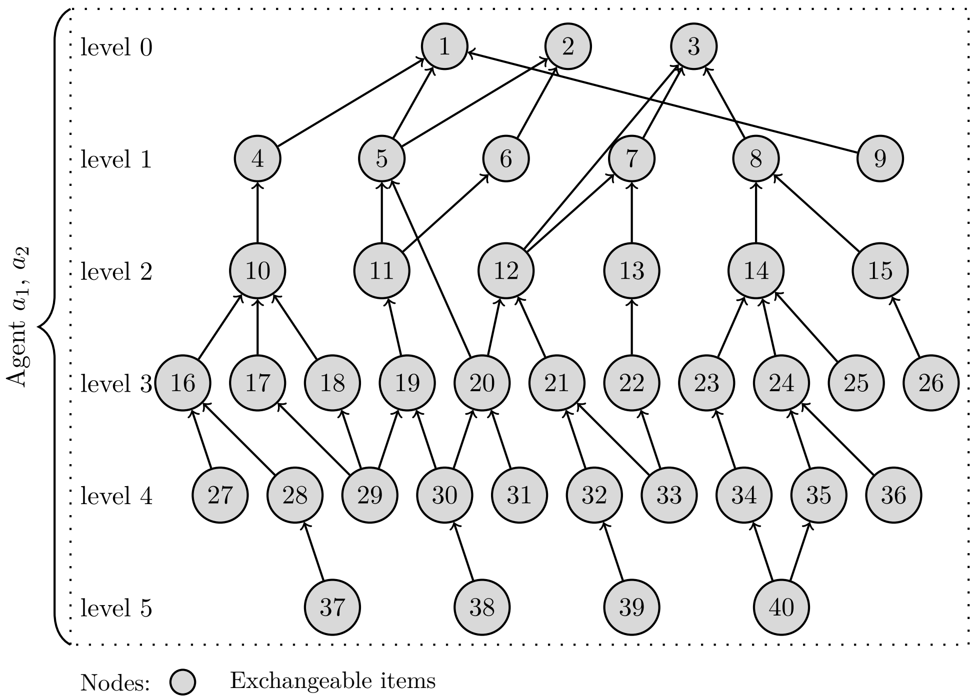


***Figure 4.2.:*** *Product structure of a DULR with multiple item-production, cf. Buer et al. (2013)*

In the considered lot-sizing problem, it is assumed that all items are exchangeable items (i.e., items might be produced by more than one agent). In Figure 4.2, agent $a_1$ and $a_2$ are able to produce each item of the product structure. They compete with each other regarding production quotas of these items instead of just rivaling in terms of the whole production volume like in the previous section. The decision, which agent produces a specific quota of an item, is made during the solution approach. A cooperative behavior is still assumed to solve this optimization problem, where each agent within the coalition tries to reduce his or her local costs by also observing that the total costs of the supply chain are reduced.

The section is organized as follows. In Subsection 4.2.1, the mathematical of a DULR with multiple item-production is presented, which is solved by a modified NBM-PWR-1 without PWR. The mechanism is presented in Subsection 4.2.2. Finally, computational studies are presented in Subsection 4.2.3.

### 4.2.1. Mathematical formulation

The formulation of a DULR with sole item-production in Subsection 4.1.1 is used and extended in this subsection. In terms of the previous formulation, one of the main issues is that a sole item-production leads to lower costs than a multiple item-production. A reason for this observation is that the reduced inventory holding costs cannot compensate the additional setup costs when the production of an exchangeable item is executed by more than one agent. Further issues are that there are no capacity issues, which means that each agent can produce an unlimited amount of items and that the contract and allocation are not aligned properly. Due to these issues, a sole item-production is usually preferred for a DULR.

Several modifications are proposed to ensure a multiple item-production. First, it is proposed that each item is an exchangeable item, which amplifies the benefit of a distributed production. Second, the known allocation variable is defined as a continuous variable instead of a binary variable, which enables the possibility of a multiple item-production. Third, unit costs are added. Unit costs should compensate additional setup costs when the production is executed by more than one agent. Fourth, overtime costs are considered by using higher unit costs when a specific amount of units of an item is produced by an agent.

In a DULR with multiple item-production, it is suggested that the set of items $I$ is equal to the set of exchangeable items $I^C$, i.e., $I = I^C$. In conclusion, each item $i \in I$ is an exchangeable item which might be produced by more than one agent within the coalition $A$. To be precise, each item $i \in I$ can be produced by at most two agents. It means that an item can be produced by one or two agents. This results in a scenario where each agent $a \in A$ is still in charge of an individual set of items $I_a$. However, the set of items $I_a$ is not equal to the set of items $I$. As a rule, the assignment of items to agents overlaps, i.e., $\bigcup_{a\in A} I_a = I$ and $\bigcap_{a\in A} I_a \neq \emptyset$. For this reason, one agent may substitute another agent with respect to some items, i.e., the agents are rivaling.

Unit costs are introduced for compensating the setup costs. The unit costs of agent $a \in A$ for producing item $i \in I$ are defined by the variable $u_{ia}$, which are added to the objective function. Due to this procedure, production capacities are considered implicitly in the objective function of the model through overtime costs. Some approaches in the literature consider capacity constraints on the production resources explicitly (Dudek and Stadtler, 2005, 2007; Taghipour and Frayret, 2012, 2013). In the proposed DULR with multiple item-production, each agent $a \in A$ has a local objective function $f_a(y, al)$, which he or she wants to minimize, while the total costs of the supply chain are given by $f(y, al) = \sum_{a\in A} f_a(y, al)$. The modified local objective function is defined as follows (Eslikizi et al., 2015):

$$f_a(y, al) = \sum_{i \in I_a} \sum_{t \in T} (s_{ia} y_{ita} + h_{ia} l_{ita} + u_{ia}(x_{ita})) \tag{4.12}$$

$$u_{ia}(x_{ita}) = \begin{cases} u_{ia} x_{ita} & \text{if } x_{ita} \leq \overline{d}_i \\ u_{ia} \overline{d}_i + 2u_{ia}(x_{ita} - \overline{d}_i) & \text{if } x_{ita} > \overline{d}_i \end{cases} \tag{4.13}$$

By considering the local objective function (4.12), it is obvious that each agent $a \in A$ tries to minimize setup, inventory holding, and unit costs for his or her set of items $I_a$. Thereby, the setup cost $s_{ia}$ as well as the inventory holding cost $h_{ia}$ are given per agent $a \in A$ and per item $i \in I_a$. In contrast, the quantity-dependent unit costs are calculated according to the function $u_{ia}(x_{ita})$. If the production quantity of an item $i \in I$ is below a given threshold $\overline{d}_i$, unit cost $u_{ia}$ will occur. However, if the production quantity exceeds the threshold $\overline{d}_i$, the unit costs will double. By this, overtime costs are integrated into the model in a straightforward way. Any value may be used as the threshold $\overline{d}_i$. In this case, the average demand per item and per period is used as the threshold, i.e., $\overline{d}_i := \sum_{t \in T} d_{it}/|T|$. The demand $d_{it}$ per period $t$ is given for all $i \in I$ that are end products. For each item $i \in I$ that are non-end products, the demand per period is determined endogenously by the mathematical model. However, the threshold $\overline{d}_i$ is determined by the bill of materials and the external demand $d_{it}$ for the end products. Therefore, $\overline{d}_i$ can be considered as a constant for a given instance of the DULR. Further constraints of the mathematical model follow the DULR in Subsection 4.1.1. Most of the constraints do not have to be updated and can be used without any modification. Just the allocation parameter has to be defined as a continuous variable $al_{ia} \geq 0$ instead of a binary variable.

### 4.2.2. Solution approach

The proposed negotiation mechanism of Section 4.1 is used and modified to solve a DULR with multiple item-production. The extended mechanism is denoted as NBM-2. In contrast to the NBM-PWR-1, the NBM-2 uses different assignment and side payment procedures, which are also used more frequently within the negotiation mechanism instead of just using it at the beginning or at the end of the algorithm. Furthermore, side payments are used within the local cost functions of each agent. Thereby, it is identified that side payments compensate the known issue of minor agents voting against globally improving solutions. That is why the PWR procedure is omitted. In the following, the NBM-2 is described by giving an overview about the whole negotiation mechanism, updated assignment procedure, and updated side payment procedure.

**General overview:** An overview of the NBM-2, which is able to deal with a multiple item-production, is given by Alg. 7. As input parameters the known parameters of the NBM-PWR-1 are used except for the PWR parameters. Additionally, some new parameters for the allocation and side payment procedure are introduced: percentage of changeable items $si^{al}$, percentage of changeable order quantity $oq^{\text{al}}$, number of allocation rounds $r^{al}_{\max}$, percentage of Shapley rounds $pe^{\text{sh}}$, and Shapley surcharge $su^{\text{sh}}$. By initializing the algorithm, a mediator generates the initial contract $c$ as known and initializes the allocation counter $r^{\mathrm{al}}$. Based on the generated contract $c$, the best allocation $al^b$ is

**Algorithm 7:** NBM-2, cf. Eslikizi et al. (2015)

**Data**: problem data, allocation parameters ($si^{\mathrm{al}}$, $oq^{\mathrm{al}}$, $r^{\mathrm{al}}_{\mathrm{max}}$), Shapley parameters ($su^{\mathrm{sh}}$, $pe^{\mathrm{sh}}$)

1 mediator — initialize allocation counter $r^{\mathrm{al}} \leftarrow 0$
2 mediator — generate initial contract $c$ randomly
3 mediator — generate initial allocation $al^b \leftarrow \mathtt{init}(c)$
4 each $a \in A$ — evaluate contract $c$ with local objective function $f_a(c, al^b)$
5 each $a \in A$ — compute initial temperature $T_a^0$ and cooling rate $\tau_a$
6 **while** $r^{al} < r^{al}_{max}$ **do**
7     mediator — initialize negotiation counter $r^n \leftarrow 0$
8     mediator — initialize Shapley counter $r^{\mathrm{sh}} \leftarrow 1$
9     mediator — update allocation $al^b \leftarrow \mathtt{fine}(al^b)$
10     **while** $r^n < r^n_{max}$ **do**
11         mediator — generate proposal $c' \leftarrow N(c)$
12         each $a \in A$ — evaluate proposal $c'$ with local objective function $f_a(c', al^b) - sh_a$
13         each $a \in A$ — accept $c'$ with probability $P_a$
14         **if** *all agents accept proposal* $c'$ **then**
15             mediator — update contract $c \leftarrow c'$
16         **end**
17         **else**
18             **if** $r^n > (pe^{sh} \cdot r^{sh} \cdot r^n_{max})$ **then**
19                 each $a \in A$ — update side payments $sh_a$
20                 mediator — $r^{\mathrm{sh}} \leftarrow r^{\mathrm{sh}} + 1$
21                 **return** to line 14
22             **end**
23         **end**
24     **end**
25     each $a \in A$ — update temperature $T_a$
26     mediator — $r^n \leftarrow r^n + 1$
27 **end**
28 mediator — $r^{\mathrm{al}} \leftarrow r^{\mathrm{al}} + 1$
29 **return** c as mutually accepted contract and $al^b$ as best allocation

determined by the initial allocation procedure (line 2 of Alg. 7), which allocates the items to the agents depending on the contract $c$. The allocation value defines the fraction of the items that are produced by rivaling agents. Then, each agent $a \in A$ evaluates the contract $c$ with allocation $al^b$ by his or her local objective function $f_a(c, al^b)$ and determines his or her cooling schedule as known. Afterward, an allocation phase takes place. Thereby, the known negotiation phase is embedded into the allocation phase, which is responsible for updating the allocation $al^b$ based on the contract $c$. There the best allocation $al^b$ is updated by slightly increasing or decreasing the allocation parameter (line 9 of Alg. 7). Based on the updated allocation $al^b$, a negotiation phase with $r^n_{\mathrm{max}}$ rounds is performed, where the mediator presents the proposal $c'$ to all agents in each negotiation round $r^n$. Here, each agent $a \in A$ evaluates the proposal $c'$ with the allocation $al^b$ with his or her individual local objective function $f_a(c', al^b)$ and reduces the value by side payments $sh_a$.

Every time when $pe^{\text{sh}}$ percent of the negotiation rounds $r^n_{\max}$ are completed, the side payments are calculated. As long as side payments are not determined, these figures are set to zero. If the proposal $c'$ decreases the local costs of an agent, the agent will accept $c'$. If $c'$ is accepted by all agents, the proposal $c'$ will become the new jointly accepted contract and is used by the mediator to generate a new proposal. Each negotiation phase terminates after $r^n_{\max}$ rounds and is repeated until $r^{al}_{\max}$ allocation rounds are completed. Finally, the best mutually accepted contract $c$ and the best allocation $al^b$ represent the encoded solution of a DULR with multiple item-production.

**Assignment procedure:** In contrast to the previous assignment procedure of Section 4.2, it is proposed to consider two different assignment procedures: one for the generation of an initial assignment (line 3 of Alg. 7) and one for slightly updating an existing assignment during the negotiation phase (line 9 of Alg. 7). Both procedures are modified in order that a multiple item-production is possible. In the previous assignment procedure of the NBM-PWR-1, just a sole item-production was possible. First, the initial assignment procedure is addressed. Second, the finetuning assignment procedure is described.

An overview about the initial assignment procedure is given by Alg. 8. As soon as the initial contract $c$ is generated, the mediator initializes the allocation variable $al_i$ and seeks for each item $i \in I$ the best allocation $al^b_i$ (line 3 of Alg. 7). In this scenario, where at most two agents can produce an item, the allocation variables can be simplified by eliminating the agent dimension. Then, one agent just receives the current listed percentages ($al_i$) while the rivaling agent receives the remaining percentage of the production volume ($1 - al_i$). For example, agent $a_1$ gets $0\,\%$ ($al_i = 0$) of item $i$'s production volume and his or her rivaling agent $a_2$ receives $100\,\%$ ($al_i = 1$) at the beginning of this algorithm. In this procedure, allocation $al_i$ is stored as best allocation $al^b_i$ and the set of rivaling agents $A^i$ is identified. Corresponding to the current allocation $al_i$ for item $i$, the demand, the lot size, and the inventory of the rivaling agents have to be updated. Then, each rivaling agent $a \in A^i$ evaluates the updated contract with $al_i$ for $a_1$ and $1 - al_i$ for $a_2$, respectively. If the allocation $al_i$ leads to less costs than the best allocation $al^b_i$, the allocation will be updated. As long as the allocation $al_i$ is less than the entire production volume, the process of updating the allocation will be repeated by increasing the allocation $al_i$ of agent $a_1$ by $1\,\%$. If a stop criterion is met, the procedure will be repeated for the remaining items in $I$. The search terminates after 100 iterations. Finally, the best allocation $al^b_i$ for each item $i \in I$ is determined and returned.

One disadvantage of this procedure is that it is computationally challenging to determine the best allocation for each item $i \in I$ by just updating the allocation by $1\,\%$ in each round. Nevertheless, this modification is necessary for identifying a suitable initial allocation because in case that a bad initial allocation is identified it is difficult to escape from this allocation by just slightly updating it during the negotiation phase.

Beside the initial allocation, the allocation phase tries to improve the best allocation $al^b_i$ by slightly changing some of the item allocations after a certain number of negotiation rounds $r^n_{\max}$ are executed. The finetuning assignment procedure is given by Alg. 9. If the finetuning allocation procedure is activated (line 9 of Alg. 7), the mediator will choose a

**Algorithm 8:** init - Initial assignment procedure, cf. Eslikizi et al. (2015)

**Data**: problem data, contract $c$

**for** $i \in I$ **do**
  mediator  identify rivaling agents $A^i$
  mediator  compute allocation by $al_i \leftarrow 0$
  mediator  update best allocation $al_i^b \leftarrow al_i$
  **while** $al_i \leq 100$ **do**
    each $a \in A^i$  evaluate contract $c$ with $f_a(c, al)$
    **if** $f(c, al_i) < f(c, al_i^b)$ **then**
      mediator  update allocation $al_i^b \leftarrow al_i$
    **end**
    mediator  update allocation $al_i \leftarrow al_i + 1$
  **end**
  **return** $al_i^b$ best allocation of item $i$
**end**

subset $I^{\mathrm{al}} \subset I$ of items randomly. The size of $I^{\mathrm{al}}$ is defined by the given parameter $si^{\mathrm{al}}$, while the percentage of changeable order quantity is defined by $oq^{\mathrm{al}}$. Then, the mediator tries to modify the best allocation $al_i^b$ for each item $i$ in $I^{\mathrm{al}}$. Therefore, the rivaling agents $a_1$ and $a_2$ with their current best allocation $al_i^b$ and $1 - al_i^b$ have to be identified. In the next step, the best allocation can be increased or decreased randomly, which is defined by the operator $o^{\mathrm{al}}$. It can be distinguished between four scenarios depending on the operation and the production volume. An increase operation for $al_i^b$ is performed either if the operator $o^{\mathrm{al}}$ is one and the maximum production volume is not exceeded or if the operator $o^{\mathrm{al}}$ is zero and the minimum production volume is reached. A decrease operation for $al_i^b$ is performed by the remaining permutations. The finetuning allocation procedure terminates after each item $i \in I^{\mathrm{al}}$ is investigated and will be repeated as soon as the negotiation phase with $r^n_{\max}$ negotiation rounds is completed and allocation rounds are still left ($r^{al} < r^{al}_{\max}$).

**Algorithm 9:** fine - Finetuning assignment procedure, cf. Eslikizi et al. (2015)

**Data**: problem data, percentage of changeable items $si^{\mathrm{al}}$, percentage of changeable order quantity $oq^{\mathrm{al}}$, best allocation $al_i^b$

mediator  choose items and store them in $I^{\mathrm{al}}$
**for** $i \in I^{al}$ **do**
  mediator  identify rivaling agents $A^i$
  mediator  choose operation $o^{\mathrm{al}} \in \{0, 1\}$ randomly
  **if** $(o^{al} = 1 \lor al_i^b + oq^{al} \leq 100) \land (o^{al} = 0 \lor al_i^b - oq^{al} \leq 0)$ **then**
    mediator  update allocation $al_i^b \leftarrow al_i^b + oq^{\mathrm{al}}$
  **end**
  **if** $(o^{al} = 1 \lor al_i^b + oq^{al} \geq 100) \land (o^{al} = 0 \lor al_i^b - oq^{al} \geq 0)$ **then**
    mediator  update allocation $al_i^b \leftarrow al_i^b - oq^{\mathrm{al}}$
  **end**
  **return** $al_i^b$ best allocation of item $i$
**end**

The idea of modifying a subset of items is inspired by the updating approach regarding the contract, where it is proposed that slight modifications are preferable. Computational studies verified this assumption. Furthermore, it is also identified that it is preferable if the percentage of changeable order quantity is set to a low value; otherwise it is difficult to improve the solution quality. However, it is difficult to escape from local optima with just slightly updating the allocation parameter.

**Side payment procedure:** Similar to Section 4.1 side payments are used to overcome frictions and to enhance the stability within a coalition of agents by distributing the cost savings obtained by a coalition to its agents. Instead of using side payments just at the end of the algorithm like in the previous section, it is suggested to use these payments within the negotiation phase. It is even proposed to use a new evaluation criterion for accepting proposals (line 12 of Alg. 7), where the local costs are reduced by the individual side payments due to cost savings obtained by the coalition. Minor agents should be compensated for voting a globally improving solution by using side payments.

In terms of side payments, the issue remains that only the coalition $A$ is able to generate a feasible production plan, while most of the sub coalitions are not able to generate a feasible plan. It means that most of the sub coalitions cannot be used for determining side payments. Three different side payment scenarios are proposed and analyzed in order to solve this issue. Each of these scenarios uses the Shapley formula under different premises. In this section, it is assumed that it is always possible to use external manufacturers for those items that cannot be produced by any agent of a given sub coalition. The costs of outsourcing the production of an item $i \in I$ are assumed to be significantly higher compared to the costs of an agent who is able to produce $i$. Three scenarios are developed with different approaches to deal with a sub coalition that is not able to produce all items.

- Scenario 1 (Sc1) evaluates only sub coalitions that can produce all items. Sub coalitions that cannot produce all items are ignored.
- Scenario 2 (Sc2) extends Sc1 by the evaluation of sub coalitions, where the sub coalition $S \cup \{a\}$ is able to produce all items in $I$ but $S$ alone is not able to produce all items. The highest costs plus a surcharge $su^{\text{sh}}$ is assumed for those items that cannot be produced by $S$.
- Scenario 3 (Sc3) includes Sc2 and handles all remaining sub coalitions which are not able to produce all required items on their own by using the highest available costs plus a surcharge for the missing items.

It is expected that the side payments increase from Sc1 to Sc3 and therefore should have a higher impact on the acceptance decision of an agent. It has to be mentioned that an agent might have a negative marginal contribution to a sub coalition, because it might be cheaper to produce a production plan without an agent in a multiple item-production. However, this issue can be neglected because it does not appear quite often.

An example is presented in order to get a better idea regarding these scenarios. In Table 4.4, a DULR with multiple item-production is given, where four agents are able to produce six items and where each item might be produced by at most two agents.

Thereby, each agent $a \in A$ is able to produce three items. For instance, agent $a_1$ is able to produce item $i1$, $i2$, and $i6$ while agent $a_2$ is able to produce item $i2$, $i3$, and $i4$.

***Table 4.4.:*** *DULR with multiple item-production*

| agent | set of items | agent | set of items |
|---|---|---|---|
| $a_1$ | $\{i1, i2, i6\}$ | $a_3$ | $\{i1, i3, i5\}$ |
| $a_2$ | $\{i2, i3, i4\}$ | $a_4$ | $\{i4, i5, i6\}$ |

Sc1 evaluates only sub coalitions that can produce all items, which means that both sub coalitions have to be feasible. By considering Sc1 and the sub coalitions $\{a_1, a_2, a_3, a_4\}$ and $\{a_1, a_2, a_3\}$, it is obvious that the marginal contribution of $v(\{a_1, a_2, a_3, a_4\}) - v(\{a_1, a_2, a_3\})$ can be determined because both sub coalitions are feasible due to the fact that the items of agent $a_4$ can be produced by the remaining agents. Sc2 extends Sc1 by the evaluation of sub coalitions where at least the sub coalition $S \cup \{a\}$ is able to produce all items. By considering the sub coalitions $\{a_1, a_2, a_3\}$ and $\{a_1, a_2\}$, the mentioned case is visualized, where the sub coalition with agent $a_1$ and agent $a_2$ is not able to produce all items while the sub coalition with agent $a_3$ is able to do this. If this case occurs, the mentioned external price for the missing items of agent $a_3$ will be used for the sub coalition with agent $a_1$ and $a_2$. Sc3 includes Sc2 and handles all remaining sub coalitions which are not able to produce all required items.

A similar algorithm like in Subsection 4.1.2 is applied for determining side payments. The difference is that some sub coalitions are not feasible. Depending on the considered scenario (Sc1, Sc2, and Sc3), these infeasible sub coalitions are skipped or the missing items are calculated by using external prices. The side payment procedure is activated as soon as $pe^{\text{sh}} \cdot r^{\text{sh}} \cdot r^n_{\max}$ negotiation rounds are performed (line 18 of Alg. 7). The variable $r^{\text{sh}}$ counts the number of Shapley rounds. The remaining parameters are given in advance and are determined by computational experiments. Due to the additional application during the negotiation phase, the side payment procedure is used more frequently, which leads to higher computing times.

### 4.2.3. Computational studies

Two computational studies are carried out in this subsection. The studies are based on the DULR test data set from Subsection 4.1.3. First, the effect of the Shapley-based side payments with different scenarios for a DULR with multiple item-production is evaluated. Second, a study is presented, which focuses on side payments.

Obviously, the DULR instances of Subsection 4.1.3 have to be extended for a DULR with multiple item-production. Two modifications are required: (1) all items are exchangeable and (2) there are unit costs. In the known DMLULSP instances, each item $i \in I$ is assigned once to an agent $a \in A$. To solve the first issue, a procedure is used which assigns each item $i \in I$ a second time to an agent $a \in A$. The decision of which item is assigned to which agent is executed randomly. Thereby, it is also ensured that both agents $a$ and

$b$, which are able to produce item $i$, are different entities, i.e., $a \neq b$. Furthermore, unit cost $u_{ia}$ ($i \in I, a \in A$) are introduced to solve the second issue. A value for the unit costs between $1.5l_i \leq u_{ia} \leq 2.5l_i$ has been chosen randomly for the items without predecessors. The unit costs for all other items are defined as the sum of the unit costs of the immediate predecessors. By further computational studies, it is identified that the combination of high setup costs and missing production capacities is inappropriate for the problem setting at hand. Each item would still be produced by a single agent and the actual production of an exchangeable item would not be split between rivaling agents. That is why the original setup cost $s'_i$ is reduced significantly to 10 %, i.e., $s_i := 0.1s'_i$ ($i \in I$). In this study, the instance groups $s3$, $s5$, $m3$, and $m5$ are considered. Instances with one and two agents are skipped, because they are not suitable for the investigation of side payments. Therefore, it is proposed to conduct instances with three agents. To generate instances with three agents, the same procedure is used as described by Homberger (2010).

In a preliminary study, appropriate parameter values are identified for the solution approach. This study uses ten random instances from the groups $s3$, $s5$, $m3$, and $m5$. The focus is on the following parameters: percentage of changeable items $si^{\text{al}}$, percentage of changeable order quantity $oq^{\text{al}}$, number of allocation rounds $r^{al}_{\text{max}}$, number of negotiation rounds $r^{n}_{\text{max}}$, percentage of Shapley rounds $pe^{\text{sh}}$, and Shapley surcharge $su^{\text{sh}}$. Table 4.5 presents the identified values. It is worth mentioning that the total number of iterations does not change compared to the previous section.

The NBM-2 was implemented in JAVA (JDK 1.7) and the computational studies were executed on a Windows 7 personal computer with Intel Core i7-2600 processor (3.4 GHz and 16 GB of main memory). The number of best generated solutions is used as evaluation criterion.

***Table 4.5.:*** *Parameter setting of NBM-2, cf. Eslikizi et al. (2015)*

| group | instances | $si^{\text{al}}$ (%) | $oq^{\text{al}}$ (%) | $r^{n}_{\text{max}}$ | $r^{\text{al}}_{\text{max}}$ | $pe^{\text{sh}}$(%) | $su^{\text{sh}}$ (%) |
|---|---|---|---|---|---|---|---|
| $s3$ | 10 | 5 | 3 | 10,000 | 5 | 20 | 25 |
| $s5$ | 10 | 5 | 3 | 10,000 | 5 | 15 | 15 |
| $m3$ | 10 | 5 | 3 | 10,000 | 40 | 10 | 25 |
| $m5$ | 10 | 5 | 3 | 10,000 | 40 | 15 | 20 |

Since there are no reference values for this DULR, the following test setting is constructed. It is proposed that each instance of the mentioned groups is solved by four solution approaches: the NBM-2 without side payments as well as three versions of NBM-2 with integrated side payments computed according to the three scenarios of Subsection 4.2.2. The computational study solves the mentioned instance groups with 192 small and 80 medium instances. Each instance is solved three times by each approach and the best result of each approach is reported. The detailed solution values are given in Table 4.6 for the instance groups $m3$ and $m5$ and the summarized results are listed in Table 4.7.

As can be seen by both tables, the integration of side payments within the negotiation phase reduces the total costs of the coalition. In total, the NBM-2 without side payments computes the best solution in only 47 cases while the approaches with side payments

**Table 4.6.:** *Detailed solution values per instance of group m3 and m5 by NBM-2, cf. Eslikizi et al. (2015)*

| id | no side payments | side payments ($m3$) | | | no side payments | side payments ($m5$) | | |
|---|---|---|---|---|---|---|---|---|
| | | Sc 1 | Sc 2 | Sc 3 | | Sc 1 | Sc 2 | Sc 3 |
| 01 | 791,977 | 764,385 | **759,985** | 774,467 | 796,983 | 818,257 | **778,937** | 808,616 |
| 02 | **687,546** | 704,172 | 700,086 | 702,632 | 743,004 | 733,538 | **727,333** | 751,408 |
| 03 | 1,111,347 | **1,086,089** | 1,100,658 | 1,103,905 | 1,139,280 | **1,106,916** | 1,127,642 | 1,120,516 |
| 04 | 877,658 | 868,582 | **860,086** | 872,704 | 867,268 | **866,018** | 877,538 | 878,678 |
| 05 | 1,632,919 | 1,633,029 | 1,600,498 | **1,580,327** | 1,588,667 | 1,607,849 | **1,560,648** | 1,623,569 |
| 06 | 125,873 | **119,291** | 128,112 | 134,692 | 166,867 | **132,524** | 141,800 | 157,608 |
| 07 | 627,936 | 640,561 | 659,806 | **622,896** | 655,213 | **630,651** | 631,102 | 675,395 |
| 08 | 347,582 | **341,278** | 345,143 | 343,070 | 357,991 | 368,046 | 359,588 | **354,097** |
| 09 | 715,512 | 745,155 | **701,352** | 708,382 | 723,425 | **707,834** | 717,293 | 709,753 |
| 10 | **695,321** | 696,560 | 707,458 | 696,023 | 704,502 | 697,144 | 693,358 | **692,166** |
| 11 | 2,969,227 | **2,878,993** | 2,911,968 | 2,925,215 | 2,980,136 | 2,947,971 | **2,904,481** | 2,937,268 |
| 12 | 2,632,372 | 2,623,264 | 2,610,371 | **2,592,416** | **2,617,285** | 2,693,055 | 2,634,090 | 2,807,470 |
| 13 | 965,808 | 942,483 | 927,037 | **915,806** | 994,482 | 1,005,268 | 984,268 | **981,255** |
| 14 | 1,331,516 | 1,280,919 | **1,264,061** | 1,278,726 | 1,412,901 | 1,313,627 | 1,280,368 | **1,255,948** |
| 15 | 2,221,925 | 2,167,058 | **2,144,133** | 2,146,823 | 2,195,681 | 2,201,474 | **2,100,622** | 2,131,215 |
| 16 | 1,239,246 | 1,250,096 | **1,210,373** | 1,232,936 | 1,405,707 | 1,305,500 | 1,306,103 | **1,276,869** |
| 17 | 2,922,411 | 2,895,215 | 2,938,392 | **2,885,857** | 3,055,464 | 3,175,039 | **2,937,496** | 3,010,843 |
| 18 | 2,257,788 | 2,249,784 | **2,213,307** | 2,294,880 | 2,304,489 | **2,284,017** | 2,352,894 | 2,302,718 |
| 19 | 823,598 | **778,880** | 804,378 | 796,388 | 855,741 | 895,241 | 917,548 | **819,053** |
| 20 | 691,790 | **659,816** | 667,941 | 660,012 | 798,983 | 774,906 | 694,251 | **682,764** |
| 21 | **1,571,573** | 1,589,967 | 1,594,974 | 1,592,216 | 1,667,150 | 1,649,855 | 1,606,204 | **1,585,372** |
| 22 | 569,639 | 570,146 | **562,050** | 578,276 | 569,391 | 558,902 | 559,848 | **558,077** |
| 23 | 2,052,912 | **2,029,046** | 2,045,344 | 2,065,518 | 2,063,207 | 2,069,918 | **2,027,577** | 2,058,411 |
| 24 | 1,021,175 | 1,052,511 | 1,013,709 | **1,009,262** | 1,031,146 | 1,007,541 | 1,002,113 | **992,967** |
| 25 | 252,287 | 254,560 | 252,702 | **247,200** | 282,118 | **258,842** | 268,087 | 372,347 |
| 26 | 199,563 | 197,854 | **195,489** | 195,687 | 203,254 | 205,825 | 206,009 | **199,680** |
| 27 | 1,413,600 | **1,390,227** | 1,418,815 | 1,429,391 | 1,459,123 | 1,419,756 | **1,415,593** | 1,445,071 |
| 28 | 543,063 | 533,370 | **522,994** | 538,844 | 535,576 | 519,119 | **519,017** | 522,557 |
| 29 | 2,398,075 | 2,359,953 | **2,348,689** | 2,356,394 | 2,368,610 | 2,343,657 | 2,343,709 | **2,337,851** |
| 30 | 955,182 | **951,175** | 958,215 | 969,925 | 941,631 | 944,065 | 952,905 | **930,846** |
| 31 | 5,426,755 | 5,442,062 | 5,307,966 | **5,277,263** | 5,695,392 | 5,509,654 | 5,493,008 | **5,478,065** |
| 32 | 251,957 | 250,408 | **250,220** | 259,033 | 263,711 | **258,061** | 282,677 | 258,798 |
| 33 | 1,790,650 | 1,777,610 | **1,734,672** | 1,757,885 | 1,907,570 | 1,957,250 | **1,796,773** | 1,891,344 |
| 34 | 1,078,816 | 1,079,668 | **1,068,251** | 1,078,372 | 1,056,998 | 1,072,744 | **1,050,902** | 1,051,878 |
| 35 | 3,844,689 | 3,817,993 | **3,758,096** | 3,827,484 | 3,900,579 | 3,833,146 | 3,933,198 | **3,820,415** |
| 36 | 1,753,138 | 1,732,724 | **1,727,580** | 1,757,061 | 1,759,074 | 1,762,185 | **1,753,711** | 1,799,246 |
| 37 | 6,799,227 | 6,719,048 | **6,517,142** | 6,597,691 | 6,781,686 | 6,789,399 | 6,620,691 | **6,618,032** |
| 38 | 2,658,514 | 2,679,002 | 2,619,113 | **2,581,254** | 2,672,403 | **2,632,069** | 2,637,389 | 2,688,702 |
| 39 | 1,542,962 | 1,531,555 | **1,515,637** | 1,562,789 | 1,626,582 | 1,548,647 | **1,541,220** | 1,577,947 |
| 40 | 956,120 | 930,827 | **923,588** | 928,330 | 953,457 | 954,919 | **945,579** | 956,309 |

**Table 4.7.:** *Number of best solutions (including average solution time), cf. Eslikizi et al. (2015)*

| group | without side payments | with side payments | | | no. of inst. | avg. time (s) |
|---|---|---|---|---|---|---|
| | | Sc1 | Sc2 | Sc3 | | |
| s3 | 21 | 25 | 27 | 23 | 96 | 2 |
| s5 | 22 | 23 | 23 | 28 | 96 | 4 |
| m3 | 3 | 9 | 19 | 9 | 40 | 238 |
| m5 | 1 | 9 | 14 | 16 | 40 | 933 |
| total | 47 | 66 | 83 | 76 | 272 | – |

compute the best solution in 225 out of 272 cases. Especially for the medium instances ($m3$ and $m5$) side payments are promising, probably because there are agents that produce only a few lower-level items which are rather cheap. Without side payments an agent often votes against a globally improving solution, because his or her individual situation worsens.

It is also identified that there is no side payment scenario which outperforms one of the other scenarios significantly. In this case, Sc2 has the best performance, but there is still room for improvement. On average the solution values of the approaches differ about 2 %–3 % from the best identified solution value per instance. It is even suggested to improve the whole solution approach due to the fact that the fluctuation range of the solutions is significant.

In the previous study, the number of the best solutions and some detailed solution values have been presented. Here, side payments are considered in more detail for the medium instance 23 with three agents, where the side payments computed by Sc1, Sc2, and Sc3 are compared with each other. The results are presented in Table 4.8. On the left side of the Table 4.8, the local costs of the three agents $a_1$, $a_2$, and $a_3$ are presented while on the right side, the side payments of the mentioned agents are listed.

***Table 4.8.:*** *Side payments for the medium instance 23 with* $|A| = 3$, *cf.* *Eslikizi et al. (2015)*

| | local costs | | | side payments | | | time |
|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $a_3$ | $a_1$ | $a_2$ | $a_3$ | (s) |
| Sc1 | 623,975 | 718,712 | 726,309 | 58,745 | 152,517 | 64,778 | 112 |
| Sc2 | 752,436 | 567,131 | 705,031 | 126,295 | 134,470 | 308,212 | 126 |
| Sc3 | 580,089 | 836,079 | 670,084 | 298,367 | 624,255 | 643,794 | 162 |

As can be seen, the lowest side payments occur with Sc1 and the highest with Sc3. This results from the higher costs that arise for an individual agent due to outsourcing. The difference in the payments is substantial. In this case, each agent receives about 92,000 side payments with Sc1, 188,000 side payments with Sc2, and 522,000 side payments with Sc3. Finally, the required computing time in the three agent case for Sc2 is about 13 % higher than in Sc1, while for Sc3 the required time is about 29 % higher than in Sc2. For the five agent case the required time soars. Sc2 is approximately 42 % slower than Sc1 and Sc3 requires 700 % more computing time than Sc2.

In this section, a DULR with multiple item-production is solved by a set of self-interested and autonomous agents. The mathematical model of Section 4.1 has been extended to consider the rivalry among agents, i.e., an item may be produced by more than one agent. In order to identify a globally improving solution, side payments are integrated within the negotiation phase. Three different scenarios are presented for calculating the side payments. As the first computational study indicates, side payments help significantly the search process in finding contracts with lower total costs. The solution approaches with side payments found the best contract in 225 out of 272 cases (compared to the approach without side payments). The effect of side payments will become larger if the number of agents and the difficulty of solving the actual optimization problem increase. Future solution approaches for this DULR should focus on more fine-grained setup decisions, where a period wise split of production between rivaling agents is considered. Furthermore, the scalability of the approach has to be improved, in particular when the number of agents increases. By solving these issues, it should be possible to reduce the fluctuation range of this solution approach.

## 4.3. Lot size planning with production limitations

In this section, a lot size planning problem with production limitations is introduced, which extends the DULR of Section 4.2 by introducing mandatory items. Mandatory items have to be produced by an appointed agent. It means that one agent has to produce the whole production volume of certain items. In lot size planning, different types of items are hardly discussed. That is why this section is presented. To consider mandatory items, it has been necessary to extend the existing DMLULSP to the DULR with multiple item-production otherwise a reference scenario would be missing for analyzing the impact of mandatory items. The DULR is extended by considering exchangeable and mandatory items at the same time. The mathematical model is denoted as DULR with production limitations (DULR-PL). The NBM-2 is used to solve a DULR-PL, which is modified by deleting the side payment procedure and introducing modified assignment procedures. The goal of this section is to analyze the effects of mandatory items on the total costs of a DULR.

The product structure of a DULR with exchangeable and mandatory items is presented in Figure 4.3. In contrast to a DULR with multiple item-production, where all items are exchangeable items, the DULR-PL assumes that some items are mandatory items. In Figure 4.3, agent $a_1$ and $a_2$ are in charge of all items within the product structure except for six items. These six items are mandatory items (item 6, 10, 21, 25, 31, and 35) and have to be produced either by agent $a_1$ or $a_2$. The decision which agent produces which mandatory item is defined by contractual obligations and not determined by the solution approach. As in the previous sections, each agent within the coalition tries to reduce his or her local costs by also observing that the total costs of the supply chain are reduced and that mandatory items are produced by the corresponding agent.

As mentioned, a DULR-PL considers not only exchangeable items but also mandatory items. The reasons for having mandatory items are for example safety concerns or premium goods for which a manufacturer has to be able to distinguish between the type of service performed on the item. In this section, two service types are considered: $E1$ service and $P1$ service. In case, the $E1$ service is requested for an item it means that this item can be produced by any agent of the coalition while an item with the $P1$ service has to be produced by an appointed manufacturer. Depending on the service, items differ with respect to their impact on costs and production plans. The corresponding items of these services are denoted as exchangeable and mandatory items.

The remaining section is structured as follows. Subsection 4.3.1 describes a DULR-PL. Subsection 4.3.2 extends the NBM-2 to solve a DULR-PL. Subsection 4.3.3 presents the results of computational studies.

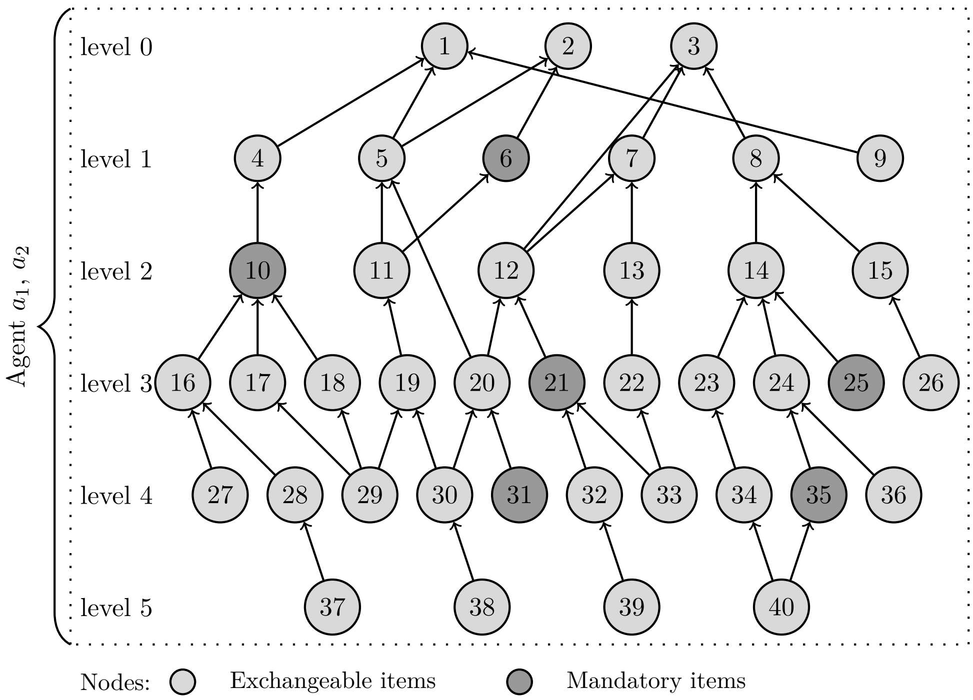


***Figure 4.3.:*** *Product structure of a DULR-PL, cf. Buer et al. (2013)*

### 4.3.1. Mathematical formulation

The mathematical model of a DULR-PL is presented in this subsection, which is based on a DULR with multiple item-production from Subsection 4.2.1. The formulation is extended by the definition of mandatory items.

A DULR-PL is jointly solved by multiple independent decision makers who have to coordinate their lot-sizing decisions over multiple planning periods in order to meet the given customer demand for each product in each period. In a DULR-PL, a set $A$ of agents is given who jointly produce a set $I$ of $m$ items. Each agent $a \in A$ produces the set of items $I_a$ with $I = \bigcup_{a \in A} I_a$. In a DULR-PL and in contrast to earlier approaches like Homberger (2010) or Buer et al. (2015), the allocation of items to agents is usually non-disjoint, i.e., $\bigcap_{a \in A} I_a \neq \emptyset$. An item is denoted as exchangeable item in case that more than one agent is able to produce it. Let $I^C$ denote the set of exchangeable items and let $I_a^C \subset I^C$ denote the set of exchangeable items of agent $a$. The existence of exchangeable items complicates the coordination problem significantly. Furthermore, there are items denoted as mandatory items. Let $I^D$ denote the set of mandatory items and let $I_a^D \subset I^D$ denote the set of mandatory items of agent $a$. The following relationship between exchangeable and mandatory items holds for each instance of a DULR-PL: $I = I^C \cup I^D$ and $I^C \cap I^D = \emptyset$.

As in the previous section, there are five types of decision variables: setup decision $y_{ita}$, lot size $x_{ita}$, internal demand $d_{ita}$, inventory $l_{ita}$, and allocation $al_{ia}$. The relative production quantity of an item, i.e., the production quota, is effective during the entire planning horizon. A production quota of, e.g., $al_{ia} = 0.35$ means that agent $a$ is awarded 35 % of the production quantity of item $i$ over all periods.

In Section 4.1, the whole production quantity of an item was always assigned to only one agent, splitting the production quantity to several agents has not been supported. The reason lies in the applied objective function for the multi-agent case, which is closely adapted from the single-agent case of the MLULSP. For such an objective function, it is not reasonable to allocate the production quantity of an item to more than one agent. To solve this issue, it was proposed to extend the objective function by integrating unit costs, overtime costs, and reducing existing setup costs in Section 4.2. In a DULR-PL $f_a(c, al)$ as the local cost function of agent $a \in A$ is used, which is the same as presented in Section 4.2. The local objective function $f_a(c, al)$ of agent $a$ consists of all individual setup costs, inventory holding costs, and unit costs. The unit costs are determined by function $u_{ia}(x_{ita})$, which assumes that the unit cost $u_{ia}$ increases when the production quantity $x_{it}$ exceeds a given threshold $\overline{d}_i$ $(i \in I)$. Here, each agent $a \in A$ wants to minimize his or her local costs, which is defined by objective function (4.1), while constraints (4.2)–(4.9) have to be satisfied. Corresponding to Ziebuhr et al. (2015), constraints (4.9) have to be replaced by the following ones:

$$al_{ia} = 1\,, \quad \forall i \in I_a^D\,, \tag{4.14}$$

$$al_{ia} \geq 0\,, \quad \forall i \in I_a^C\,. \tag{4.15}$$

In terms of the multiple-item production, constraints (4.14) ensure that all units of a mandatory item $i \in I_a^D$ have to be produced by the appointed agent $a$ while constraints (4.15) guarantee that the allocation parameter $al_{ia}$ represents a continuous variable for exchangeable items. The total costs of the supply chain are given by $f(y, al) = \sum_{a \in A} f_a(y, al)$.

### 4.3.2. Solution approach

In general, minimizing the local costs of each agent and minimizing the total costs of the supply chain are conflicting goals. Therefore, a collaborative planning approach based on negotiations is used as in the previous sections. The NBM-2 is used to solve the described DULR-PL and modified by a new assignment procedure, eliminating the side payment procedure, and integrating mandatory items. The extended NBM-2 is denoted as NBM-3. By applying the NBM-2 for the DULR, it is identified that the quality of the solutions as well as the fluctuation range of the solution approach can be improved by modifying the assignement procedure of the NBM-2. The following subsection is organized as follows. First, a general overview of the NBM-3 is given. Second, the modified assignment procedure is discussed and third, the integration of mandatory items is presented.

**General overview:** The NBM-3 is still controlled by a mediator and is outlined in Alg. 10. By initializing the algorithm, the contract $c$ with allocation $al$ has to be determined by the mediator. The first contract is generated randomly while the initial allocation is generated by splitting the fraction of each item equally between the rivaling agents. Based on these data, each agent $a \in A$ evaluates $c$ and $al$ by his or her local objective function $f_a(c, al)$ and determines his or her cooling rate $\tau_a$ and temperature $T_a^0$.

Then, the negotiation phase takes place, where a new proposal with allocation is generated by the mediator in each round $r^n$ out of $r^n_{max}$ negotiation rounds. A new allocation parameter $al'$ is generated by slightly updating the allocation parameter $al$ (see Alg. 9) while a new proposal $c'$ is generated as known. Based on the updated allocation parameter $al'$ and proposal $c'$, each agent $a \in A$ evaluates these data by his or her individual local objective function $f_a(c', al')$. An agent accepts the proposal $c'$ with allocation $al'$ if it reduces his or her local costs or by a specific probability $P_a$ if it increases his or her local cost. In Section 4.2, it is proposed to consider side payments for the evaluation of a production plan, however, they are difficult to compute under asymmetric information and not directly related to mandatory items. That is why side payments are not used in this solution approach. If a proposal with an allocation is accepted by all agents, the proposal $c'$ and the allocation parameter $al'$ will be accepted as $c$ and $al$, respectively. Furthermore, they will be used to generate a new proposal and allocation parameter. After a specific number of negotiation rounds, a contract is used for an allocation improvement procedure, where the best allocation for each item $i \in I$ is determined (see Alg. 8). In each negotiation round, the individual temperatures $T_a$ are updated. The negotiation phase is terminated as soon as each negotiation round is investigated. Finally, the best mutually accepted contract and allocation parameter are returned. These parameters represent the production plan of the agents.

**Algorithm 10:** NBM-3, cf. Ziebuhr et al. (2015)

**Data**: problem data, allocation parameter ($si^{\text{al}}$, $oq^{\text{al}}$)
mediator — generate initial contract $c$
mediator — generate initial allocation $al$
each $a \in A$ — evaluate contract $c$ with allocation $al$ by function $f_a(c, al)$
each $a \in A$ — compute initial temperature $T^0_a$ and cooling rate $\tau_a$
**while** $r^n < r^n_{max}$ **do**
  mediator — generate allocation $al' \leftarrow \texttt{fine}(al)$
  mediator — generate proposal $c' \leftarrow N(c)$
  each $a \in A$ — evaluate proposal $c'$ with allocation $al'$ by function $f_a(c', al')$
  each $a \in A$ — accept proposal $c'$ with allocation $al'$ with probability $P_a$
  **if** *all agents accept proposal* $c'$ *with allocation* $al'$ **then**
    mediator — update contract $c \leftarrow c'$
    mediator — update allocation $al \leftarrow al'$
    **if** *allocation improvement is activated* **then**
      mediator — identify best allocation $al \leftarrow \texttt{init}(c)$
    **end**
  **end**
  each $a \in A$ — update temperature $T_a$
  mediator — $r^n \leftarrow r^n + 1$
**end**
**return** $c$ mutually accepted contract and $al$ allocation

**Assignment procedure:** As mentioned, the assignment procedure of Section 4.2 is modified with the goal to improve the quality of the solutions. In Section 4.2, it is proposed that the allocation parameter is updated after a specific number of negotiation rounds. It is identified that it is preferable when the allocation parameter is updated slightly like a contract in each negotiation round. Furthermore, an allocation improvement procedure is used during the negotiation phase instead of using it for identifying a promising initial allocation parameter like in NBM-2.

For generating an initial allocation, it is proposed that the fraction of each item is split equally between the rivaling agents. It means in a scenario with two rivaling agents that both agents produce 50 % of the demanded units of an item. In Section 4.2, the initial allocation parameter was generated by identifying the best allocation for each item $i \in I$ based on the initial contract $c$. This procedure cannot be recommended, because the initial contract is generated randomly and therefore might lead into a local optimum.

In each negotiation round of NBM-3, the allocation parameter $al$ is updated by slightly changing some of the item allocations. The procedure is represented by Alg. 9. There the mediator chooses a subset $I^{\mathrm{al}} \subset$ I of items randomly. The size of $I^{\mathrm{al}}$ is defined by the given parameter $si^{\mathrm{al}}$ and the change of order quantity $oq^{\mathrm{al}}$. Each item $i$ in $I^{\mathrm{al}}$ is investigated for the modification of the allocation. Corresponding to the selected item, the rivaling agents $a_1$ and $a_2$ are identified with their current best allocation $al_i$ and $1-al_i$. Then, the allocation can be increased or decreased randomly which is defined by the operator $o^{\mathrm{al}}$. The allocation updating procedure terminates when each item has been investigated. The procedure is repeated in each negotiation round.

Instead of identifying a promising allocation parameter for the initial contract like in the NBM-2, it is proposed to use this procedure during the negotiation when the solution approach is close to be trapped in a local optimum. The allocation improvement procedure is executed several times. The first time the procedure is executed as soon as a certain number of negotiation rounds ($fi^{\mathrm{al}}$) are carried out. Then, it is repeated every time when a new mutually accepted proposal and allocation are identified and when at least 1,000 negotiation rounds are performed. As soon as the procedure is activated, the allocation parameter is rebuilt from scratch by seeking the best allocation parameter $al_i$ for each item $i \in I$, which is defined by Alg. 8. For the initial allocation one agent $a_1$ gets 0 % ($al'_i = 0$) of the production volume of item $i$ and his or her rivaling agent $a_2$ gets 100 %. In the first round, $al'_i$ is stored as the best allocation parameter $al_i$ and the set of rivaling agents $A^i$ is identified. Corresponding to the current allocation parameter $al'_i$ for that item $i$, the demand, the lot size, and the inventory of the rivaling agents have to be updated. Each agent $a \in A^i$ evaluates the updated contract with $al'_i$ for $a_1$ and $1 - al'_i$ for $a_2$, respectively. If the allocation parameter $al'_i$ leads to less costs than the best allocation parameter $al_i$, the parameter will be updated. As long as the allocation parameter $al'_i$ does not include the whole production volume, the process will be repeated by increasing the allocation parameter $al'_i$ of agent $a_1$ by 0.5 %. If the stop criterion is reached, the procedure will be repeated for the remaining items in $I$.

**Mandatory items:** To be suitable for mandatory items, it is proposed to modify the assignment procedure defined by Alg. 8 and Alg. 9. Therefore, it is necessary to introduce new subsets of items in the whole negotiation mechanism. Then, the initial allocation procedure in line 2 of Alg. 10 is updated. If a mandatory item is selected by the procedure, the appointed agent will receive the whole production share of the item. Furthermore, the

Alg. 9 has to be modified in order that mandatory items are skipped for the updating phase. At last, the initial allocation procedure of line 14 of Alg. 10 is modified in order that the appointed agent of a mandatory item receives the whole production share of this item. Based on these extensions, the solution approach can be used for solving a DULR-PL.

### 4.3.3. Computational studies

The performance of the NBM-3 for a DULR with multiple item-production and the impact of mandatory items in DULR-PL are evaluated in this subsection. First, the setup of the computational study and the generated test instances are described. Second, NBM-3 is compared with NBM-2 within a benchmark study. Third, the impact of mandatory items is analyzed by two additional studies.

Obviously, there are no test instances for a DULR-PL in the literature. That is why it is proposed to modify the DULR instances of Section 4.2 in order that some items can only be produced by one agent. These DULR instances include four instance groups denoted as $s3$, $s5$, $m3$, and $m5$ with a total of 178 instances. In this subsection, the instance groups $m3$ and $m5$ are used. The remaining instance groups are excluded from the investigation, because their number of items is limited to five items. Instances with five items are not suitable for determining the impact of mandatory items. In the following computational studies, instances with different ratios of mandatory items are generated. Ratios of 10 %, 20 %, 30 %, and 40 % are considered. For example, 10 % means that 10 % of all items are mandatory items and 90 % are exchangeable items. Obviously, it is necessary to determine which item is selected as a mandatory item and which rivaling agent produces all units of a mandatory item. Both selections are executed randomly. 15 samples are generated for each ratio and instance. In total, 60 samples are generated per instance.

In a preliminary study, appropriate parameter values for the NBM-3 are identified. This study uses ten random instances from the instance groups $m3$ and $m5$. The focus of the study is on the percentage of changeable items $si^{al}$, the percentage of changeable order quantity $oq^{al}$, the number of negotiation rounds $r^n_{max}$, the end temperature $T^{end}$, and the first activation of the allocation improvement phase $fi^{al}$. Table 4.9 shows the identified values. The NBM-3 was implemented in JAVA (JDK 1.7) and the computational studies were executed on a Windows 7 personal computer with Intel Core i7-2600 processor (3.4 GHz and 16 GB of main memory).

***Table 4.9.:*** *Parameter setting of NBM-3*

| group | $si^{\mathrm{al}}$ (%) | $oq^{\mathrm{al}}$ (%) | $r^n_{\mathrm{max}}$ | $T^{end}$ (Â°$C$) | $fi^{\mathrm{al}}$ |
|---|---|---|---|---|---|
| m3 | 2.5 | 0.1 | 400,000 | 0.01 | 160,000 |
| m5 | 2.5 | 0.1 | 400,000 | 10 | 120,000 |

NBM-3 and NBM-2 of Section 4.2 are compared regarding the quality of the solutions and the fluctuation range of the solutions. The benchmark study is based on the instance groups $m3$ and $m5$ for the DULR. Here, each instance is solved three times per solution approach. The best solutions from both solution approaches are presented in Table 4.10. Thereby, the best solution of the NBM-2 is determined out of twelve solutions because the NBM-2 uses four different solution strategies and each solution strategy is applied three times per instance.

***Table 4.10.:*** *Detailed solution values per instance of group $m3$ and $m5$ by NBM-2 and NBM-3, cf. Ziebuhr et al. (2015)*

| id | $\lvert A\rvert = 3$ agents | | $\lvert A\rvert = 5$ agents | |
|---|---|---|---|---|
| | NBM-2 | NBM-3 | NBM-2 | NBM-3 |
| 01 | 759,985.02 | 685,141.81 | 778,936.79 | 712,127.65 |
| 02 | 687,545.60 | 614,205.20 | 727,333.09 | 641,543.70 |
| 03 | 1,086,089.11 | 1,010,118.84 | 1,106,915.64 | 1,007,190.56 |
| 04 | 860,086.20 | 805,668.98 | 866,018.42 | 800,859.28 |
| 05 | 1,580,327.01 | 1,445,365.10 | 1,560,647.75 | 1,415,650.00 |
| 06 | 119,290.82 | 105,682.38 | 132,523.86 | 111,466.31 |
| 07 | 622,895.81 | 573,645.31 | 630,651.38 | 592,323.62 |
| 08 | 341,277.72 | 322,437.10 | 354,097.10 | 328,454.34 |
| 09 | 701,352.42 | 669,115.25 | 707,833.82 | 668,569.27 |
| 10 | 695,321.03 | 620,964.45 | 692,166.15 | 624,480.07 |
| 11 | 2,878,993.28 | 2,751,008.15 | 2,904,481.25 | 2,714,465.48 |
| 12 | 2,592,415.63 | 2,314,365.52 | 2,617,284.94 | 2,349,151.61 |
| 13 | 915,806.01 | 860,621.23 | 981,255.49 | 878,482.00 |
| 14 | 1,264,060.93 | 1,077,118.95 | 1,255,948.30 | 1,108,384.33 |
| 15 | 2,144,132.99 | 1,971,335.68 | 2,100,621.57 | 1,952,109.18 |
| 16 | 1,210,373.02 | 1,075,056.03 | 1,276,869.30 | 1,107,464.10 |
| 17 | 2,885,857.03 | 2,709,147.90 | 2,937,496.08 | 2,737,008.52 |
| 18 | 2,213,306.91 | 1,946,159.95 | 2,284,017.16 | 1,953,077.38 |
| 19 | 778,879.63 | 697,896.15 | 819,053.25 | 736,060.31 |
| 20 | 659,816.25 | 571,054.81 | 682,764.42 | 595,560.01 |
| 21 | 1,571,572.91 | 1,445,561.03 | 1,585,372.43 | 1,457,849.85 |
| 22 | 733,602.68 | 675,464.79 | 721,462.56 | 659,424.94 |
| 23 | 2,029,045.86 | 1,878,867.59 | 2,027,576.94 | 1,881,469.41 |
| 24 | 1,342,887.04 | 1,245,906.77 | 1,335,475.59 | 1,234,960.46 |
| 25 | 247,199.80 | 233,676.94 | 258,842.50 | 234,070.02 |
| 26 | 246,334.21 | 227,652.17 | 253,456.11 | 226,948.91 |
| 27 | 1,390,227.37 | 1,291,415.78 | 1,415,592.65 | 1,302,254.72 |
| 28 | 683,908.57 | 641,018.66 | 687,528.08 | 642,794.74 |
| 29 | 2,348,688.68 | 2,221,812.47 | 2,337,850.64 | 2,216,032.48 |
| 30 | 1,265,675.40 | 1,200,274.46 | 1,223,338.26 | 1,174,446.83 |
| 31 | 5,277,262.81 | 5,040,619.25 | 5,478,064.86 | 5,121,352.26 |
| 32 | 314,004.11 | 289,300.36 | 322,084.86 | 298,130.71 |
| 33 | 1,734,672.10 | 1,611,625.96 | 1,796,772.88 | 1,620,648.81 |
| 34 | 1,368,306.39 | 1,228,153.37 | 1,319,663.31 | 1,208,041.70 |
| 35 | 3,758,096.16 | 3,460,015.42 | 3,820,414.65 | 3,433,879.13 |
| 36 | 2,181,687.99 | 2,042,050.46 | 2,223,391.42 | 2,056,270.27 |
| 37 | 6,517,142.22 | 6,234,355.45 | 6,618,031.87 | 6,211,692.30 |
| 38 | 3,353,490.36 | 3,176,007.16 | 3,408,114.91 | 3,182,427.85 |
| 39 | 1,515,636.55 | 1,380,737.27 | 1,541,220.44 | 1,392,516.36 |
| 40 | 1,166,154.56 | 1,077,637.27 | 1,202,244.64 | 1,093,404.97 |
| mean | 1,601,085.20 | 1,485,706.54 | 1,624,835.38 | 1,492,076.11 |

Table 4.10 indicates that NBM-3 outperforms NBM-2 on 80 out of 80 instances. The achieved total cost reduction is about 7.94 % per instance of the instance group $m3$ and 8.77 % per instance of the instance group $m5$. Furthermore, the fluctuation rate of the solutions can be reduced from 5.6 % to 1.2 % per instance for instance group $m5$ and from 3.2 % to 0.9 % per instance for instance group $m3$. A fluctuation range of 1.2 % means that the worst solution has 1.2 % higher total costs than the best solution of a particular instance. Obviously, NBM-3 is the favorable solution approach that even outperforms NBM-2 on all instances in case that the solution approach is only executed once. Corre-

sponding to these figures, it is ensured that the modified assignment procedure represents a valid extension for solving the DULR. Especially the reduced fluctuation of the solution quality is important for the investigation of the impact of mandatory items because a high fluctuation rate might falsify the result of the computational studies. A disadvantage of the NBM-3 is the forced acceptance of the allocation parameter after the assignment improvement procedure, where a globally improving solution is preferred. However, the assignment improvement procedure is activated in less than 0.01 % of all negotiation rounds and the procedure generates a total costs reduction of about 4.5 % per medium sized instance (id 20–29). Furthermore, it is worth mentioning that the computational effort doubles compared to the NMB-PWR.

In the following computational studies, the impact of mandatory items is examined by considering different ratios of mandatory items. As in the previous section, each instance of the instance groups $m3$ and $m5$ for the DULR-PL is solved. Thereby, 15 samples per ratio are considered. Each instance is solved once due to the computational effort. As mentioned, ratios between 10 %–40 % are investigated. In Table 4.11, the increasing costs per mandatory item are listed.

The figures of the medium instance 01 can be interpreted as follows: a coalition with three agents has to compensate a mean total costs increase of about 0.57 % of the original production costs per mandatory item. Table 4.11 indicates that the consideration of mandatory items always leads to higher production costs than without them. Furthermore, it can be derived that the increase of costs is almost independent from the size of the coalition. On average, the coalition has additional costs of 0.46 % ($m3$) or 0.47 % ($m5$) per mandatory item. By considering the results for the different ratios, it is also observed that higher ratios of mandatory items tend to decrease the increase in costs. It is usually expected that higher ratios lead to higher additional costs. However, the DULR-PL has the characteristics of an uncapacitated production volume with different cost levels corresponding to the product structure. It is assumed that if an item of a higher level of the production structure is selected as a mandatory item, it will lead to higher additional costs compared to an item on a lower level. Thereby, it is obvious that there are higher costs for low ratios because the instances are more sensitive in these scenarios.

A second computational study is applied with the goal to confirm the mentioned assumption concerning the dependence of the additional costs and the product structure. In contrast to the former study, a random selection of mandatory items is not used. It is proposed that each item of the same level of the product structure is selected as a mandatory item. Furthermore, the medium instances 21–40 are not considered in this investigation, because their product structures have several predecessors, which belong to different agents on a different production level. This study is applied on the remaining instances with two different production structures. One product structure is denoted as $t1$ (id 01, 03, 05, 07, 09, 11, 13, 15, 17, 19) and has a five level product structure. The other one is denoted as $t2$ (id 02, 04, 06, 08, 10, 12, 14, 16, 18, 20) and has an eight level product structure. In Table 4.12, the results are presented, which can be interpreted as in the previous study.

***Table 4.11.:*** *Increase of costs by considering mandatory items (in %), cf. Ziebuhr et al. (2015)*

| id | $\lvert A\rvert = 3$ agents | | | | | $\lvert A\rvert = 5$ agents | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 10 % | 20 % | 30 % | 40 % | mean | 10 % | 20 % | 30 % | 40 % | mean |
| 01 | 0.59 | 0.56 | 0.58 | 0.56 | **0.57** | 0.51 | 0.48 | 0.48 | 0.50 | **0.49** |
| 02 | 0.35 | 0.39 | 0.45 | 0.44 | **0.41** | 0.62 | 0.37 | 0.45 | 0.43 | **0.47** |
| 03 | 0.32 | 0.43 | 0.53 | 0.41 | **0.42** | 0.79 | 0.61 | 0.56 | 0.52 | **0.62** |
| 04 | 0.44 | 0.43 | 0.36 | 0.45 | **0.42** | 0.42 | 0.36 | 0.35 | 0.39 | **0.38** |
| 05 | 0.97 | 0.70 | 0.71 | 0.72 | **0.78** | 1.29 | 0.94 | 0.77 | 0.79 | **0.95** |
| 06 | 2.11 | 1.35 | 1.28 | 0.81 | **1.39** | 0.60 | 0.60 | 0.36 | 0.60 | **0.54** |
| 07 | 0.43 | 0.47 | 0.36 | 0.42 | **0.42** | 0.34 | 0.32 | 0.32 | 0.44 | **0.36** |
| 08 | 0.37 | 0.29 | 0.38 | 0.28 | **0.33** | 0.24 | 0.19 | 0.23 | 0.29 | **0.24** |
| 09 | 0.30 | 0.39 | 0.31 | 0.31 | **0.33** | 0.44 | 0.42 | 0.39 | 0.35 | **0.40** |
| 10 | 0.52 | 0.52 | 0.49 | 0.57 | **0.53** | 0.54 | 0.47 | 0.47 | 0.48 | **0.49** |
| 11 | 0.56 | 0.54 | 0.48 | 0.52 | **0.53** | 0.62 | 0.60 | 0.61 | 0.48 | **0.58** |
| 12 | 0.58 | 0.63 | 0.55 | 0.56 | **0.58** | 0.62 | 0.53 | 0.54 | 0.49 | **0.54** |
| 13 | 0.48 | 0.32 | 0.38 | 0.29 | **0.37** | 0.40 | 0.40 | 0.30 | 0.34 | **0.36** |
| 14 | 0.71 | 0.50 | 0.53 | 0.50 | **0.56** | 0.73 | 0.58 | 0.51 | 0.39 | **0.55** |
| 15 | 0.49 | 0.56 | 0.43 | 0.50 | **0.49** | 0.45 | 0.42 | 0.42 | 0.51 | **0.45** |
| 16 | 0.57 | 0.58 | 0.53 | 0.40 | **0.52** | 0.64 | 0.55 | 0.43 | 0.55 | **0.54** |
| 17 | 0.48 | 0.46 | 0.52 | 0.46 | **0.48** | 0.49 | 0.56 | 0.55 | 0.54 | **0.53** |
| 18 | 0.54 | 0.52 | 0.53 | 0.55 | **0.54** | 0.76 | 0.61 | 0.59 | 0.56 | **0.63** |
| 19 | 0.46 | 0.45 | 0.43 | 0.43 | **0.44** | 0.43 | 0.36 | 0.30 | 0.28 | **0.34** |
| 20 | 0.46 | 0.34 | 0.45 | 0.35 | **0.40** | 0.44 | 0.37 | 0.39 | 0.25 | **0.36** |
| 21 | 0.43 | 0.42 | 0.48 | 0.41 | **0.44** | 0.47 | 0.46 | 0.43 | 0.44 | **0.45** |
| 22 | 0.49 | 0.53 | 0.50 | 0.37 | **0.47** | 0.57 | 0.51 | 0.44 | 0.45 | **0.49** |
| 23 | 0.36 | 0.45 | 0.40 | 0.46 | **0.42** | 0.46 | 0.54 | 0.49 | 0.43 | **0.48** |
| 24 | 0.52 | 0.54 | 0.53 | 0.49 | **0.52** | 0.72 | 0.46 | 0.43 | 0.46 | **0.52** |
| 25 | 0.41 | 0.07 | 0.10 | 0.22 | **0.20** | 0.66 | 0.50 | 0.63 | 0.52 | **0.58** |
| 26 | 0.33 | 0.28 | 0.24 | 0.26 | **0.28** | 0.76 | 0.44 | 0.38 | 0.28 | **0.46** |
| 27 | 0.45 | 0.44 | 0.46 | 0.46 | **0.45** | 0.48 | 0.43 | 0.39 | 0.39 | **0.42** |
| 28 | 0.30 | 0.32 | 0.33 | 0.33 | **0.32** | 0.26 | 0.30 | 0.35 | 0.31 | **0.31** |
| 29 | 0.47 | 0.56 | 0.38 | 0.43 | **0.46** | 0.39 | 0.50 | 0.52 | 0.38 | **0.45** |
| 30 | 0.44 | 0.44 | 0.36 | 0.39 | **0.41** | 0.40 | 0.29 | 0.39 | 0.38 | **0.37** |
| 31 | 0.36 | 0.41 | 0.43 | 0.44 | **0.41** | 0.45 | 0.43 | 0.36 | 0.40 | **0.41** |
| 32 | 0.68 | 0.41 | 0.23 | 0.23 | **0.39** | 0.05 | 0.08 | 0.06 | 0.08 | **0.07** |
| 33 | 0.32 | 0.36 | 0.41 | 0.34 | **0.36** | 0.39 | 0.48 | 0.55 | 0.49 | **0.48** |
| 34 | 0.50 | 0.49 | 0.46 | 0.45 | **0.47** | 0.51 | 0.47 | 0.47 | 0.44 | **0.47** |
| 35 | 0.55 | 0.51 | 0.48 | 0.47 | **0.51** | 0.57 | 0.58 | 0.43 | 0.49 | **0.51** |
| 36 | 0.54 | 0.51 | 0.47 | 0.44 | **0.49** | 0.53 | 0.51 | 0.53 | 0.43 | **0.50** |
| 37 | 0.34 | 0.45 | 0.43 | 0.43 | **0.41** | 0.49 | 0.50 | 0.51 | 0.49 | **0.50** |
| 38 | 0.36 | 0.39 | 0.47 | 0.39 | **0.40** | 0.52 | 0.44 | 0.41 | 0.41 | **0.45** |
| 39 | 0.38 | 0.40 | 0.38 | 0.40 | **0.39** | 0.58 | 0.43 | 0.44 | 0.45 | **0.47** |
| 40 | 0.28 | 0.28 | 0.40 | 0.29 | **0.31** | 0.61 | 0.46 | 0.32 | 0.36 | **0.44** |
| mean | **0.51** | **0.47** | **0.45** | **0.43** | **0.46** | **0.53** | **0.46** | **0.44** | **0.43** | **0.47** |

***Table 4.12.:*** *Impact of the product structure (in %), cf. Ziebuhr et al. (2015)*

| type | $\lvert A\rvert$ | level of product structure | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| t1 | 3 | 7.86 | 1.52 | 0.49 | 0.23 | 0.16 | - | - | - | - |
| t1 | 5 | 6.04 | 1.33 | 0.57 | 0.25 | 0.21 | - | - | - | - |
| t2 | 3 | 7.50 | 1.38 | 0.83 | 0.87 | 0.50 | 0.45 | 0.39 | 0.32 | 0.31 |
| t2 | 5 | 4.97 | 1.21 | 0.61 | 0.52 | 0.43 | 0.39 | 0.32 | 0.30 | 0.18 |

It can be concluded that the increase of costs per mandatory item decreases from the first production level to the last one in both scenarios ($t1$, $t2$). Furthermore, the product level of an item has a significant impact on the additional costs. For example, when an item of the highest level concerning the product structure is selected as a mandatory item, then it is recommended to be aware of higher costs as compared to the item of the lowest level in the product structure. By considering the figures of Table 4.12, it can be observed that especially the highest level of the product structure has a significant impact on the solution.

This section analyzes a DULR-PL. A DULR-PL is an inter-organizational lot-sizing problem with rivaling agents where some items have to be produced by an appointed agent of the coalition. These items are denoted as mandatory items. A negotiation mechanism denoted as NBM-3 is applied to solve this problem. The NBM-3 extends the NBM-2 by a new procedure for identifying a suitable multiple item-production among the agents and a procedure for handling mandatory items. In a benchmark study, NBM-3 is compared with the only existing solution approach NBM-2 for DULR with multiple item-production. Thereby, it is identified that NBM-3 outperforms the other solution approach on 80 out of 80 instances where on average over all instances a total costs reduction of about 8 % per instance is achieved by simultaneously reducing the fluctuation range of the solution approach. Based on these figures, NBM-3 is used for solving a DULR-PL, where several findings could be derived. The computational studies indicate that mandatory items always lead to higher production costs and that items on a higher level of the product structure have higher impact on the total costs. In this study, it is observed that each mandatory item causes additional costs of about 0.47 %. For future research, it appears promising to apply the findings for developing an improved solution approach, where heuristics focus on the investigation of items on a higher level of the product structure because their impact on the solution quality is more significant.

# Part II.

# Transportation planning with Mandatory Tasks

# 5. Transportation Planning

In order to produce goods, it has to be ensured that the required raw materials and components are available. Therefore, transportation planning is required. Similar to lot-size planning it is not sufficient to solve the transportation planning process by manual planning. Methods of operations research are suggested. This chapter focuses on basic optimization problems of transportation companies. In this case, TPPs and vehicle routing problems are presented and analyzed, where goods have to be picked up and delivered. Internal and external fulfillment modes are applicable to transport these goods.

First, the topic of transportation planning is addressed in Section 5.1. In general, transportation planning represents a planning function within supply chain management and is responsible for determining transportation flows. Thereby, the thesis focuses on the determination of a transportation plan by solving different vehicle routing problems depending on the applicable fulfillment modes for a transportation request.

Second, TPP and the corresponding vehicle routing problems are addressed in Section 5.2. In literature, one of the main assumptions was that transportation requests in TPP are fulfilled by internal resources. However, transportation companies are forced to use different fulfillment modes due to high demand fluctuations. That is why new mathematical models are developed, where transportation requests can be fulfilled by internal and external resources, respectively. Today, a transportation company can choose among self-fulfillment, subcontracting, and collaboration for fulfilling a transportation request. Corresponding to this observation, this section is separated into two subsections: single decision making problems and group decision making problems. In single decision making problems, just TPP with internal resources are presented, where one decision maker is in charge of all relevant planning data. In contrast, group decision making problems consider internal and external fulfillment modes where several decision makers are usually in charge of relevant planning data and each decision maker is aware of its individual data. In both cases, a brief literature review is presented. However, it is mainly focused on group decision making problems which are used in the following chapters.

Third, the topic of mandatory tasks in TPP is addressed in Section 5.3. As in production planning, there are also mandatory tasks in TPP, which have to be fulfilled by certain fulfillment modes due to contractual obligations. In TPP, these mandatory tasks are denoted as mandatory requests. First, the practical relevance of mandatory requests is discussed. For example, a shipper asks a transportation company for using its internal resources because the shipper does not trust its external resources. Second, a literature overview regarding the topic of TPP with exchangeable and mandatory requests is given. Compared to production planning, some publications are identified which deal with the topic of mandatory requests in TPP. Just a few of them propose new mathematical models and solution approaches, where mandatory requests are considered. Most of the publications only mention the existence of mandatory requests.

Fourth, the basic solution approach of this part is explained in Section 5.4. The solution approach was introduced by Wang et al. (2014) and is known as CGB-heuristic (i.e., column generation based heuristic). The idea of the CGB-heuristic is to split a TPP into a master problem and subproblem. The goal of the subproblem is the identification of promising columns while the master problems has to identify a feasible transportation plan out of the submitted columns.

## 5.1. Relevance of transportation planning

Transportation planning belongs to logistics and represents an important planning task within supply chain management, where goods have to be transported between different machines or between a variety of locations (Voß and Woodruff, 2006, p. 219). Thereby, transportation planning deals with the determination of the transportation flows between factories and customers. It means that transportation planning tries to optimize the transportation sequence and corresponding transportation volume. (Ohrt, 2008, p. 8) In recent years, transportation planning is getting more important due to the globalization, where more and more goods have to be transported far greater distances. That is why a sufficient planning is crucial for many transportation companies. (Rushton et al., 2010, pp. 331-335)

Logistics can be defined as "the management of all activities which facilitate movement and the coordination of supply and demand in the creation of time and place utility." (Heskett et al., 1973) The term "logistics" is sometimes used equivalent to supply chain management. This is not appropriate, because logistics focuses on the material flow among the different companies, while supply chain management is a comprehensive approach. It means that logistics represents one part of supply chain management. Nevertheless, the significance of logistics has been rising for several years because logistics has a great potential to reduce the costs of a supply chain. (Arndt, 2008, pp. 27-33) In logistics, the objective is to improve performance (e.g., quality and time) and costs of a company. However, these objectives are conflicting (e.g., requiring a high quality can lead to long production times and high costs), which is why the goal is to identify the best trade-off between these objectives. (Arndt, 2008, pp. 124-130) The origin of logistics comes from military science. There logistics is responsible to provide, organize and transport troops, weapons, and equipment. The topic of logistics gains attention within companies since the mid 1950s. (Arndt, 2008, pp. 27-33)

The topic of transportation planning is assigned to the short-term planning of distribution within supply chains. The dimension distribution deals with the replenishment of warehouses with goods and transportation of goods from factories to customers (Wenger, 2010; Ohrt, 2008, p. 39, p. 5), while the dimension short-term planning means that activities in distribution planning on the lowest planning level with a planning horizon between few days and up to three months are considered (Stadtler et al., 2015, p. 82). This classification does not mean that transportation costs only occur in distribution, e.g., they also occur in procurement (Stadtler et al., 2015, p. 87). In recent years, it is observed that there are more and more transportation requests with small transportation volumes

(Wenger, 2010, p. 40). Sophisticated approaches (like operations research) are proposed to handle this situation in transportation planning.

In the European Union (EU-28), the freight transport is measured by Eurostat. Corresponding to Eurostat, it is observed that the total inland freight transport in the EU-28 has an estimated volume of 2200 billion ton-kilometer (tkm) in 2013. This represents an increase of about 100 billion tkm compared to 2012. Different transport modes are applicable for transporting goods. In the EU zone, inland freight transport is mainly performed by road transport, rail transport, and inland waterways. (Eurostat, 2016) Each of these transport modes has a different share within the freight transport of a country. In Figure 5.1, the share of these transport modes is presented for six countries within the EU-28 and the EU-28 itself. Thereby, it is worth mentioning that intercontinental freight transports like maritime, air, and pipeline transport are not included in this figure. Corresponding to this figure: 74.9 % of the total inland freight transport was transported

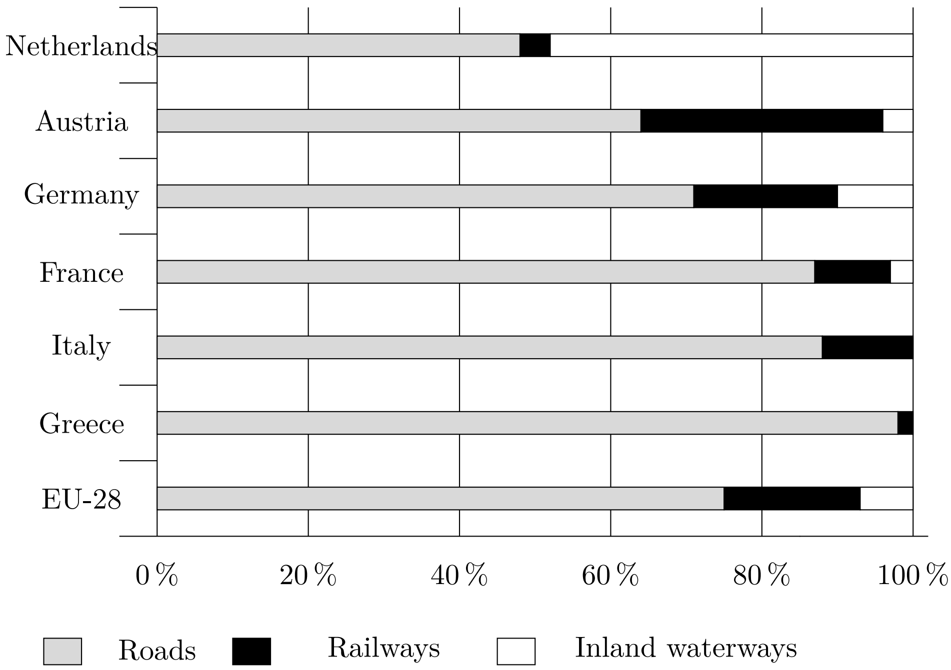


***Figure 5.1.:*** *Modal split of inland freight transport (in %) of total inland tkm, 2013, cf. Eurostat (2016)*

over roads, 18.2 % over rails, and 6.9 % over inland waterways in the EU-28 zone in 2013. It means that road transport is the dominant mode in terms of inland freight transport. Nevertheless, there are also some countries in the EU-28 where rail transport (e.g., Austria) and inland waterways (e.g., Netherlands) are also quite popular compared to road transport. The decision, which transport mode should be used for a product, depends on the trade-off between costs and service. (Rushton et al., 2010, pp. 331-335) The different advantages and disadvantages of the transport modes are listed in Rushton et al. (2010, pp. 334-340). In general, it is expected that road transport will remain the dominant transport mode in the near future. That is why this part focuses on road transport.

A recent trend in road transport is that transportation companies are confronted with high demand fluctuations and high fixed costs in competitive markets (Chu, 2005). That is why companies in transportation markets are forced to reduce their costs and to improve their flexibility by using for the fulfillment of requests both, a limited private fleet and external resources. In a poll performed in 2006, it was found that 80 % of the participants use third-party logistics and that the most common outsourced logistics function is the transportation of goods (Eyefortransport, 2006). In terms of transportation tasks, many companies use freight forwarders for fulfilling transportation requests. These freight forwarders take the responsibility for the transport. This means that they "act as principals to the transport contract, for example by providing road and container groupage services or air freight consolidation" (Rushton et al., 2010, pp. 349-350). For example, a medium sized freight forwarder is investigated in Kopfer et al. (2006) where about 30 % of the transportation requests are served by the private fleet and the remaining ones are served by external resources.

As soon as the basic requirements are determined, like which and how many vehicles are necessary, these vehicles have to be optimized regarding their daily utilization. Therefore, it is necessary to determine the "specific delivery route requirements and then calculate from these how many vehicles and drivers are required to undertake the operation" (Rushton et al., 2010, p. 440). This task is also known as vehicle routing, which deals with the determination of a transportation plan that fulfills all restrictions and tries to optimize the objectives simultaneously (Wenger, 2010, pp. 39-40). Depending on the problem size, this can be done manually or by using computer programming. The goal is usually to minimize the number of used vehicles by simultaneously serving customers as efficiently as possible. In vehicle routing problems, different restrictions like loading, time, and routing constraints as well as different objectives like minimizing transportation costs or minimizing $CO_2$ emissions are considered. (Rushton et al., 2010, p. 440)

## 5.2. Optimization problems in transportation planning

Common TPP and vehicle routing problems are presented and discussed in this section. In TPP, different vehicle routing problems have to be solved depending on the fulfillment mode. Thereby, vehicle routing deals with the determination of a transportation plan of a transportation company with one fulfillment mode. Vehicle routing problems are common and well studied optimization problems in operations research. Two decision problems have usually to be solved: a clustering and a routing problem. In a clustering problem, a set of requests has to be assigned to vehicles, while in a routing problem the sequence of requests has to be determined on each vehicle. (Wenger, 2010; Cordeau et al., 2007, p. 40, p. 377) In literature, the most common vehicle routing problems are: the traveling salesman problem (TSP), the vehicle routing problem (VRP), and the pickup and delivery problem (PDP). Each of the mentioned vehicle routing problems uses internal resources for fulfilling requests and can be categorized as a single decision making problem, where one decision maker is in charge of all planning data.

However, these mathematical models are extended by considering external resources for fulfilling requests in recent years. Common TPP with internal and external resources are: the integrated operational transportation planning (IOTP) problem, the collaborative operational planning (CTP) problem, and the collaborative integrated operational transportation planning (CIOTP) problem. In contrast to the previous vehicle routing problems, the latter ones have usually several decision makers. That is why these problems are denoted as group decision making problems.

First, single decision making problems like the TSP, VRP, and PDP are described in Subsection 5.2.1, where one decision maker tries to minimize the total costs. Second, group decision making problems like the IOTP, CTP, and CIOTP problem are discussed in Subsection 5.2.2, where several decision makers are considered by including external resources. All of the described group decision making problems are based on a PDP. It is worth mentioning that the IOTP problem can also be modeled as a single decision making problem. In this thesis, the problem is classified as a group decision making problem due to the appearance of several stakeholders.

### 5.2.1. Single decision making

In this subsection, three vehicle routing problems are described. Thereby, it is mainly focused on the PDP while the VRP and TSP are briefly described. Each of these vehicle routing problems is connected with each other in order that the PDP represents a generalization of the VRP and the VRP represents a generalization of the TSP.

**Traveling salesman problem:** The basic and simplest model of vehicle routing is the TSP, which is a well studied optimization problem and has various applications in logistics, genetics, manufacturing, and telecommunications (Applegate et al., 2007). Based on the TSP, different optimization problems can be modeled in vehicle routing. (Voß and Woodruff, 2006, pp. 220ff.) The objective of the TSP is to find the shortest tour that serves each location in a given list exactly once and returns to the starting location (i.e., depot). A tour defines the set of locations which have to be served by one vehicle. The ordering in which the locations are served is called a route. The origin of the TSP is not proved. Corresponding to Müller-Merbach (1983), it was identified that the TSP is already mentioned in a German handbook for Salesmen from 1832. In this manual, five tours are suggested through Germany and Switzerland, while one of them is a traveling salesman tour in accordance to the known definition. However, a mathematical formulation of a TSP is missing in the manual. (Schrijver, 2005) One of the first mathematical formulations of a TSP appeared in 1930s. In Figure 5.2, a feasible transportation plan for a TSP is given, which contains one vehicle, one depot, and 13 locations.

As can be seen in this figure, a list of locations is served by a salesman. Each location is served exactly once and a salesman begins and ends his or her tour at the depot $d$. Obviously, this is just one possible solution. In this case, the route with the lowest transportation costs is presented. A routing problem has to be solved for identifying a solution for a TSP. However, it is not easy to identify the optimal solution for a TSP. Karp (1972) proofs that the TSP is NP-hard. That is why heuristics are often preferred. Different exact and heuristic approaches can be found for solving the TSP. To measure the progress of the solution approaches, it is recorded which sizes of instances have been solved

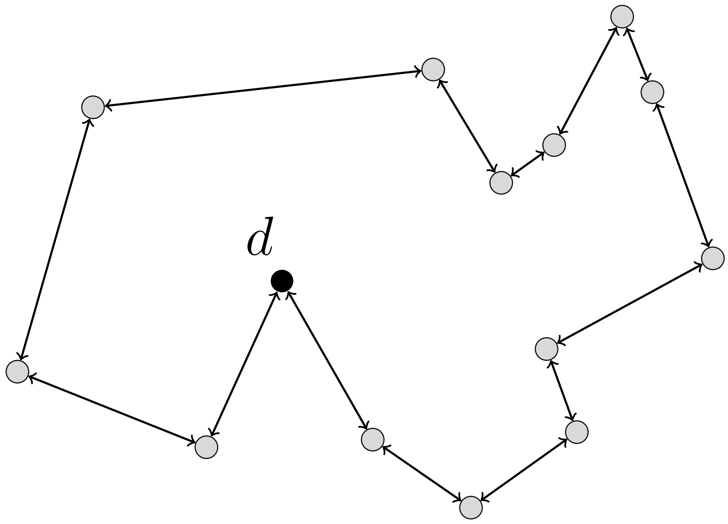


***Figure 5.2.:*** *Transportation plan of a TSP*

successfully in the past, while in 1952 TSP instances with 49 cities could be solved, recent algorithms are able to solve instances with up to 85,900 cities (Applegate et al., 2007). Further information regarding solution approaches for a TSP can be found by Schrijver (2005) and Applegate et al. (2007).

**Vehicle routing problem:** One of the most popular problems in vehicle routing, which has to be solved "each day by thousands of companies and organizations engaged in the delivery and collection of goods and people", is the VRP (Cordeau et al., 2007, p. 367). The VRP was introduced by Dantzig and Ramser (1959), where a practical problem of delivering gasoline is described and a mathematical model is presented. Corresponding to Laporte (1992), the VRP can be described as "the problem of designing optimal delivery or collection routes from one or several depots to a number of geographically scattered cities or customers, subject to side constraints". Thereby, each location is served exactly once by one vehicle. In contrast to the TSP, where it is assumed that each location can be served within one tour, the VRP assumes a set of vehicles and this set of vehicles has to serve a set of customers with a given demand. There are many applications for the VRP in practice, which differ regarding the objectives. The common objective of a VRP is to minimize the total distances of all tours. In Figure 5.3, an example for a VRP with four vehicles and 14 locations is given. The vehicles and customers have a given capacity or demand, respectively.

As can be seen in this figure, each vehicle starts and ends its tours at the depot by serving each location exactly once per vehicle. In this case, the objective is to minimize the driven distance. Besides this formulation, many variants of the VRP are studied in literature. There is also a wide variety of solution approaches. Each of them has to solve a clustering and routing problem. It means that locations have to be assigned to vehicles and to be ordered within the tours. Since the VRP represents a generalization of the TSP, it is obvious that the VRP is therefore NP-hard and that is why heuristics are preferred. (Cordeau et al., 2007, p. 368) Different solution approaches are developed for the VRP. An overview can be found by Cordeau et al. (2007, pp. 370-385).

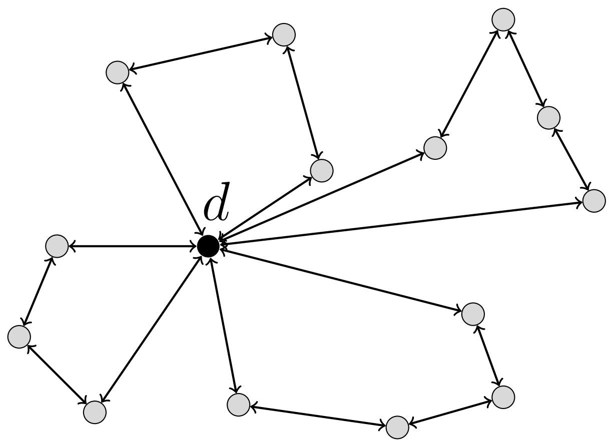


***Figure 5.3.:*** *Transportation plan of a VRP*

The mentioned vehicle routing problems (i.e., TSP and VRP) can be extended by time windows. By using time windows, it is defined when a service at a customer has to be started. This might be relevant in case that a vehicle has to deliver goods within a certain time window. Due to time windows, a scheduling problem has to be solved as well. Two type of time windows ar known: hard and soft time windows. Hard time windows cannot be violated without leading to an infeasible solution, while soft time windows can be violated and thereby leading to additional costs. (Ohrt, 2008, pp. 16-17) This thesis focuses on hard time windows.

**Pickup and delivery problem:** The PDP is a generalization of the VRP, where goods or persons have to be transported from a pickup location to a delivery location. The latter scenario, where persons are transported from home to hospital, is also known as dial-a-ride problem and is one of the most popular PDP in literature. Corresponding to Cordeau et al. (2007), the PDP "aims to design a set of least cost vehicle routes starting and ending at a common depot in order to satisfy a set of pickup and delivery requests between location pairs, subject to side constraints". In this thesis, a one-to-one PDP is considered, where each request with one pickup location is assigned to one delivery location. Different kinds of PDP formulations can be found by Cordeau et al. (2007). In practice, a PDP has many applications, including "the transport of the disabled and elderly, sealift and airlift of cargo and troops, and pickup and delivery for overnight carriers or urban services" (Desaulniers et al., 2001). Depending on the application, there is a wide variety of PDP in literature, an overview can be found by Savelsbergh and Sol (1995).

The considered pickup and delivery problem with time windows (PDPTW) can be defined on a directed graph $G = (V, A)$, where $V$ represents the set of nodes and $A$ represents the set of edges with $A = V \times V$. A PDPTW includes $n$ less than truckload requests and $m$ vehicles. The set of nodes contains the set of pickup nodes $P = \{1, ..., n\}$, set of delivery nodes $D = \{n+1, ..., 2n\}$, start depot $\{o\}$, and end depot $\{o'\}$, i.e., $V = P \cup D \cup \{o\} \cup \{o'\}$. For the fulfillment of a pickup and delivery request (referred to as request) with load $l_i \geq 0$, goods have to be transported from the pickup location $i$ to the delivery location $n+i$ with $l_{n+i} = -l_i$. A service of duration $s_i$ has to be started at node $i$ within a hard time window $[a_i,\ b_i]$. The corresponding travel time $t_{ij}$ and distance+$d_{ij}$ are given for each edge $(i,j) \in A$. The set of vehicles is denoted by $K = \{1, ..., m\}$. All vehicles have the same capacity $Q$, fixed cost rates $\alpha$, and variable cost rates $\beta$.

Three types of decision variables are used in a PDPTW. The binary variables $x_{ijk}$ are one if vehicle $k$ travels from node $i$ to node $j$ and zero otherwise. The starting time of a service at node $i$ by vehicle $k$ is represented by $w_{ik}$, while the variable $L_{ik}$ defines the load of vehicle $k$ after the service is completed at node $i$. Based on Desaulniers et al. (2001); Ropke and Pisinger (2006), a PDPTW can be modeled as follows:

$$\min IP^{PDP} = \sum_{k \varepsilon K} \sum_{(i,j) \varepsilon A} \beta d_{ij} x_{ijk} \tag{5.1}$$

$$\text{s. t.} \quad \sum_{k \varepsilon K} \sum_{j \varepsilon V \backslash \{o\}} x_{ijk} = 1 \qquad \forall i \in P, \tag{5.2}$$

$$\sum_{j \varepsilon P \cup D} x_{ijk} - \sum_{j \varepsilon P \cup D} x_{j,n+i,k} = 0 \qquad \forall k \in K, \forall i \in P, \tag{5.3}$$

$$\sum_{j \varepsilon P \cup \{o'\}} x_{ojk} = 1 \qquad \forall k \in K, \tag{5.4}$$

$$\sum_{i \varepsilon D \cup \{o\}} x_{io'k} = 1 \qquad \forall k \in K, \tag{5.5}$$

$$\sum_{i \varepsilon V \backslash \{o'\}} x_{ijk} - \sum_{j \varepsilon V \backslash \{o\}} x_{jik} = 0 \qquad \forall k \in K, \forall j \in P \cup D, \tag{5.6}$$

$$w_{ik} + s_i + d_{ij} - M(1 - x_{ijk}) \leq w_{jk} \qquad \forall k \in K, \forall (i,j) \in A, \tag{5.7}$$

$$a_i \leq w_{ik} \leq b_i \qquad \forall k \in K, \forall i \in V, \tag{5.8}$$

$$w_{ik} \leq w_{n+i,k} \qquad \forall k \in K, \forall i \in P, \tag{5.9}$$

$$L_{ik} + l_j - M(1 - x_{ijk}) \leq L_{jk} \qquad \forall k \in K, \forall (i,j) \in A, \tag{5.10}$$

$$L_{ik} \leq Q \qquad \forall k \in K, \forall i \in V, \tag{5.11}$$

$$L_{ok} = L_{o'k} = 0 \qquad \forall k \in K, \tag{5.12}$$

$$x_{ijk} \in (0,1) \qquad \forall k \in K, \forall (i,j) \in A, \tag{5.13}$$

$$w_{ik} \geq 0 \qquad \forall k \in K, \forall i \in V, \tag{5.14}$$

$$L_{ik} \geq 0 \qquad \forall k \in K, \forall i \in V. \tag{5.15}$$

The goal of a PDPTW is to minimize the total transportation costs (5.1), which correspond to the fixed and variable costs for the vehicles. In the linear objective function, the fixed cost term of the fleet $\sum_{k \in K} \alpha_k$ is omitted because it is a constant and cannot be optimized. Constraints (5.2) ensure that each request at pickup location $i \in P$ is assigned to exactly one vehicle, while the precedence constraints (5.3) define that the same vehicle $k$ is assigned to the request with the pickup location $i$ and the corresponding delivery location $n + i$. Constraints (5.4) and (5.5) ensure that each vehicle $k \in K$ starts and ends at the corresponding depot $o$ or $o'$. Constraints (5.6) are the flow balancing constraints, which ensure that if a pickup or delivery location is served by a vehicle, the same vehicle will leave this location. Constraints (5.7) define the starting time of a service at a location, which has to lie within the time window defined by constraints (5.8). These constraints ensure that subtours are excluded. Furthermore, constraints (5.9) guarantee that goods are picked up before a delivery option can be performed. The load variable is defined by constraints (5.10)–(5.12), which ensure that the vehicle capacity is not exceeded and that each vehicle is empty at the depot. At the end of the formulation, binary decision variables $x_{ijk}$ and decision variables $w_{ik}$ and $L_{ik}$ are defined by constraints (5.13)–(5.15). It is worth mentioning that constraints (5.7) and (5.10) are given in their linearized form.

Since the PDP is a special case of the TSP, it is obvious that the PDPTW is NP-hard. To solve a PDPTW, different exact and heuristic approaches are developed over the last 30 years. Thereby, it is distinguished between single-vehicle and multiple-vehicle PDPTW. In the past, solution approaches for the former scenario were developed, while the current research focuses on the multiple-vehicle scenario. This thesis focuses on the latter case with multiple vehicles because it is more common in practice.

One of the first exact approaches for the multi-vehicle PDPTW can be found by Dumas et al. (1991), where a column generation scheme is presented to solve the mentioned vehicle routing problem. Thereby, a set-partitioning formulated master problem is solved by a branch-and-bound approach while a constrained shortest path formulated subproblem is solved by a dynamic programming algorithm. A similar solution approach can be found by Savelsbergh and Sol (1998). In this case, a set-partitioning formulated master problem is solved by a branch-and-price algorithm while the subproblem is solved by a heuristic construction algorithm with the goal to reduce the computational effort compared to a dynamic programming algorithm. Another efficient solution mythology for solving the PDPTW is given by branch-and-cut algorithms. For example, Lu and Dessouky (2004) propose a branch-and-cut algorithm for a PDP and PDPTW by using two-index flow variables, while Ropke et al. (2007) propose a branch-and-cut-algorithm for two new formulations of the PDPTW, which contain an exponential number of constraints but fewer variables and tighter bounds. An overview about exact approaches can be found by Cordeau et al. (2007).

Heuristics are used to solve larger instances. In the past, simple heuristics like cluster-first, route-second methods, and insertion procedures are developed. However, these solution approaches often get stuck in local optima that is why metaheuristics are preferred. A common solution methodology for solving a PDP is the tabu search algorithm. In Nanry and Barnes (2000), a reactive tabu search algorithm is used, which applies different neighborhoods during the local search. Li and Lim (2001) propose a tabu-embedded SA. A tabu search is used in Lau and Liang (2001), where bundles of requests are exchanged instead of single ones. A second solution methodology is given by a large neighborhood search (LNS). In Bent and Hentenryck (2006), a two-stage heuristic for the PDPTW is proposed, where an SA is applied to minimize the number of routes and a LNS is applied to reduce the total route length. Ropke and Pisinger (2006) extend the LNS by an adaptive mechanism, where several removal and insertion heuristics are used during the local search. The solution approaches based on LNS are the most promising solution approaches for a PDP. Especially the LNS of Ropke and Pisinger (2006) is one of the best heuristics based on the reported results for instances with up to 1,000 locations.

### 5.2.2. Group decision making

In the previous subsection, TPPs with one fulfillment mode have been presented, where one decision maker is in charge of all planning data. However, freight forwarders use different fulfillment modes for fulfilling transportation requests due to the globalization. In this case, it is not sufficient that just the private fleet is optimized, external capacities have to be optimized as well. Thereby, it occurs that additional companies are involved in the transportation planning, which support the transportation process. That is why existing mathematical models and solution approaches have to be extended. In this subsection, the combinations of self-fulfillment and subcontracting, self-fulfillment and collaboration and self-fulfillment, subcontracting, and collaboration are discussed.

In Figure 5.4, a TPP of a freight forwarder is presented. This freight forwarder has to identify the best fulfillment mode for each of his or her transportation requests. The freight forwarder can choose among the fulfillment modes self-fulfillment, subcontracting, and collaboration. In total, the freight forwarder has to solve three subproblems in terms of this TPP: mode selection, vehicle routing and scheduling as well as freight consolidation.

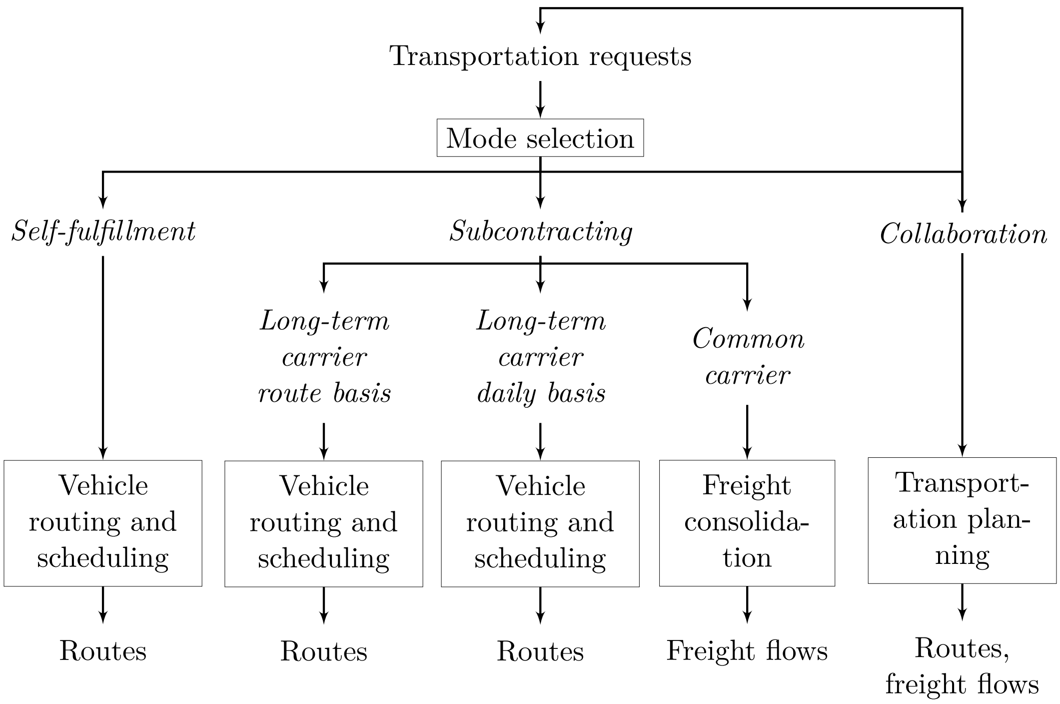


***Figure 5.4.:*** *Transportation planning with different fulfillment modes, c.f. Wang (2014)*

First, the freight forwarder has to decide which fulfillment mode he or she should select for the fulfillment of a transportation request. On the left side of this figure, the self-fulfillment mode is listed, which represents the traditional mode for fulfilling a request. In case that self-fulfillment is selected as fulfillment mode, it means that a vehicle of the private fleet is used, where the routing is done by the freight forwarder. The size of the private fleet is limited due to high fixed costs (e.g., buying vehicles and paying drivers). The most attractive requests are usually assigned to the private fleet and the remaining requests are outsourced. This procedure is also known as "cherry-picking". One type of external resources is given by subcontracting, where there is a hierarchical related partnership between the freight forwarder and carriers. It means that the freight forwarder is in power. In terms of subcontracting, it is distinguished between long-term carriers and common carriers. Common carriers are employed from case to case for transportation requests in exchange of a freight charge on the spot market. An example of a virtual spot market is Teleroute, where vehicle capacities are exchanged with the goal to achieve a better utilization of the capacities (Werner, 2013, p. 197). Freight forwarders also hire transportation capacities of carriers to an agreed limit (e.g., maximal time period or route length) based on long-term contractual agreements and take over the planning for the hired capacities (Song and Regan, 2003). These long-term carriers can be paid on a route basis (RB) or on a daily basis (DB). A second option of external resources is represented by collaborative planning. In CTP, independent freight forwarders try to improve their planning situation by reallocating their transportation requests or capacities within a horizontal coalition (Wang and Kopfer, 2014). Each of the freight forwarders faces an individual TPP. There the goal is the identification of a transportation plan where each freight forwarder reduces his or her operational costs by exchanging transportation requests with each other. Based on the decision which fulfillment mode should be selected (i.e., first subproblem), the second subproblem is the vehicle routing and scheduling problem, which has to be solved for each of the mentioned fulfillment modes except for the common carrier option. In case a common carrier is employed, a freight consolidation problem has to be solved as third subproblem.

To get a better idea regarding the financial impact of different fulfillment modes, an overview is given by Figure 5.5. In this figure, the transportation costs regarding the degree of activity for private vehicles (graph a), rented vehicles on mode RB (graph b), rented vehicles on mode DB (graph c), and common carriers (graph d) are presented.

In terms of private vehicles, a freight forwarder has variable and fixed costs. As can be seen, variable costs depend on the driven distance by multiplying it with a constant cost rate, while fixed costs are independent from the driven distance and represent a predefined costs block. A high utilization rate of the private fleet is aimed due to these fixed costs. Furthermore, a request can be subcontracted either by hiring transportation capacities or by using a common carrier on a spot market. For rented vehicles on mode RB variable costs are charged. These costs depend on the driven distance and are usually much higher than the variable costs of a private vehicle. In contrast, there are only fixed costs as a kind of a daily flat rate for a rented vehicle on mode DB, where a vehicle can be used up to an agreed limit (e.g., travel distance or time). It is obvious that this flat rate has higher fixed costs than by using private vehicles. Another type of subcontracting is given by using common carriers, where a service is requested. This type is also known as freight consolidation, where the fee for a request depends on the length and the weight of the flow of cargo. (Krajewska and Kopfer, 2009)

**Integrated operational transportation planning:** Simultaneously solving the combined problem of vehicle routing for the private fleet and the optimal employment of common and long-term carriers is known as IOTP (Krajewska and Kopfer, 2009). In contrast to other publications, which investigate TPP with subcontracting, several types of subcon-

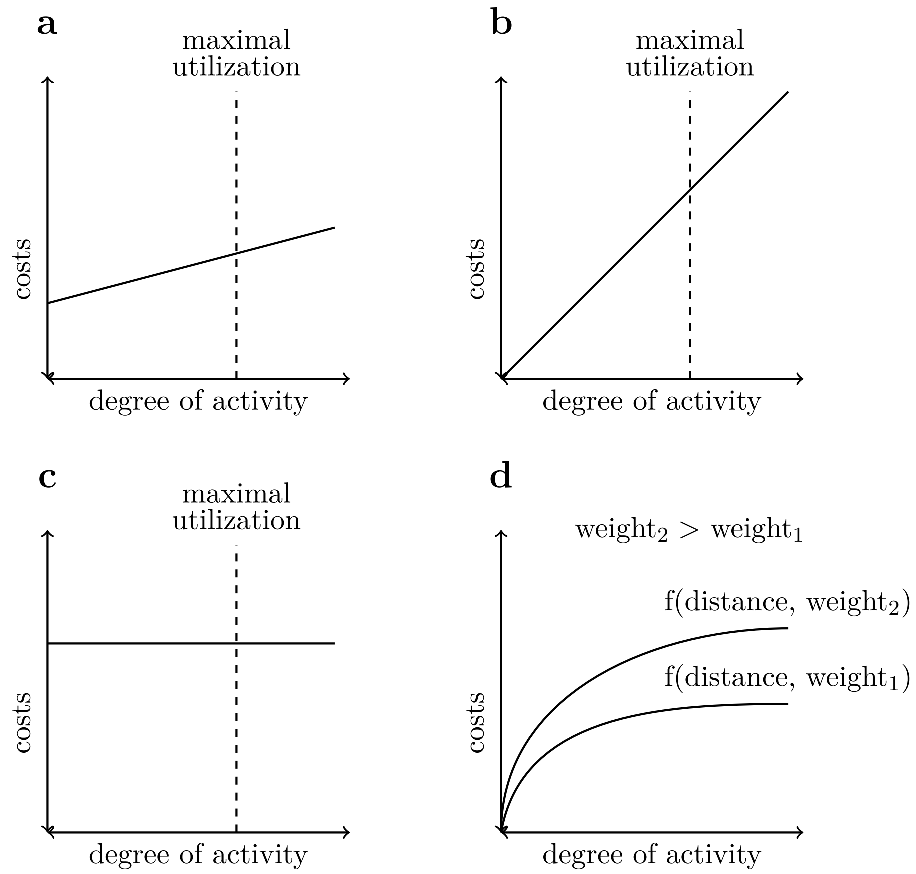


***Figure 5.5.:*** *Transportation costs of different fulfillment modes, cf. Krajewska and Kopfer (2009)*

tracting can be used for request fulfillment in IOTP. A formulation of an IOTP problem can be found by Wang et al. (2014), where the PDPTW formulation of Desaulniers et al. (2001) is extended to the IOTP problem. The formulation of the IOTP problem is similar to the PDPTW formulation in Subsection 5.2.1. An IOTP problem can be defined on a graph $G = (V, A)$, where $V$ represents the set of nodes and $A$ represents the set of edges with $A = V \times V$. An IOTP problem includes $n$ less than truckload requests and $m$ vehicles. Three types of vehicles are applicable for the fulfillment of a request. The set of individual and external vehicles $K$ with $K = \cup_{e=1}^{3} K_e$ is represented by the set of private vehicles ($K_1$), rented vehicles based on mode RB ($K_2$), and rented vehicles based on mode DB ($K_3$). Each vehicle $k \in K$ has the same capacity $Q$, while different fixed rates $\alpha_k$ and variable rates $\beta_k$ are charged. In terms of costs, private vehicles are paid by a variable rate ($\beta_1$) and a fixed rate ($\alpha_1$); rented vehicles on mode RB are paid by a variable rate ($\beta_2$); and rented vehicles on mode DB are paid by a fixed rate ($\alpha_3$). It is worth mentioning that the rented vehicles on mode DB have a maximal route length $L^{DB}$. The maximal route length cannot be exceeded. The third option of subcontracting is the employment of a common carrier (CC). A CC charges a fee $\gamma_i$ for fulfilling the request with pickup node $i$.

Five types of decision variables are used in an IOTP problem. The decision variables $x_{ijk}$, $w_{ik}$, and $L_{ik}$ are used as known from the description of the PDPTW. Moreover, $y_k^{DB}$ respectively $y_i^{CC}$ indicate whether a rented vehicle $k \in K_3$ on mode DB is used, respectively a CC is employed for the fulfillment of the request with pickup node $i$. Corresponding to Wang et al. (2014), an IOTP problem can be modeled as follows:

$$\min IP^{IOTP} = \sum_{k \in K_1 \cup K_2} \sum_{(i,j) \in A} \beta_k d_{ij} x_{ijk} + \sum_{k \in K_3} \alpha_k y_k^{DB} + \sum_{i \in P} \gamma_i y_i^{CC} \tag{5.16}$$

$$\text{s. t.} \quad \sum_{k \in K} \sum_{j \in V} x_{ijk} + y_i^{CC} = 1\,, \qquad \forall i \in P, \tag{5.17}$$

$$\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{j,n+i,k} = 0\,, \qquad \forall k \in K,\, \forall i \in P, \tag{5.18}$$

$$\sum_{j \in P \cup \{o\}} x_{ojk} = 1\,, \qquad \forall k \in K, \tag{5.19}$$

$$\sum_{i \in D \cup \{o\}} x_{iok} = 1\,, \qquad \forall k \in K, \tag{5.20}$$

$$\sum_{i \in V} x_{ijk} - \sum_{i \in V} x_{jik} = 0\,, \qquad \forall k \in K,\, \forall j \in V, \tag{5.21}$$

$$\sum_{j \in P} x_{ojk} = y_k^{DB}\,, \qquad \forall k \in K_3, \tag{5.22}$$

$$\sum_{(i,j) \in A} d_{ij} x_{ijk} \leq L^{DB}\,, \qquad \forall k \in K_3, \tag{5.23}$$

$$w_{ik} + s_i + t_{ij} - M(1 - x_{ijk}) \leq w_{jk}\,, \qquad \forall k \in K,\, \forall (i,j) \in A, \tag{5.24}$$

$$a_i \leq w_{ik} \leq b_i\,, \qquad \forall k \in K,\, \forall i \in V, \tag{5.25}$$

$$w_{ik} \leq w_{n+i,k}\,, \qquad \forall k \in K,\, \forall i \in P, \tag{5.26}$$

$$L_{ik} + l_j - M(1 - x_{ijk}) \leq L_{jk}\,, \qquad \forall k \in K,\, \forall (i,j) \in A, \tag{5.27}$$

$$L_{ik} \leq Q\,, \qquad \forall k \in K,\, \forall i \in V, \tag{5.28}$$

$$L_{ok} = 0\,, \qquad \forall k \in K, \tag{5.29}$$

$$x_{ijk} \in \{0,1\}\,, \qquad \forall k \in K,\, \forall (i,j) \in A, \tag{5.30}$$

$$y_k^{DB} \in \{0,1\}\,, \qquad \forall k \in K_3, \tag{5.31}$$

$$y_i^{CC} \in \{0,1\}\,, \qquad \forall i \in P, \tag{5.32}$$

$$w_{ik} \geq 0\,, \qquad \forall k \in K,\, \forall i \in V, \tag{5.33}$$

$$L_{ik} \geq 0\,, \qquad \forall k \in K,\, \forall i \in V. \tag{5.34}$$

The goal of an IOTP problem is to minimize the total transportation costs (5.16), which correspond to the fixed and variable costs for the different fulfillment modes and the sum of the freight charges paid to the CCs. The fixed cost term of the private fleet $\sum_{k \in K_1} \alpha_k$ is omitted in the objective function as in the previous subsection. By considering the problem formulation, an IOTP problem contains all constraints of a PDPTW. Thereby, constraints (5.2) are replaced by constraints (5.17), which ensure that either a private vehicle, a rented vehicle or a CC is used for fulfilling a request at a pickup node. Furthermore, the PDPTW has to be extended by constraints (5.22) by doing this it is ensured that the vehicle's fixed costs are included in the objective function in case that a rented vehicle on mode DB is used. The maximal route length of a rented vehicle on mode DB is limited by constraints (5.23) and the binary decision variables for mode DB and CC are defined by constraints (5.31) and (5.32), respectively.

The selection between self-fulfillment and subcontracting is a key decision for logistics managers because it affects customer service, operating expenses, capital investment, and managerial responsibilities (Min, 1998). Since the early 1970s, mode and carrier decisions have been the subject of investigations. Several publications investigate vehicle routing problems in combination with mode choice and carrier selection. These publications consider the requests to be independent and each of these requests can either be fulfilled by self-fulfillment or by subcontracting. Thereby, subcontracting is used when the private fleet capacity is exceeded or when it is less costly to do so (Côté and Potvin, 2009). Early research relies on the combination of different vehicle routing problems with the decision if either the private fleet or a CC should be used for the fulfillment (e.g., Ball et al., 1983; Agarwal, 1985; Klincewicz et al., 1990; Diaby and Ramesh, 1995). Recently, most of the relevant research refers to the extension of the well-known VRP where CCs are used for request fulfillment. The extended problem is known as the vehicle routing problem with private fleet and common carrier and was introduced by Chu (2005). In literature, different solution approaches are proposed for the vehicle routing problem with private fleet and common carrier, such as a modified savings heuristic (e.g., Chu, 2005; Bolduc et al., 2007, 2008), tabu search (e.g., Côté and Potvin, 2009; Potvin and Naud, 2011), genetic algorithm (e.g., Kratica et al., 2012), and variable neighborhood search heuristic (e.g., Stenger et al., 2013; Vidal et al., 2015). Simultaneously routing the private fleet and those from the long-term carriers represents a special case of the heterogeneous fleet vehicle routing problem. In the heterogeneous fleet vehicle routing problem, the fulfillment of requests is done by different types of vehicles. A literature overview can be found by Baldacci et al. (2008). An example for simultaneously routing the private fleet and those from the long-term carriers can be found by Ballou and Chowdhury (1980); Savelsbergh and Sol (1998). In Ballou and Chowdhury (1980), a TSP with pickup and delivery nodes is discussed where a request can be fulfilled by private vehicles or by any transportation service such as contract carriage, common carriage, and agency. As solution approach the Clarke-Wright savings method is applied. In Savelsbergh and Sol (1998), a PDP is studied, where some vehicles are rented permanently and some are rented on a daily basis for working periods. As solution approach a branch-and-price algorithm is applied. Another branch of research is given by considering the case where several types of subcontracting can be used for the fulfillment of requests. Few publications deal with this topic. In Krajewska and Kopfer (2009), the PDPTW is extended to the mentioned IOTP problem, which is solved by a tabu search. In Wang et al. (2014), the same IOTP problem is studied, where a CGB-heuristic is used. There the subproblem is solved by an ANLS and the master problem is solved by a commercial solver. An IOTP problem can also be found in Ceschia et al. (2011). There a more realistic vehicle routing problem with time windows (VRPTW) is studied by taking a heterogeneous vehicle fleet into account, where each vehicle belongs to one carrier. A tabu search is used for solving this TPP.

**Collaborative transportation planning:** A request can also be fulfilled by a freight forwarder of the horizontal coalition. In this context, the cooperation within a horizontal coalition can be understood as a "joint decision making process for aligning of individual supply chain members with the aim of achieving coordination in light of information asymmetry" (Stadtler, 2009). In practice, the topic of CTP is getting more and more important (Cruijssen et al., 2007), which is why new mathematical models and solution approaches are developed. Simultaneously solving the combined problem of vehicle routing for the private fleet by also exchanging transportation requests or capacities within a horizontal coalition is known as CTP (Wang and Kopfer, 2014). CTP is especially relevant for small and medium sized freight forwarders, which want to achieve economies of scales (e.g., extended request portfolio) and economies of scope (e.g., using vehicles with refrigerator units) by participating within a horizontal coalition. In general, the goal of CTP is to reduce the transportation costs of each freight forwarder by preserving the autonomy of the freight forwarders. A joint benefit should be achieved by using CTP and the benefit should be distributed equally among the freight forwarders of the coalition. Several computational studies confirmed the cost-saving potential of CTP. It is identified that the cost reduction accounts up to 30 % (Cruijssen and Salomon, 2004; Cruijssen et al., 2007; Krajewska et al., 2008) and that there is a decrement of the used vehicles between 7.3 % and 10 % (Cruijssen and Salomon, 2004; Krajewska et al., 2008). The cost-saving potential is determined by comparing the cost difference between isolated planning (IP) and centralized planning (CP). IP means that there is no exchange of requests and each freight forwarder determines his or her individual transportation schedule, while CP means that all companies act as a single entity. A general framework for CTP is proposed by Krajewska and Kopfer (2007), which enables the cost-saving potential and contains three phases: preprocessing, profit optimization, and profit sharing. They propose that the set of exchangeable requests and the corresponding payments should be determined before the profit optimization takes place. Afterward, the vehicle routing problem with the offered requests is solved and at the end the joint benefit is distributed among the freight forwarders of the coalition. This part focuses on the profit optimization phase of CTP.

Here, a CTP problem is described which considers the horizontal coalition of $m$ freight forwarders within the coalition $F$. The freight forwarders are in charge of less than truckload pickup and delivery requests. In the considered CTP problem, each freight forwarder $c \in F$ faces the described PDPTW of Subsection 5.2.1. Thereby, each freight forwarder $c \in F$ is in charge of an individual request portfolio $P_c$, which is offered for exchange within the coalition. It is assumed that each request can be fulfilled by any vehicle within the coalition. To fulfill a request, each freight forwarder $c \in F$ is in charge of an individual set of homogenous vehicles $K_c$, while there might be different types of vehicles within the coalition regarding the cost rates and loading capacities. Since there is no subcontracting in this CTP problem, it is assumed that a freight forwarder has enough loading capacity for fulfilling his or her individual request portfolio.

The described CTP problem can be solved by IP, CP, or collaborative planning. In case of IP, each freight forwarder $c \in F$ solves the PDPTW with his or her individual request portfolio $P_c$ and the total transportation costs can be determined by summing up the individual local costs of each freight forwarder within the coalition. In contrast to IP, CP assumes that there is just one multi depot PDPTW with the goal to reduce the total costs of the coalition instead of the individual costs of the freight forwarders. By solving the CTP problem of this thesis by collaborative planning, each freight forwarder $c \in F$ offers his or her entire request portfolio $P_c$ ($P = \sum_{c=1}^{m} P_c$) for exchange and receives a new portfolio $P'_c$ after the allocation process is completed. Depending on the transportation plans of the freight forwarders, each freight forwarder $c \in F$ transfers a set of requests ($P_c^-$) to the coalition and receives a set of requests ($P_c^+$) from the coalition. As the freight forwarders are self-interested they aim to minimize their individual costs (local costs) by preserving customer payments and cost structure information. The goal of this CTP problem is to minimize the local costs ($IP^{PDP}$) by considering the updated request portfolio $P'$. The problem can be modeled as follows (Wang and Kopfer, 2014):

$$\min CTP = \sum_{c=1}^{m} IP_c^{PDP} \tag{5.35}$$

$$P'_c \cap P'_j = \varnothing\,, \qquad \forall c, j = 1, ..., m,\; c \neq j, \tag{5.36}$$

$$\cup_{c=1}^{m} P_c^- = \cup_{c=1}^{m} P_c^+. \tag{5.37}$$

The goal of a CTP problem is to minimize the local costs of each freight forwarder of the coalition (objective (5.35)) by ensuring that each offered request is assigned to exactly one freight forwarder (equations (5.36)–(5.37)).

The existing literature on horizontal coalition focuses on the exchange of requests (Verdonck et al., 2013). In case of request exchange, two solution approaches are common, either freight forwarders offer all their requests or just a subset of their requests for exchange. Most of the publications focus on the former case, where a freight forwarder offers every request of his or her request portfolio (e.g., Cruijssen et al., 2007; Ergun et al., 2007; Clifton et al., 2008; Krajewska et al., 2008; Agarwal and Ergun, 2010; Liu et al., 2010; Wang and Kopfer, 2014). However, forwarding just a subset of their requests seems to be more realistic than the former one because freight forwarders do not want to reveal their entire request portfolio in a competitive environment. Recently, some publications have addressed this topic (e.g., Schwind et al., 2009; Berger and Bierwirth, 2010; Özener et al., 2011). In Berger and Bierwirth (2010) a TSP with precedence constraints and two auction mechanisms are presented, while a lane exchange among full-truckload carriers is presented in Özener et al. (2011). Schwind et al. (2009) consider a VRPTW with two different auction mechanisms. In general, these publications apply the mentioned cherry-picking procedure for identifying profitable requests for self-fulfillment and unprofitable ones for external freight forwarders. The corresponding outsourcing process can be done by exchanging single requests, request bundles or complete vehicle routes among the freight forwarders of the coalition. Single requests or request bundles are usually exchanged. In Wang and Kopfer (2014), a route-based exchange mechanism is proposed, where complete vehicle routes are exchanged within the coalition by a CGB-heuristic.

**Collaborative integrated operational transportation planning:** In the previous descriptions, it is always assumed that there is just one type of external resources either subcontracting or collaboration, but a freight forwarder can usually use all of the mentioned fulfillment modes: self-fulfillment, subcontracting, and collaboration for fulfilling a request. The combination of of the CTP and IOTP problem is denoted as CIOTP problem.

In the CIOTP problem, $m$ freight forwarders align their individual transportation plans by exchanging their requests with each other. Besides the option of exchanging requests, each freight forwarder $c \in F$ faces an IOTP problem, where a request can be fulfilled by self-fulfillment or subcontracting. In terms of the CC, it is assumed that each freight forwarder within the coalition pays the same freight charge. Corresponding to the previous statements regarding the CTP problem, the CIOTP problem can be solved by IP, CP or collaborative planning. In case of IP, each freight forwarder solves his or her IOTP problem and the total transportation costs are calculated as known, while in case of CP the CIOTP problem is solved as IOTP problem and in terms of collaborative planning the CIOTP problem is solved like the CTP problem except that each freight forwarder solves an IOTP problem instead of a PDPTW.

A lack of research is identified regarding the consideration of TPP with self-fulfillment, subcontracting, and collaboration as fulfillment modes at the same time. Krajewska and Kopfer (2007) describe this problem formally and introduce theoretical framework for solving this optimization problem. In Wang et al. (2014), a TPP based on a PDPTW with self-fulfillment, subcontracting, and collaboration is investigated, where a mathematical model is presented and solved by a CGB-heuristic.

## 5.3. Mandatory requests

In this section, the topic of mandatory requests in TPP is addressed. Thereby, the practical relevance as well as a literature review are presented. Regarding the literature overview, the publications which are based on this thesis are also included in this overview without describing them in detail. First of all, it is worth mentioning that two different denotations can be found in the literature regarding mandatory requests. They are denoted as compulsory (e.g., Schönberger, 2005; Ramaekers et al., 2016) or reserved requests (e.g., Li et al., 2016; Chen, 2016). In this thesis, the term mandatory requests is preferred because the term seems to be more appropriate and obvious than the other ones. One goal of this thesis is the determination of the impact of mandatory requests. Corresponding to Schönberger (2005), it is expected that mandatory requests lead to higher costs because additional restrictions often contradict with the goal of minimizing costs.

In practice, freight forwarders are in charge of requests which can be fulfilled by any fulfillment mode and requests which are prohibited to be fulfilled by certain fulfillment modes. The latter type of requests is denoted as mandatory requests. In literature, this topic is rarely addressed. There it is usually assumed that each request in a TPP can be fulfilled by any fulfillment mode. It means that each freight forwarder tries to minimize his or her transportation costs without taking care of which request is fulfilled by which fulfillment mode. However, this assumption represents a simplification because there are mandatory requests in practice. In recent years, some publications addressed this topic, while most of them just mention the existence of mandatory requests. The topic is mentioned by Schmidt (1989); Krajewska and Kopfer (2005); Jurczyk et al. (2006); Özener et al. (2011).

By taking a deeper look into the motivation for considering mandatory requests, it is obvious that the compulsiveness of requests is motivated by shippers and freight forwarders. For example, Schmidt (1989); Schönberger (2005); Özener et al. (2011) mention that contractual obligations by suppliers might be a reason for considering mandatory requests. Since requiring self-fulfillment from their freight forwarders can lead to advantages for shippers, it therefore justifies the prohibition. Corresponding to Farris II (2008); Vahrenkamp (2015), the following advantages can often be derived by using self-fulfillment instead of external resources for fulfilling requests:

- greater control and flexibility in terms of customer requirements,
- reduced damages and claims with experienced companies and drivers,
- option of using the truck as rolling billboard,
- lower risk of revealing customer details to competitors,
- appearance of the driver as a salesperson for products or services.

On the other hand, the consideration of mandatory requests is motivated by freight forwarders themselves, especially if some of their shippers are responsible for a large portion of their business (Özener et al., 2011). This might be explained by the fact that sensitive information about "valuable clients" represents a trade secret for freight forwarders, which should not be revealed to competitors (Vahrenkamp, 2015). Further reasons for considering mandatory requests are: long-term contracts, commercial reasons, trustiness, transportation of high quality goods, reliability concerns, and requirements of strategic clients (Krajewska and Kopfer, 2005; Jurczyk et al., 2006; Ziebuhr and Kopfer, 2014; Ramaekers et al., 2015, 2016). Thereby, it is worth mentioning that the topic of mandatory requests is especially relevant for small and medium sized shippers and freight forwarders, who have thin margins in a competitive environment (Vahrenkamp, 2015).

Few publications address a TPP with exchangeable and mandatory requests. Except for those publications which are written based on this thesis, just three publications present computational studies for a TPP with exchangeable and mandatory requests. That is why the topic of mandatory requests seems to be investigated insufficiently. The benefits of restricting outsourcing decisions are mentioned in several publications without presenting a detailed computational study. Table 5.1 presents an overview about the publications which deal with mandatory requests.

As can be seen by Table 5.1, 14 publications mention the topic of mandatory requests, while eight of them, including the five publications of this thesis, develop a mathematical model, where exchangeable and mandatory requests are considered and investigated by computational studies (e.g. Schönberger, 2005; Ziebuhr and Kopfer, 2014; Li et al., 2016; Ziebuhr and Kopfer, 2015, 2016a,b, 2017; Chen, 2016). Two of the remaining publications

***Table 5.1.:*** *Overview of mandatory requests (tasks) in TPPs*

| Publication | Mention mandatory tasks | Analyze mandatory tasks | Assume different mandatory tasks | Chapter |
|---|---|---|---|---|
| Schmidt (1989) | x | - | - | - |
| Krajewska and Kopfer (2005) | x | - | - | - |
| Schönberger (2005) | x | PDSP | - | - |
| Jurczyk et al. (2006) | x | - | - | - |
| Özener et al. (2011) | x | - | - | - |
| Ziebuhr and Kopfer (2014) | x | IOTP | x | 7 |
| Ramaekers et al. (2015) | x | - | - | - |
| Li et al. (2016) | x | PDSP | - | - |
| Ziebuhr and Kopfer (2016a) | x | IOTP | x | 7 |
| Ramaekers et al. (2016) | x | - | - | - |
| Ziebuhr and Kopfer (2015) | x | CTP | x | 8 |
| Chen (2016) | x | CTP | - | - |
| Ziebuhr and Kopfer (2016b) | x | CIOTP | x | 9 |
| Ziebuhr and Kopfer (2017) | x | CIOTP | x | 9 |

present a mathematical model and solution approach without presenting computational studies (Ramaekers et al., 2015, 2016). Most of the publications are published within the last two years and all of them use a PDPTW as a main vehicle routing problem. In detail, there is a pickup and delivery selection problem (PDSP) in Ramaekers et al. (2015); Li et al. (2016); Ramaekers et al. (2016), a PDSP with CCs in Schönberger (2005), an IOTP problem in Ziebuhr and Kopfer (2014, 2016a), a CTP problem in Ziebuhr and Kopfer (2015); Chen (2016) and a CIOTP problem in Ziebuhr and Kopfer (2016b, 2017). In terms of external publications, a common feature is that they consider one type of mandatory requests and this type can only be fulfilled by self-fulfillment, while the publications of this thesis consider different types of mandatory requests.

In Schönberger (2005), a capacitated PDSP is considered, which represents an extension of the PDSP. A PDSP is a PDP (here a PDPTW) where a freight forwarder has to decide whether to accept or decline a certain request depending on the profitability of the request. In contrast to the PDSP, the capacitated PDSP assumes that requests can be outsourced to a CC instead of not visiting a request like in the PDSP. A CC is employed for a request in exchange of a freight charge. Furthermore, the capacitated PDSP is extended by mandatory requests. Thereby, it is assumed that a CC cannot be employed for mandatory requests. The problem is denoted as pickup and delivery selection problem with compulsory requests (PDSP-CR). The goal of the PDSP-CR is to minimize the total costs by considering that each mandatory request is fulfilled by the private fleet. Schönberger (2005) applies a memetic algorithm, which is a genetic algorithm combined with a local search. In a memetic algorithm, each individual out of a population is locally optimized. Three modifications are proposed for handling mandatory requests: (1) penalization of mandatory requests, (2) double ranking approach, and (3) alternating and converging constraint approach. In approach (1), it is proposed that the freight charges are enlarged significantly for mandatory requests. Thereby, different ideas are analyzed like: static penalties, dynamic penalties, and adaptive penalties. In approach (2), it is proposed that the evolution process considers customer satisfaction (number of routed mandatory requests) and transportation costs. First, a feasible solution is aimed and second, low transportation costs are aimed without losing the feasibility. In approach (3), it is proposed that the second approach is modified in order that the ratio of feasibility is slightly increased in each iteration instead of generating a feasible solution right at the start. In a benchmark study, it is identified that the third approach is preferable for the PDSP-CR in terms of the solution quality even if the performance in terms of the feasibility is slightly worse compared to the second approach.

Ramaekers et al. (2015, 2016) discuss the topic of mandatory requests for a PDSP, where each mandatory request has to be accepted and cannot be declined, while the remaining requests can be neglected and remain unvisited. To solve this optimization problem, an extended tabu-embedded SA algorithm is applied, which was introduced by Li and Lim (2001). Thereby, the initial solution is generated by an insertion heuristic. Afterward, an improvement heuristic is applied. The heuristic keeps track of the solutions by a tabu list and an SA is used as an acceptance criterion. The main idea of this approach is the application of a reset procedure, where the solution approach returns to a previous solution as soon as there is no further improvement. In terms of the consideration of mandatory requests, it is proposed to use a kind of preinsertion phase, where mandatory requests are inserted in front of the remaining requests. This preinsertion phase ensures that mandatory requests can be fulfilled by the private fleet. One disadvantage of this research is that there are no computational studies, which verify the application of the solution approach for the described vehicle routing problem.

In Li et al. (2016), a PDSP like the one presented by Ramaekers et al. (2015, 2016) is solved and extended by the consideration of mandatory requests. One difference is that in Ramaekers et al. (2015) vehicles can remain unused at the depot and that it is assumed that there is an alliance which fulfills the unvisited requests. In this publication, just the bid generation problem is considered, where one freight forwarder solves the PDSP with the goal of maximizing his or her individual profits. Thereby, the freight forwarder is in charge of exchangeable and mandatory requests, where mandatory requests have to be fulfilled by the private fleet and exchangeable requests can remain unvisited. A modified ALNS of Ropke and Pisinger (2006) is applied to solve this problem. The ALNS is extended by four modifications regarding the mandatory requests: (1) reset approach with different initial solutions, (2) successive segments, (3) meta-destroy operators, and (4) modified destroy and insertion operators. The idea of the reset approach (1) is that the ALNS is repeated several times with different initial solutions. These initial solutions are generated based on different policies, where it is proposed that the mandatory requests with the highest profits are inserted in front of exchangeable requests. The policies differ in terms of the consideration of exchangeable requests. By using successive segments (2), it is proposed that the ALNS is split into several segments, where each segment contains several iterations with destroying and repairing a solution. In these segments, two different policies are used. There the idea is that at the beginning of each segment the solution space is extended by removing exchangeable and mandatory requests, while at the end the solution space is reduced by removing only exchangeable requests. A third approach is the meta-destroy operator (3), where it is proposed to apply two destroy operators instead of one when the local search of the ALNS is stuck in a local optimum.

Finally, the existing destroy and insertion operators (4) of Ropke and Pisinger (2006) are modified in terms of the mentioned policies and meta-destroy operator. Furthermore, two new destroy operators are added. In the evaluation section, two computational studies are carried out. Thereby, the instances are derived from the PDPTW instances from Ropke and Pisinger (2006), which are based on the PDPTW instances from Li and Lim (2001). In total, 54 instances are selected and extended by missing data (e.g., definition of mandatory requests and prices for requests). For example, each PDPTW instance is split into nine instances. Out of these nine instance, three sets with three instances are generated, which have 33.33 % mandatory requests, 50 % mandatory requests, or 66.66 % mandatory requests. However, it is not specified how the selection, which requests is a mandatory or exchangeable request, is performed. In a benchmark study, the PDSP with mandatory requests is solved by the solver CPLEX and the modified ALNS.

Chen (2016) proposes a CTP problem where multiple freight forwarders exchange their pickup and delivery requests. There each freight forwarder considers a PDPTW, where some requests (i.e., exchangeable requests) can be exchanged with other freight forwarders within the coalition. Moreover, there are some requests (i.e., mandatory requests) which have to be fulfilled by the freight forwarder itself due to contractual obligations with the customer. A combinatorial clock-proxy auction is used to solve this problem, which uses two phases. It starts with the clock phase, which is an iterative auction procedure, where an auctioneer announces prices for each exchangeable request and based on these data each freight forwarder determines his or her requests to offer and buy. This problem is solved by CPLEX or a CGB-heuristic with a simple local search for solving the subproblem. The CGB-heuristic is used to solve large instances. In the second phase, the proxy phase takes place, which is a package bidding procedure, where proxy agents submit package bids to the auctioneer based on the bidders data. This problem (i.e., winner determination problem) is solved by CPLEX. In an evaluation section, the performance of the solution approach is evaluated based on new test instances. Two instance sets are considered, each of them contains ten instances with a coalition of three freight forwarders. The small instance set contains two subsets: one with 15 and 24 requests, while the large instance set contains three subsets: one with 45, 90, and 150 requests. In terms of the definition of mandatory requests, each freight forwarder solves his or her PDPTW and all profitable requests are defined as mandatory requests. In one computational study, three different ratios of mandatory requests are solved for the small instance set by presenting the individual profit of each carrier without presenting any further investigation regarding the mandatory requests.

In all of these publications, mandatory requests are selected due to contractual obligations, while the remaining ones are selected due to their profitability. This means that a freight forwarder has to use a certain fulfillment mode for a request with contractual obligations, while the remaining requests can be served by the most profitable fulfillment mode. All of the mentioned publications assume that mandatory requests have to be fulfilled by self-fulfillment, which means that the employment of CCs, long-term carriers or freight forwarders is prohibited. In this thesis, a request is denoted as a mandatory request in case that the request is limited in terms of the fulfillment modes and a common feature is that at least a CC cannot be employed for the request. Due to this extension, it is possible to analyze up to four different types of mandatory requests. The idea of considering different types of mandatory requests is motivated by the fact that a freight forwarder should be able to offer various services to a shipper and each of these services may differ with respect to its impact on transportation costs and plans. The following transportation services are considered:

- $P1$ service: self-fulfillment,
- $P2$ service: self-fulfillment, long-term carrier,
- $P3$ service: self-fulfillment, collaboration,
- $P4$ service: self-fulfillment, long-term carrier, collaboration,
- $EX$ service: self-fulfillment, subcontracting, collaboration.

The corresponding requests of the mentioned services are denoted as $EX$, $P1$, $P2$, $P3$, and $P4$ requests. For example, the $P3$ service demands the application of a vehicle of the private fleet of any freight forwarder within the horizontal coalition, while the application of subcontracting is prohibited. All $P$ requests are mandatory requests and have a common feature of being unable to use CCs. The remaining publications in Table 5.1 are part of this thesis and will be explained in the corresponding chapters.

## 5.4. Column generation-based heuristic

Here, the basic solution approach of the second part of the thesis is presented. The solution approach was introduced by Wang et al. (2014) and is known as CGB-heuristic. The CGB-heuristic is based on a column generation which is devised for linear programs and is a popular solution approach for solving large scale integer programming problems (Lübbecke and Desrosiers, 2004). The idea of a column generation is the reformulation of the original problem into two problems: a master problem (selection of columns) and subproblem (generation of columns). In the following, the CGB-heuristic is described more in detail. Thereby, it is worth mentioning that the presented CGB-heuristic is not able to solve TPP with mandatory requests. Therefore, it is necessary to extend the solution approach by solution strategies for handling mandatory requests, which are presented in Chapter 6.

The CGB-heuristic is described for a TPP with pickup and delivery nodes. The individual adjustments for the different TPPs including the vehicle routing problems are discussed in the corresponding Chapters 7, 8, and 9. A general overview of the CGB-heuristic is given in Subsection 5.4.1, while the column generation is explained in Subsection 5.4.2 and the ALNS is described in Subsection 5.4.3.

### 5.4.1. General overview

The applied solution approach uses a column generation, which is often applied to solve TPPs including the vehicle routing problems. To be precise, the CGB-heuristic of Wang et al. (2014) is used. There a metaheuristic is used for identifying promising vehicle routes and a solver is used for generating a feasible TPP solution based on the submitted vehicle routes. An overview of the CGB-heuristic is given by Figure 5.6. It can be seen that several steps have to be executed before a TPP solution is identified.

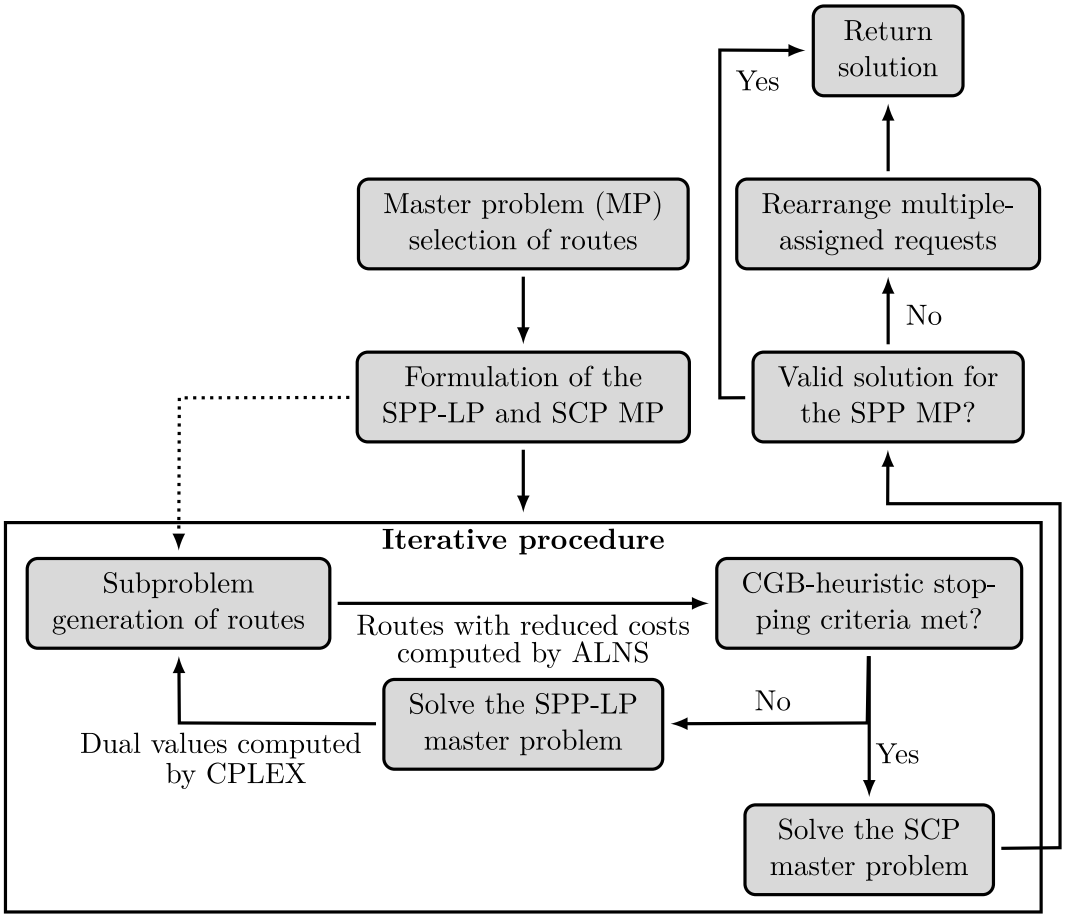


***Figure 5.6.:*** *Overview of the CGB-heuristic, cf. Ziebuhr and Kopfer (2016a)*

At the beginning of the CGB-heuristic, the TPP is formulated as set partitioning problem (SPP) master problem. This formulation is responsible for selecting the best columns out of the generated columns in order that all selected columns represent a feasible transportation plan for the TPP. Here, a column is either a vehicle route fulfilling the loading, time, and routing constraints of the TPP or an outsourced request fulfilled by a CC. Then, the master problem is formulated as a linear relaxed SPP (SPP-LP) and set covering problem (SCP). Both formulations are used within the iterative procedure of the column generation. Afterward, the iterative procedure of the CGB-heuristic is initiated by solving the subproblem of the TPP. Thereby, an initial set of vehicle routes is generated. which contains feasible vehicle routes. These vehicle routes have the potential to reduce the transportation costs and this set of vehicle routes is always supplemented by a set where each request is served by a CC. A feasible solution can be ensured due to this procedure. The subproblem is solved by a slightly modified ALNS of Ropke and Pisinger (2006), where each freight forwarder solves the subproblem on his or her own. Every time the subproblem is solved, it is determined whether one of the stopping criteria of the CGB-heuristic is met. In case that none of the stopping criteria is met, the SPP-LP master problem is solved by the solver CPLEX, which identifies a transportation plan and corresponding dual values associated with the requests and vehicles. The master problem is solved by a mediator. These dual values are used to update the data of the subproblem and can thereby support the local search in terms of identifying more promising vehicle routes, which are able to reduce the transportation costs. This procedure of generating and selecting vehicle routes is repeated. As soon as one stopping criterion is met, the best vehicle routes are forwarded to the SCP master problem. Due to this procedure, it is possible that requests can be assigned to several fulfillment modes. Nevertheless, this procedure is highly recommended because a solution of the SPP-LP master problem contains fractional values in terms of the vehicle routes and is usually difficult to repair as a feasible solution of the SPP master problem. As soon as the SCP master problem is solved, it is determined whether the solution is feasible for the SPP master problem or not. In case that requests are served more than once (i.e., multiple assigned requests), a mechanism takes place which deletes these multiple assigned requests and reinserts them into the best available position of the vehicle routes by using the so called regret-k heuristic from Ropke and Pisinger (2006) and keeping the remaining part of the solution unchanged. Finally, a feasible solution for the TPP is identified.

In Wang et al. (2014), two versions of the CGB-heuristic are presented, one with a homogenous vehicle fleet and one with a heterogeneous vehicle fleet. The difference is that the former one solves the TPP for the entire vehicle fleet, while the latter one solves it separately for every fulfillment mode. In Chapter 6, it is investigated which version is more suitable to solve a TPP with exchangeable and mandatory requests.

### 5.4.2. Column generation

The applied CGB-heuristic uses three formulations of the master problem: SPP, SPP-LP, and SCP formulation. The SPP formulation represents the TPP, which has to be relaxed to the SPP-LP formulation for the iterative procedure. The SCP formulation is used at the end of the iterative procedure with the goal of reducing the computational effort for identifying a feasible solution for the SPP master problem.

In the SPP master problem, the set of vehicle routes is defined by the variable $R$, which includes up to three subsets depending on the considered TPP. In case of the IOTP and CIOTP problem, the set of vehicle routes of the freight forwarder $c$ contains: the set of routes fulfilled by self-fulfillment ($R_c^1$), the set of routes fulfilled by route basis ($R_c^2$), and the set of routes fulfilled by daily basis ($R_c^3$), i.e., $R_c = \cup_{e=1}^{3} R_c^e$. Thereby, each set of vehicle routes belongs to a fulfillment mode $e \in E$ with up to $E = \{1, 2, 3\}$ where each set has a specific fleet size defined by the known vehicle set $K = \cup_{e=1}^{3} K_c^e$. In case of the CTP problem, the set of vehicle routes is equal to the set of routes fulfilled by self-fulfillment. The transportation costs, i.e., the route costs $c_r$ for each vehicle route $r \in R_c$ and the freight charge for requests forwarded to CCs are defined by the fulfillment mode specific cost rates as specified by objective function (5.16). The route costs $c_r$ does not include the fixed costs for the private fleet.

Two indicators and two decision variables are used in the SPP formulation. In case that the vehicle route $r$ belongs to the fulfillment mode $e$, the mode indicator $f_{er} \in \{0,1\}$ is one and zero otherwise. A second indicator $a_{ir} \in \{0,1\}$ defines whether the pickup node $i$ is served by the route $r$ ($a_{ir} = 1$) or not ($a_{ir} = 0$). The decision variable $u_r$ indicates whether the vehicle route $r$ is selected by the mediator for being an element in the set of the best routes, while the variable $u_i^{CC}$ defines whether a CC is employed for the fulfillment of the request at pickup node $i$ or not. The SPP master problem can be formulated as follows (Wang et al., 2014):

$$\min C_c = \sum_{r \in R_c} c_r u_r + \sum_{i \in P_c} \gamma_p u_i^{CC} \tag{5.38}$$

$$\text{s. t.} \quad \sum_{r \in R_c} a_{ir} u_r + u_i^{CC} = 1\,, \qquad \forall i \in P_c, \tag{5.39}$$

$$\sum_{r \in R_c} f_{er} u_r \leq |K_c^e|\,, \qquad \forall e \in E, \tag{5.40}$$

$$u_r \in \{0,1\}\,, \qquad \forall r \in R_c. \tag{5.41}$$

The goal of the SPP master problem, which is defined by objective function (5.38), is to minimize the total transportation costs (i.e., the sum of freight charges and tour execution costs). Constraints (5.39) ensure that every request is served either by a vehicle route or by a CC. The fleet size per fulfillment mode is limited by constraints (5.40) and binary decision variables $u_r$ are defined by constraints (5.41). In a column generation, a master problem is usually solved by the simplex method. That it why it is proposed to relax the binary variable $u_r$ of the SPP master problem to a continuous variable by replacing constraints (5.41) by $u_r \geq 0$. The resulting linear relaxed master problem is denoted as the mentioned SPP-LP master problem. Based on this linear relaxed formulation, the dual variables $\pi$ and $\sigma$ can be derived from constraints (5.39) and (5.40), respectively. The values of the dual variables are used to guide the local search of the applied ALNS. For example, in case that a certain request is selected quite often in the set of best routes (measured by constraints (5.39)), a dual value is generated which ensures that the request is less attractive for vehicles in the next iteration of the CGB-heuristic and vice versa for the remaining scenario. To get the mentioned SCP master problem, the SPP master problem is updated by modifying constraints (5.39) in order that more than one vehicle or CC can be selected for a single request ($\sum_{r \in R} a_{ir} u_r + u_i^{CC} \geq 1\,, \forall i \in P$).

The subproblem is solved for identifying promising, feasible vehicle routes. In a minimization problem, a promising vehicle route for vehicle $k$ is one with negative reduced costs $c_k' = c_k - \sum_{i \in P_c} \pi_i a_{ik} - \sum_{e \in E} \sigma^e f_{ek}$ which fulfills constraints (5.17)–(5.34) of the IOTP problem or constraints (5.2)–(5.15) of the PDPTW depending on the considered TPP. The goal of a subproblem is usually the identification of one column (i.e., vehicle route) by solving the optimization problem with the objective of minimizing $c_k'$. However, Wang et al. (2014) propose that several vehicle routes are generated simultaneously by minimizing the total reduced costs of all its columns. The corresponding objective function can be formulated as follows (Wang et al., 2014):

$$\min \sum_{k \in K_c} c'_k = \sum_{k \in K_c} c_k - \sum_{k \in K_c} \sum_{i \in P_c} \pi_i a_{ik} - \sum_{e \in E} \sum_{k \in K_c} \sigma^e f_{ek} \tag{5.42}$$

The total reduced costs of a freight forwarder include the total route costs of the corresponding objective function, while the other terms sum up the dual values. By reformulating this objective function, it is possible to generate the same objective function as the corresponding objective functions of the different TPPs. The reformulation is described in detail by Wang et al. (2014). Due to this observation, it is possible to use the ALNS for identifying new vehicle routes.

Several vehicle routes are generated by solving the subproblem. It is obvious that it is impossible to use all vehicle routes for the master problem because the computational effort is too high. That is why a subset $R'_c \supset R_c$ is generated which contains the best, feasible vehicle routes. The size of the subset $R'_c$ (i.e., number of recorded solutions) is defined in advance based on computational studies. Thereby, it is worth mentioning that a vehicle route in $R'_c$ might exist in more than one fulfillment mode because vehicle routes are transferred into a different mode if it is possible. This procedure is not used in case of the CTP problem, because there are no long-term carriers in this scenario.

The described CGB-heuristic has to be extended in case of the consideration of different freight forwarders within a horizontal coalition (i.e., CTP and CIOTP problem). Most of the modifications are minor and related to the data exchange among the freight forwarders and the mediator. The freight forwarders are in charge of their subproblem, while the mediator deals with the master problem. Here, it is assumed that each freight forwarder offers all of his or her requests for exchange, which means that each freight forwarder solves his or her subproblem by considering all requests of the coalition. The objective function (5.42) remains as described except for the fact that the request set $R_c$ is replaced by the set $R$. In terms of the submission of the vehicle routes, each freight forwarder submits the generated vehicle routes including their corresponding costs to the mediator and the maximum number of requests which he or she can fulfill. Then, the mediator solves the master problem by identifying the most promising vehicle routes out of all submitted routes. By generating the dual values, there is one modification in case of the dual value $\pi$, which is submitted in a revised form of $\pi'_i = max\{0, \pi_i\}, \forall i \in P$. Another modification can be found by the final repair mechanism in order that the mediator assigns a multi assigned request to this freight forwarder who has won more multiple assigned requests. As soon as the CGB-heuristic is finished, the chosen bundles are declared as winning bundles and the mediator pays the corresponding costs to the freight forwarders. In case that a CC is selected for fulfilling a request, the mediator returns the request to the corresponding freight forwarder, where he or she is responsible for employing a CC by getting the fee from the mediator. Finally, collaborative savings can be distributed. However, the topic of distributing cost savings is not addressed in this part of the thesis.

### 5.4.3. Adaptive large neighborhood search

The subproblem with objective function (5.42) is solved by an ALNS. This metaheuristic, which was introduced by Ropke and Pisinger (2006), represents an extension of the mentioned LNS. The idea of an LNS is to perform large moves by rearranging up to 30 %–40 % of all requests in a single iteration, which allows to move from one promising solution area to another one. The LNS was introduced by Shaw (1997), who applied the metaheuristic for the VRPTW, while Bent and Hentenryck (2006) applied it to solve a PDPTW and Ropke and Pisinger (2006) extended the LNS to the ALNS.

An LNS is a local search, where a new neighborhood is generated by removing and inserting requests out of a solution. Depending on an acceptance criterion, it is decided if a new solution should be accepted or not. As acceptance criterion the metropolis acceptance criterion of the SA is often applied. The described procedure of removing and inserting requests is visualized by Figure 5.7, which is explained based on a VRPTW.

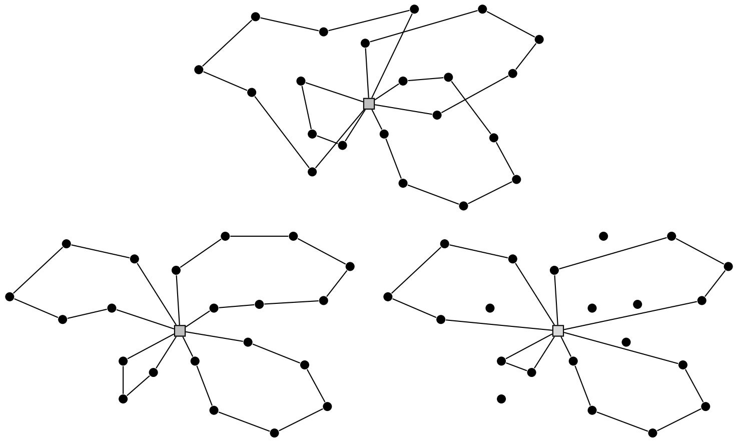

***Figure 5.7.:*** *Idea of the LNS, cf. Pisinger and Ropke (2010)*

At the top figure, a feasible VRPTW solution is presented. An LNS proposes to destroy the solution by removing requests of the solution (bottom right figure). Afterward, the solution is rebuilt by reinserting the removed requests at different positions in the vehicle routes (bottom left figure). This procedure is repeated several times by an LNS. The difference between an LNS and an ALNS is that an ALNS uses several competing removal and insertion operators during the local search depending on their historic performance. An overview about the ALNS of Ropke and Pisinger (2006) can be found in Alg. 11.

**General overview:** At the beginning of the applied ALNS, the initial solution $s$ and the scope of the search $q$ have to be determined. Then, the initial solution $s$ is set to the best solution $s_b$ and the temperature $T$ is set to the start temperature $T_0$. The start temperature is calculated by setting the value in order that a solution that is 5 % worse (based on the total variable route costs) than the initial solution is accepted with a probability of 50 %. Afterward, the ALNS tries to optimize the best solution $s_b$ by using different removal and insertion operators in an iterative procedure with $it_{\max}$ iterations. Thereby, the current solution $s'$ is initialized by the solution $s$. Then, the removal and insertion operator have to be determined according to their weights. The weight of each operator depends on the

**Algorithm 11:** Procedure of an ALNS, cf. Wang (2014)

**Data**: problem data, initial solution $s$, scope of the search $q$
1 initialize best solution $s_b \leftarrow s$
2 initialize temperature $T_c \leftarrow T_0$
3 **for** $i \leq it_{max}$ **do**
4     update current solution $s' \leftarrow s$
5     choose removal and insertion operator according to their weight
6     remove $q$ requests from $s'$ using the chosen removal operator
7     reinsert requests into $s'$ using the chosen insertion operator
8     **if** $f_{s'} < f_{s_b}$ **then**
9         update best solution $s_b \leftarrow s'$
10     **end**
11     **if** $f_{s'} \leq f_s$ **then**
12         update solution $s \leftarrow s'$
13     **end**
14     **else if** $random[0,1] < e^{-\frac{f_{s'} - f_s}{T}}$ **then**
15         update solution $s \leftarrow s'$
16     **end**
17     update temperature $T_c \leftarrow T_{c-1} \cdot$ cooling rate
18     **if** *weight update is activated* **then**
19         update weights of operators
20     **end**
21 **end**
22 **return** $s_b$ best solution

historic performance of the operator during the local search. In line 6 of Alg. 11, $q$ requests are removed from solution $s'$ by the chosen operator and reinserted in line 7 of Alg. 11. The remaining part of the algorithm evaluates the solution $s'$. If the solution $s'$ has lower costs than the best solution $s_b$, the best solution will be updated. If the solution $s'$ is accepted by the known metropolis acceptance criterion (line 11-16 of Alg. 11), the solution $s'$ will be used as new current solution for the next iteration. In line 17 of Alg. 11, the temperature variable $T_c$ is updated by decreasing the value according to a cooling rate. The cooling rate is generated in the range of $(0, 1)$. After a certain number of iterations, the weights of the operators are updated. The described iterative procedure is repeated until $it_{\max}$ iterations are executed. Finally, the best solution $s_b$ is returned.

**Removal operators:** In line 7 of Alg. 11, a removal operator is applied, which destroys a solution by eliminating a predefined number of requests out of the solution. Ropke and Pisinger (2006) propose three removal operators: Shaw removal, random removal, and worst removal. Each of these operators uses as an input a solution and the scope of the search. Furthermore, all of them use a random factor for avoiding getting stuck in a local optimum.

The Shaw removal operator, which was introduced by Shaw (1997), proposes to remove similar transportation requests from a solution. The idea of this operator is that it might be easier to replace similar requests with each other instead of requests with different characteristics because in this case requests would probably be reinserted at their original position. The similarity of two requests $i$ and $j$ is evaluated by the relatedness measure $R(i,j)$. The pickup and delivery location of request $i$ are given by the parameter $P(i)$ or $D(i)$, respectively. Furthermore, the parameter $w_{P(i)}$ defines the time when the service starts at the pickup location of request $i$, while $K_i$ represents the set of vehicles, which can serve request $i$. The relatedness measure $R(i,j)$ can be calculated as follows (Ropke and Pisinger, 2006):

$$R(i,j) = \phi(d_{P(i),P(j)} + d_{D(i),D(j)}) + \chi(|w_{P(i)} - w_{P(j)}| + |w_{D(i)} - w_{D(j)}|) + \psi|l_i - l_j| + \omega\left(1 - \frac{|K_i \cap K_j|}{\min\{|K_i|, |K_j|\}}\right) \quad (5.43)$$

As can be seen, the similarity between the requests $i$ and $j$ depends on four terms: a distance term, a time term, a capacity term, and a term that considers which vehicles can be used to serve the requests $i$ and $j$. Thereby, it is proposed that the first three parameters $d_{ij}$, $w_i$ and $l_i$ are normalized by scaling the parameters in the interval of $[0,1]$. Furthermore, each of the four terms is weighted by a predefined factor: $\phi, \chi, \psi$, and $\omega$. The goal of the Shaw removal operator is to minimize the relatedness measure $R(i,j)$.

A second operator is the random removal operator, which selects requests randomly and removes them from the solution. The idea of this approach is the reduction of the computational effort. A third operator is the worst removal operator which removes the requests with the highest costs. The costs of the request $i$ in the solution $s$ can be calculated as the difference of the solution with and without the request $i$, e.g., $cost(i,s) = f(s) - f_{-i}(s)$. The idea of this approach is to remove these requests which might be placed in a wrong position in the solution.

**Insertion operators:** As soon as a solution is destroyed, it is necessary to rebuild the solution by using an insertion operator in line 7 of Alg. 11. In Ropke and Pisinger (2006), two parallel insertion operators are suggested which construct several vehicle routes at the same time. The following insertion operators are proposed: basic greedy heuristic and regret heuristic. In general, it is worth mentioning that the insertion operator represents this part of the ALNS which has the highest computational effort.

In the basic greedy heuristic (also known as regret-1 heuristic), the set of unplanned requests $U$ is inserted at the position of a vehicle which increases the insertion costs the least. The insertion cost $c_{ik}$ represents the increment of the route costs after request $i$ is inserted at the best position within a vehicle route $k$. The costs of inserting request $i$ at its best position overall is defined by the variable $c_i$. First, the operator orders each request $i \in U$ in accordance to their cost $c_i$ and second, inserts the current request $i$ with lowest insertion costs overall in each iteration. One main disadvantage of this operator is that it is difficult to find good insertion positions for requests with high minimum insertion costs.

A more comprehensive approach is given by the regret heuristic, which consider $k$'th best insertion positions of a request instead of just the best one. Therefore, the variable $x_{ik} = 1, 2, ..., |K|$ is introduced, which indicates the route with the $k$'th lowest insertion costs of the request $i$. The variable $c_{ik}$ can be rewritten as $c_{i,x_{ik}}$. In the regret heuristic, the goal is to identify the request $i$ with the highest regret value $c_i^*$ which considers up to $k$'th best insertion positions of a request and determines the difference of the insertion costs between the best route and its $k$'th best route. For example, in a regret-2 heuristic the regret value $c_i^*$ can be determined by calculating the difference of the insertion costs of the best and second best route, i.e., $c_i^* = c_{ix_{i2}} - c_{ix_{i1}}$. A generalization of the regret heuristic is the regret-$k'$ heuristic which chooses the request $i$ by considering $k'$ best insertion positions that maximizes (Ropke and Pisinger, 2006):

$$\max_{i \in U} \{ \sum_{j=1}^{k'} (c_{i,x_{ij}} - c_{i,x_{i1}}) \} \tag{5.44}$$

Depending on the scope of the search, it might appear that some requests cannot be inserted into at $|K| - q + 1$ routes, then the request, which can be inserted in the fewest routes, is inserted first. In case of ties, it is proposed to select the request with best insertion costs overall routes $c_i$. In the applied ALNS, the basic greedy (regret-1), regret-2, regret-3, regret-4, and regret-|K| heuristics are applied as insertion operators.

**Noise:** In Ropke and Pisinger (2006), it is proposed to randomize the insertion operator by adding a noise term to the insertion costs of a request into a route. Every time the insertion costs of a request is determined, a random value noise in the interval $[-maxN, maxN]$ is calculated and added on the insertion costs $c' = max\{0, c + noise\}$. The amount of noise depends on a control parameter $\eta$ and the maximum distance between two nodes within an instance. It can be calculated by $maxN = \eta \cdot max_{i,j \in V}(d_{ij})$. It is worth mentioning that it is decided for each iteration separately if the noise term is used or not. This decision is taken by an adaptive mechanism.

**Roulette wheel selection principle:** In an LNS, just one removal and insertion operator is selected throughout the local search. However, it is observed that each of the mentioned operators performs differently depending on the instance. That is why Ropke and Pisinger (2006) propose the application of different operators throughout the local search with the goal to install a more robust heuristic. Therefore, a roulette wheel selection principle is proposed, where each operator $i \in \{1, 2, ..., k\}$ has an individual weight $w_i$ and the probability for accepting the operator $j$ can be calculated as follows (Ropke and Pisinger, 2006):

$$\frac{w_j}{\sum_{i=1}^{k} w_i} \tag{5.45}$$

It is worth mentioning that the selection of the removal and insertion operator are executed separately from each other and that there is a separate roulette wheel for the removal and insertion operator. At the beginning of the ALNS, each operator has the same weight, which is updated after every segment (one segment represents 100 iterations). The weight of an operator $i$ for the segment $j+1$ can be calculated as follows (Ropke and Pisinger, 2006):

$$w_{i,j+1} = w_{ij}(1 - r_z) + r_z \frac{\pi_i}{\theta_i} \tag{5.46}$$

The variable $\pi_i$ measures the score of an operator by three parameters, which depend on the historic performance. The score will be increased by predefined amounts ($\sigma_1, \sigma_2$, and $\sigma_3$) if (1) a new global best solution is identified, (2) a new solution with lower costs than the current one is identified, and (3) a new solution with higher costs than the current one is identified which was accepted by the metropolis acceptance criterion. In terms of the second and third scenario, it is further necessary that a new solution was not accepted before. The parameter $\theta_i$ represents the number of times how often the ALNS attempted to use the operator $i$ in the last segment. Furthermore, there is a reaction factor $r_z \in [0, 1]$ which defines the relevance of the score.

**Modifications for the CGB-heuristic:** The described ALNS of Ropke and Pisinger (2006) is slightly modified to be suitable for the CGB-heuristic. First, it is proposed to use cost savings ($\gamma_i - c_{ik}$) instead of insertion costs ($c_{ik}$) for the insertion operators and that requests with negative cost savings can also be inserted into vehicle route in case that they are lower than a threshold parameter $h$ multiplied by the current temperature $T_c$, i.e., $c_{ik} - \gamma_i \leq h \cdot T_c$. Second, a fictive variable cost rate is used for the rented vehicles on mode DB because otherwise the insertion costs for any request in an empty vehicle would be the flat rate.

# 6. Column generation-based Heuristic with Solution Strategies

In the previous chapter, the existing CGB-heuristic of Wang et al. (2014) is presented. This solution approach can be applied to solve different TPP including the corresponding vehicle routing problems (e.g., IOTP, CTP, and CIOTP problems) where requests can be fulfilled by any fulfillment mode. This chapter describes how the CGB-heuristic can be modified in order to solve TPP with exchangeable and mandatory requests.

An existing code of the CGB-heuristic is used in this thesis, which is originally designed for a dynamic CTP problem with full-truckload pickup and delivery requests by Wang and Kopfer (2015). This code is modified in order to solve static TPPs with less-than-truckload pickup and delivery requests. Thereby, it is ensured that the modified code is comparable with the one presented by Wang et al. (2014) for IOTP, CTP, and CIOTP.

This chapter introduces solution strategies for the CGB-heuristic. These solution strategies are able to deal with TPP where some requests have to be fulfilled by certain fulfillment modes. The following solution strategies are proposed: strict generation procedure, strict composition procedure, and repair procedure. These solution strategies differ with respect to the part of the iterative procedure of the CGB-heuristic which is chosen for taking the forwarding limitations into account. Forwarding limitations mean that there are limitations in terms of the fulfillment of requests. The strict generation procedure considers forwarding limitations by solving the subproblem, while the strict composition procedure considers them by solving the SPP-LP or SCP master problem. The repair procedure installs feasibility for TPPs with forwarding limitations as soon as the CGB-heuristic is executed.

This chapter is separated into two sections. It mainly explains and evaluates the solution strategies based on the IOTP problem with forwarding limitations (IOTPP-FL). The IOTPP-FL considers the mentioned request types: $EX$, $P1$, and $P2$ requests. As known $P1$ and $P2$ requests are mandatory requests. In Section 6.1, the solution strategies for handling these mandatory requests are presented, while computational studies are presented in Section 6.2. These studies evaluate different versions of the CGB-heuristic with each other for the IOTPP-FL and compare the performance with an existing solution approach from literature. This chapter is based on Ziebuhr and Kopfer (2014) as well as Ziebuhr and Kopfer (2016a).

## 6.1. Solution strategies

For solving an IOTPP-FL, the CGB-heuristic is extended by different solution strategies. The extensions for the remaining TPPs like the CTP and CIOTP problem are described in the corresponding chapters. The strict generation procedure is described in Subsection 6.1.1. The strict composition procedure and the repair procedure are explained in Subsection 6.1.2 and Subsection 6.1.3, respectively.

### 6.1.1. Strict generation procedure

The strict generation procedure considers forwarding limitations by solving the subproblem of the CGB-heuristic while the master problem remains unchanged. It means that the strict generation procedure solves the subproblem related to the IOTPP-FL and the SPP-LP or SCP master problem related to the IOTP problem (IOTPP). The structure of this solution strategy is presented in Figure 6.1.

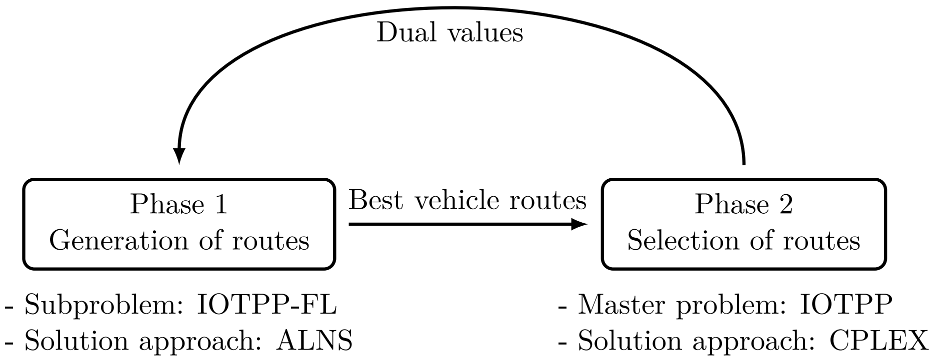


***Figure 6.1.:*** *Structure of the strict generation procedure, cf. Ziebuhr and Kopfer (2016a)*

As indicated in Figure 6.1, the compulsiveness of requests is strictly observed by the subproblem. The idea is that only "feasible vehicle routes" are accepted by the SA during the local search of the ALNS. In this scenario, the term feasible means that a request is fulfilled by a fulfillment mode corresponding to the type of service attached to the request. For example, that $P1$ requests are fulfilled by the private fleet and $P2$ requests are fulfilled by the private or rented fleet. As soon as the subproblem is solved, the best feasible vehicle routes are submitted to the SPP-LP or SCP master problem. Because all submitted routes are feasible for the IOTPP-FL, it is not necessary to consider forwarding limitations by solving the SPP-LP or SCP master problem. Consequently, the master problem can be solved as presented in Section 5.4.

In terms of the subproblem, an IOTPP-FL has to be solved. A formulation of the IOTPP-FL can be found by using the objective function (5.16) and constraints (5.17)–(5.34) of the IOTPP and replacing constraints (5.17) by the following constraints, which define the applicable fulfillment modes for each request type (Ziebuhr and Kopfer, 2014):

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} + y_i^{CC} = 1\,, \qquad \forall i \in EX, \tag{6.1}$$

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1\,, \qquad \forall i \in P2, \tag{6.2}$$

$$\sum_{k \in K_1} \sum_{j \in V} x_{ijk} = 1\,, \qquad \forall i \in P1. \tag{6.3}$$

Constraints (6.1) ensure that either a private vehicle, a rented vehicle or a CC is used for fulfilling a $EX$ request. For $P2$ requests, a private or rented vehicle can be used (constraints (6.2)), while $P1$ requests can only be served by a private vehicle (constraints (6.3)).

To generate as many feasible vehicle routes as possible, the ALNS is modified by the following proposals. First, the CC option is penalized by using higher freight charges. For each mandatory request, a freight charge is determined by multiplying the largest freight charge value over all instances with a constant. Furthermore, it has to be ensured that the modes DB and RB are prohibited for $P1$ requests. Therefore, a second modification is installed. This modification ensures that all insertion positions, which are forbidden for $P1$ requests, are skipped in the insertion phase of the ALNS. Due to this modification, it might be difficult to insert a $P1$ request into one of the existing vehicle routes because many insertion positions are blocked. That is why the insertion phase is split into one insertion phase for mandatory requests and a second one for exchangeable requests. Thus, mandatory requests do not compete with exchangeable requests for limited insertion positions. One main disadvantage of this extension is that many attractive request combinations might not be considered during the local search of the ALNS. To determine the best procedure, a computational study was executed. The results of this study indicate that the procedure with two insertion phases has a better performance in terms of the solution quality as well as the feasibility of instances. That is why in this case the ALNS in the CGB-heuristic uses two insertion phases instead of one insertion phase.

### 6.1.2. Strict composition procedure

By applying the strict generation procedure, many request combinations remain unconsidered by the local search of the ALNS because the route generation (i.e., phase 1) is severely limited. The strict composition procedure proposes to relax the conditions for generating vehicle routes. This solution strategy allows the generation of feasible and "incompatible vehicle routes" when solving the subproblem. Vehicle routes are incompatible when they comply with all constraints of the IOTPP-FL except for the forwarding limitations. As soon as the subproblem related to the IOTPP is solved, the best feasible and incompatible vehicle routes are submitted to the SPP-LP or SCP master problem related to the IOTPP-FL, where a feasible transportation plan is generated out of the submitted vehicle routes.

To generate a feasible transportation plan out of feasible and incompatible best vehicle routes, it is necessary to extend the master problem by forwarding limitations. Different approaches can be used. For example, either by adding new constraints which extend constraints (5.39) to address forwarding limitations, or by splitting this constraint so that only the permitted fulfillment modes are allowed. Based on a computational study, the latter approach was chosen. That is why constraints (5.39) are split into three disjoint constraints depending on the set of pickup nodes $EX$, $P1$, and $P2$. The SPP master problem related to the IOTPP-FL is obtained by replacing constraints (5.39) by the following constraints:

$$\sum_{r \in R} a_{ir} u_r + u_i^{CC} = 1\,, \qquad \forall i \in EX, \tag{6.4}$$

$$\sum_{r \in R} a_{ir} u_r = 1\,, \qquad \forall i \in P2, \tag{6.5}$$

$$\sum_{r \in R^1} a_{ir} u_r = 1\,, \qquad \forall i \in P1. \tag{6.6}$$

The SPP formulation of the IOTPP-FL consists of the mathematical model with constraints (5.38) and (5.40)–(6.6). Constraints (6.4) mean that every fulfillment mode is applicable for exchangeable requests. Constraints (6.5) ensure that a CC cannot be chosen for a $P2$ request. Furthermore, it has to be ensured that a private vehicle is used for a $P1$ request. Therefore, constraints (6.6) ensure that a vehicle of the private fleet is selected for request fulfillment. The SPP-LP formulation of the IOTPP-FL is given by replacing constraints (5.41) with $u_r \geq 0$. An SCP formulation of the IOTPP-FL is obtained by relaxing the request constraints (6.4)–(6.6), thus allowing the selection of more than one vehicle or CC for a request.

The structure of the strict composition procedure is presented in Figure 6.2. As soon as the subproblem related to the IOTPP is solved by the ALNS, the best vehicle routes are submitted to the SPP-LP or SCP master problem related to the IOTPP-FL. Since forwarding limitations are ignored by the ALNS, many of these submitted best vehicle routes may contain mandatory requests which are served by an improper fulfillment mode. The SPP-LP master problem is extended as described in order that feasible vehicle routes are selected. Due to the modified SPP-LP master problem related to the IOTPP-FL, it is necessary to extend the objective function of the ALNS. Thus, the objective function (5.42) is modified by replacing $\sum_{i \in P} \pi_i a_{ik}$ by $\sum_{i \in EX} \pi_i a_{ik} + \sum_{i \in P1} \pi_i a_{ik} + \sum_{i \in P2} \pi_i a_{ik}$.

By solving the SPP-LP master problem related to the IOTPP-FL, the solver CPLEX computes the dual values, which are used to guide the local search of the ALNS, in order that many of the submitted vehicle routes are feasible. The iterative procedure of generating and selecting vehicle routes is repeated until one of the stopping criteria of the CGB-heuristic is met. Afterward, the known mechanism to eliminate multiple assigned requests takes place. Thereby, it is observed that mandatory requests are often inserted into an incompatible vehicle route by the regret-k heuristic. That is why the mechanism to eliminate multiple assigned requests is modified for the strict composition procedure. The reason for this behavior is that the regret-k heuristic selects the best vehicle route for a mandatory request instead of the best feasible vehicle route regarding forwarding limitations. To solve this issue, the regret-k heuristic is applied twice: once for mandatory

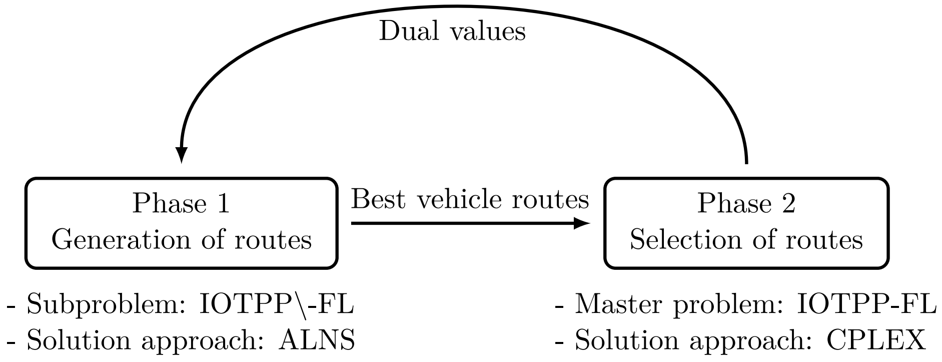


***Figure 6.2.:*** *Structure of the strict composition procedure, cf. Ziebuhr and Kopfer (2016a)*

requests and once for exchangeable requests. Due to this, it is ensured that the best, feasible vehicle route is selected for mandatory requests. In general, this modification is also recommended for the CGB-heuristic with the strict generation procedure, although, in this case, the transportation plans are still feasible without this modification.

In computational studies, it was observed that it is often impossible to find a feasible transportation plan for IOTPP-FL instances with a high ratio of mandatory requests. This issue occurs after solving the subproblem for the first time, within the iterative procedure of the CGB-heuristic, where the set of submitted routes is not guided by the dual values of the SPP-LP master problem. Two extensions are proposed for handling this issue. One extension proposes that a specific ratio (up to 10 %) of the submitted best vehicle routes contains vehicle routes where as many mandatory requests as possible are served by a suitable fulfillment mode. Thereby, it does not matter whether these vehicle routes have a high solution quality because the remaining iterations of the CGB-heuristic are guided by the dual values. Another proposal is that the IOTPP-FL is solved by the strict generation procedure in the first (and only the first) iteration of the iterative procedure. It means that in the first iteration just feasible vehicle routes are submitted to the SPP-LP master problem. A comparison of both extensions indicated that the latter approach performs better in terms of the ratio of feasible instances and the quality of the solutions. In summary, the strict composition procedure solves the subproblem related to the IOTPP-FL in its first iteration and the subproblem related to the IOTPP in subsequent iterations, while the SPP-LP or SCP master problem related to the IOTPP-FL is always solved.

### 6.1.3. Repair procedure

A third solution strategy for handling forwarding limitations is the repair procedure. This solution strategy repairs violations in terms of forwarding limitations after the CGB-heuristic has solved the IOTPP. The repair procedure is motivated by the idea that a high quality IOTPP solution might be easy to repair. Repairing can be done by shifting mandatory requests within the IOTPP solution in order that all forwarding limitations are satisfied. Especially for small ratios of mandatory requests, the procedure is expected to achieve good results. The repair procedure is divided into two phases: feasibility and improvement phase. The feasibility phase is responsible for transforming IOTPP solutions to feasible IOTPP-FL solutions, while the improvement phase rearranges requests of feasible IOTPP-FL solutions with the goal to reduce the transportation costs. The algorithm of the feasibility phase is outlined in Alg. 12.

**Algorithm 12:** Repair procedure: feasibility phase, cf. Ziebuhr and Kopfer (2016a)

**Data**: problem data, best IOTPP solution $s_b$, set of requests served by a CC $G^{\text{CC}}$, set of properly served mandatory requests $H$, set of improperly served mandatory requests $G$

**for** *each mandatory request* $g \in G$ **do**
  initialize solution $s' \leftarrow s_b$
  initialize solution $s_g \leftarrow$ all requests are served by CC
  initialize sets $\Omega_g \leftarrow \{\}$; $\Omega_2^A, \Omega_3^A \leftarrow \{g\}$; $\Omega_2^B, \Omega_3^B \leftarrow G\backslash\{g\}$; $\Omega_2^C, \Omega_3^C \leftarrow G^{\text{CC}}$
  apply successive insertion operator $\text{SIO}(s', s_g, \{g\}, \Omega_g)$
  **for** *each private vehicle route* $k_1 \in K_1$ **do**
    **for** *each rented fleet vehicle route* $k_2 \in K_2 \cup K_3$ **do**
      initialize solution $s'' \leftarrow s'$, request set $\Omega_2' \leftarrow \Omega_2$
      change fulfillment mode of vehicle route $k_1$ with $k_2$ in $s''$
      **if** *vehicle route* $k_1$ *or* $k_2$ *leads to an infeasible solution* **then**
        delete incompatible vehicle route out of $s''$
        update request set $\Omega_2'$
      **end**
      apply successive insertion operator $\text{SIO}(s'', s_g, \Omega_2', \Omega_g)$
    **end**
  **end**
  **for** *each private vehicle route* $k_1 \in K_1$ **do**
    initialize set $\Phi_{K1}$ of randomly chosen PD-pairs in vehicle route $k_1$
    **for** *each PD-pair* $j \in \Phi_{K1}$ **do**
      initialize solution $s'' \leftarrow s'$, request set $\Omega_3' \leftarrow \Omega_3$
      delete PD-pair $j$ out of solution $s''$
      update request set $\Omega_3'$
      apply successive insertion operator $\text{SIO}(s'', s_g, \Omega_3', \Omega_g)$
    **end**
  **end**
  update solution $s_b \leftarrow s_g$
  update set $G \leftarrow G\backslash\Omega_g$
  update set $H \leftarrow H \cup \Omega_g$
**end**
**return** $s_b$ best IOTPP-FL solution

In the feasibility phase, the best IOTPP solution $s_b$, the set of exchangeable requests served by a CC $G^{\text{CC}}$, the set of properly served mandatory requests $H$, and the set of improperly served mandatory requests $G$ (ordered randomly) are given. Three insertion attempts (INS) are applied subsequently for each improperly served mandatory request $g \in G$. Thereby, the goal is the identification of a feasible IOTPP-FL solution by observing that an accepted solution always contains at least the current mandatory request $g$ and the request set $H$. The applied INS use different procedures for the insertion of improperly served mandatory requests into the existing vehicle routes based on the regret-k heuristic from Ropke and Pisinger (2006). To summarize the different INS: insertion attempt 1 uses the regret-k heuristic, insertion attempt 2 uses an inter-route exchange operator and the regret-k heuristic, and insertion attempt 3 uses a removal operator and the regret-$k$ heuristic. Different request sets are used depending on the insertion attempt. Insertion attempt 1 tries to insert the current mandatory request $g$ into the solution $s'$, while the remaining INS try to insert several request sets into the solution $s''$. Insertion attempt 2 considers the request sets $\Omega_2^A$, $\Omega_2^B$, and $\Omega_2^C$, while insertion attempt 3 considers the request sets $\Omega_3^A$, $\Omega_3^B$, and $\Omega_3^C$. Thereby, $\Omega_2^A$ and $\Omega_3^A$ contain the current mandatory request $\{g\}$, $\Omega_2^B$ and $\Omega_3^B$ contain the remaining improperly served mandatory request set $G\backslash\{g\}$, and $\Omega_2^C$ and $\Omega_3^C$ contain the current set of exchangeable requests served by a CC $G^{\text{CC}}$. The idea of partitioning the request sets into three subsets is to provide a priority order among the requests during insertion. For each mandatory request $g \in G$, the current solution $s'$, the current best solution $s_g$, the current best request set $\Omega_g$ as well as the current request sets $\Omega_2$ (i.e., $\Omega_2 = \Omega_2^A \cup \Omega_2^B \cup \Omega_2^C$) and $\Omega_3$ (i.e., $\Omega_3 = \Omega_3^A \cup \Omega_3^B \cup \Omega_3^C$) are first initialized. Then, the INS are applied. All INS use a three step insertion approach. The insertion approach is denoted as successive insertion operator (SIO). The SIO is outlined in Alg. 13.

**Algorithm 13:** Successive insertion operator, cf. Ziebuhr and Kopfer (2016a)

**Data**: problem data, current solution $s'$, current best solution $s_g$, current request set $\Omega$, current best request set $\Omega_g$

**for** *each request set* $\omega \in \Omega$ **do**
    initialize solution $s'' \leftarrow s'$
    try to insert each request $i \in w$ into $s''$ by the regret-$k$ heuristic
    **if** $s''$ *is a feasible IOTPP-FL solution and* $f(s'') \leq f(s_g)$ **then**
        update solution $s_g \leftarrow s''$, request set $\Omega$, request set $\Omega_g$
    **end**
**end**
**return** $s_g$ best solution, $\Omega$ request set, and $\Omega_g$ best request set

The input to the SIO is given by: the current solution $s'$, the current best solution $s_g$, the current request set $\Omega$, and the current best request set $\Omega_g$. The set of requests $\Omega$ depends on the insertion attempt. The SIO is applied separately for each request set $\omega \in \Omega$. Based on the input data, the operator tries to insert each request $i \in w$ into the solution $s''$ by the regret-k heuristic. In case that the modified solution $s''$ is feasible and has lower costs than the current best solution $s_g$, the current best solution $s_g$ is updated by the solution $s''$, the current best set of requests $\Omega_g$ is updated by the inserted mandatory requests out of the set $G$, and the current set of requests $\Omega$ is updated by the inserted requests. The latter updating process means that if a request from the set $\Omega$ can be inserted into a vehicle route by the regret-$k$ heuristic, this request will be removed from the set $\Omega$. The SIO is repeated until each request set $\omega \in \Omega$ is analyzed.

As long as the set of improperly served mandatory requests $G$ is not empty, different attempts for insertion are considered. Insertion attempt 1 (line 5 of Alg. 12) tries to generate a feasible solution by applying the SIO for the existing solution $s'$ by inserting the current mandatory request $g$. It means that the existing solution $s'$ is not modified prior to the application of the SIO. Insertion attempt 2 (lines 6–16 of Alg. 12) tries to modify the existing solution $s'$ by a route exchange operator before the SIO is performed, where vehicle routes with different fulfillment modes are exchanged. This insertion attempt is repeated for every possible combination of vehicle routes. In case that a shifted vehicle route is not feasible for a fulfillment mode, the incompatible route is deleted first and then rebuilt. Insertion attempt 3 (lines 17–25 of Alg. 12) destroys the existing solution by removing one to eight pickup and delivery pairs (referred to as PD-pair) out of a private vehicle route. Each of these PD-pairs is stored in the set $\Phi_{K1}$. As soon as the solution is destroyed, the SIO is applied for the insertion of the updated request set $\Omega_3'$. Insertion attempt 3 is repeated until each PD-pair combination is analyzed. Afterward, the best solution $s_b$, the request set $G$, and the request set $H$ are updated.

As soon as the feasibility phase is completed, the improvement phase begins to improve the IOTPP-FL solution by applying different well-known intra-route exchange (Swap and Or-Operator) and inter-route exchange (Exchange and Relocation Operator) operators. The operators are briefly outlined in Alg. 14, while a detailed explanation is given by Funke et al. (2005). In Alg. 14, two improvement steps are proposed with different combinations of the operators and numbers of exchangeable PD-pairs. At every improvement step, each PD-pair of the solution is analyzed. Thereby, it is always ensured that an accepted solution represents a feasible IOTPP-FL solution. In the first improvement step (lines 2–13 of Alg. 14), one PD-pair with pickup node $i_1$ and delivery node $d_1$ is exchanged with a second PD-pair with pickup node $i_2$ and delivery node $d_2$, assuming that a feasible IOTPP-FL solution can be obtained. For example, it is not possible to exchange a $P1$ request in a private vehicle route with a $EX$ request in a rented vehicle route. In case that both PD-pairs can be exchanged, two scenarios are possible. In Scenario 1, both PD-pairs belong to the same vehicle route. In this case a Swap Operator (where two nodes in the same vehicle route exchange their positions) is used for the pickup nodes $i_1$ and $i_2$ and an Or-Operator (where a selected node is moved from its current position to a different position in the same vehicle route) is used for the delivery nodes $d_1$ and $d_2$. In Scenario 2, both PD-pairs belong to different vehicle routes. In this case, an Exchange Operator (where two nodes in two different vehicle routes exchange their positions) is used for pickup nodes $i_1$ and $i_2$ and a Relocation Operator (where a selected node is transferred from one vehicle route to another) is used for the delivery nodes $d_1$ and $d_2$. It means that intra-route exchange operators are considered in Scenario 1 and inter-route exchange operators are considered in Scenario 2. These combinations of operators are selected because they are able to generate feasible solutions in terms of time, routing, and loading constraints. The best solution $s_b$ is updated every time a modified solution $s''$ is feasible for the IOTPP-FL and has lower costs than the so far best solution. In the second improvement step (lines 14–22 of Alg. 14), different sets of PD-pairs are removed from the solution $s''$ and reinserted afterward by the regret-k heuristic. These PD-pair sets are stored in the set $\Phi_K$, where one PD-pair set contains one to five PD-pairs of the same vehicle route. Thereby, it is ensured that all possible combinations of the one to five PD-pairs are included into the set $\Phi_K$. The best solution $s_b$ is updated every time a modified solution $s''$ is feasible for the IOTPP-FL and has lower costs than the so far best solution. Finally, the best IOTPP-FL solution is returned.

**Algorithm 14:** Repair procedure: improvement phase, cf. Ziebuhr and Kopfer (2016a)

**Data**: problem data, best IOTPP-FL solution $s_b$

1 initialize solution $s' \leftarrow s_b$
2 **for** *each pickup node* $i_1 \in P$ **do**
3     **for** *each pickup node* $i_2 \in P$ **do**
4         initialize solution $s'' \leftarrow s'$
5         identify corresponding delivery nodes $d_1$ and $d_2$ in s”
6         **if** $i_1$ *can be exchanged by* $i_2$ **then**
7             execute either Scenario 1 (same vehicle route) or Scenario 2 (different vehicle routes)
8         **end**
9         **if** $s''$ *is a feasible IOTPP-FL solution and* $f(s'') \leq f(s_b)$ **then**
10             update solution $s_b \leftarrow s''$
11         **end**
12     **end**
13 **end**
14 initialize set $\Phi_K$ of the one to five PD-pairs in all vehicle routes $K$
15 **for** *each PD-pair set* $j \in \Phi_K$ **do**
16     initialize solution $s'' \leftarrow s'$
17     delete PD-pair set $j$ out of solution s”
18     try to reinsert PD-pair set $j$ into $s''$ by the regret-k heuristic
19     **if** $s''$ *is a feasible IOTPP-FL solution and* $f(s'') \leq f(s_b)$ **then**
20         update solution $s_b \leftarrow s''$
21     **end**
22 **end**
23 **return** $s_b$ best IOTPP-FL solution

Inspired by the results of some computational studies, a fourth solution strategy for handling mandatory requests is proposed. The idea is to combine the strict composition procedure with the repair procedure by repairing incompatible vehicle routes before the SPP-LP or SCP master problem is solved. It means to repair the submitted incompatible best vehicle routes, which are generated by solving the subproblem related to the IOTPP, instead of achieving a feasible transportation plan by solving the SPP-LP or SCP master problem related to the IOTPP-FL. Since the dual values introduce a bias towards the generation of feasible columns, it is expected that the submitted best vehicle routes are easier to repair than repairing the final IOTPP solution (which actually has to be done in case of the original repair procedure). Because the execution of the repair procedure is rather time-consuming, two limited repair procedures are applied instead of the known repair procedure. The first limited repair procedure consists only of the known feasibility phase, while the second one contains the known feasibility phase and the first part of the improvement phase (lines 1–13 of Alg. 14). The limited repair procedures are included within the ALNS. To compensate the computational effort for repairing incompatible vehicle routes, the existing time limit of the ALNS is doubled in order to generate and to repair a suitable amount of incompatible vehicle routes, which can be transferred to the SPP-LP or SCP master problem.

## 6.2. Computational studies

In this section, the results of two computational studies are presented. First, different basic solution approaches with the strict generation procedure are compared with each other regarding the quality of their solutions as well as their computational effort for solving the IOTPP-FL. To be precise, the ALNS without the column generation scheme, the CGB-heuristic with a homogenous (CGB-HOM) and heterogeneous vehicle fleet (CGB-HER) are used as basic solution approaches. Second, the best basic solution approach with the strict generation procedure is used for a benchmark study with the existing solution approach of Schönberger (2005) for solving the PDSP-CR. The remaining solution approaches of Li et al. (2016); Chen (2016) could not be used for a benchmark study, because mandatory requests are not selected randomly or a detailed description of the instances is missing.

Test instances are required to perform the mentioned computational studies. In this part, all instances are based on the VRPTW instance classes $R1$ ($r100$–$r112$), $C1$ ($c100$–$c109$), and $RC1$ ($rc100$–$rc108$) with 100 customers from Solomon (1987). Depending on the computational study, certain VRPTW instances are used. In this thesis, the VRPTW instance classes $R2$, $C2$, and $RC2$ are not used, because the best-known solutions of these instances need less than three vehicles for request fulfillment compared to more than ten vehicles for the instance classes $R1$, $C1$, and $RC1$. Based on the number of used vehicles, it is obvious that the instance classes $R2$, $C2$, and $RC2$ cannot be recommended for transportation scenarios with three different vehicle types. It is possible to derive PDPTW instances by using the VRPTW instances from Solomon (1987). In literature, two approaches are available for the extension, Lau and Liang (2001) paired up customers appearing in the routes of the optimal solutions of VRPTW one by one, while Li and Lim (2001) randomly paired up the customer locations within routes. In this thesis, the latter approach is mainly used, while the former approach is only used in Subsection 6.2.2.

In Subsection 6.2.1, the IOTPP-FL is considered. Therefore, the PDPTW instances of Li and Lim (2001) are used, which are extended by Wang et al. (2014) in order to make them suitable for the consideration of different fulfillment modes. Several modifications are executed. First, the size of the total vehicle fleet $|K|$ is set to the number of vehicles used in the best-known solutions for the PDPTW. It means that every instance might be solved without CCs. Second, the total vehicle fleet is subdivided into three parts: 40 % private vehicles ($|K_1|$), 30 % vehicles on mode RB ($|K_2|$), and 30 % vehicles on mode DB ($|K_3|$). Third, the cost structure of these different vehicle types is defined by determining an average route length $l_{\text{ref}}$ (80) of all chosen PDPTW instances and a basic cost rate $c_{\text{ref}}$ (3.8) adjusted by adding a random term (up to 0.38). Based on these reference values, the mentioned cost sets are defined: $\beta_1 = 0.32 \cdot c_{\text{ref}}$, $\beta_2 = c_{\text{ref}}$, $\beta_3 = 0$, $\alpha_1 = 0.48 \cdot l_{\text{ref}} \cdot c_{\text{ref}}$, $\alpha_2 = 0$, and $\alpha_3 = 0.9 \cdot l_{\text{ref}} \cdot c_{\text{ref}}$. Finally, the freight charge $\gamma_i$ for the employment of a CC is defined by the approximation function of Krajewska and Kopfer (2009). By applying this approximation function, the freight charge depends on the load to be transported and the distance between the pickup and delivery node. In the following studies, it is assumed that the applied pairing of the locations, time windows, load variables, and parameters for calculating the payments to subcontractors are well defined. That is why they are fixed and will not be diversified in the computational studies. In the studies, instances with different ratios of mandatory requests are generated. Ratios of 5 %, 10 %, 15 %, and 20 % mandatory requests are considered, which means e.g., that 20% of all requests are mandatory requests and 80 % are exchangeable requests. For every ratio and instance, a certain number of samples is generated depending on the computational study. Instead of considering both types of mandatory requests in one study, the impact of $P1$ and $P2$ requests is investigated separately from each other. That is why it is always restricted to the investigation of scenarios with exchangeable requests and the occurrence of one single type of mandatory requests. Hence, each sample is solved once for $P1$ and once for $P2$ requests. As an evaluation criterion, the percentaged increase of costs between the IOTPP-FL solution $f(EX \cup P1)$ or $f(EX \cup P2)$ computed by the best heuristic and best-known IOTPP solution $f^b(P^*)$ as presented by Wang et al. (2014) is used. $P^*$ represents the IOTPP instance, which ignores forwarding limitations. The percentaged increase of costs $\mathcal{G}_a$, due to solving the, IOTPP-FL with one of the proposed strategies instead of solving the corresponding IOTPP with the CGB-heuristic, can be calculated as follows:

$$\mathcal{G}_a = \frac{f(EX \cup P1 or P2) - f^b(P^*)}{f^b(P^*)} \cdot 100\,\% \qquad (6.7)$$

In Subsection 6.2.2, the PDSP-CR is considered. Three instances per class of the VRPTW instances of Solomon (1987) are solved. To be precise, the instance classes $R1$ ($r103$, $r104$, and $r107$), $C1$ ($c101$, $c104$, and $c105$), and $RC1$ ($rc103$, $rc104$, and $rc107$) are solved. Schönberger (2005) uses the same idea as proposed by Lau and Liang (2001) for generating PDPTW instances. In Schönberger (2005), each of the mentioned instances is extended by the definition of pickup and delivery requests, where the pairings of pickup and delivery nodes are generated randomly based on the best solutions of the VRPTW instances. Three different pairings are proposed for each instance. Furthermore, the instances are extended by the definition of the freight charges. To calculate the freight charge of the request $r$, a constant is multiplied by a coefficient and with the distance between the pickup and delivery node of the request $r$. The coefficient of the request $r$ is determined by taking the route $k$, which serves the request $r$, out of the best VRPTW solution and dividing the driven distance of the route $k$ through the sum of the distances between all pickup and delivery locations within the route $k$. Then, PDSP instances are generated. To be suitable for the PDSP-CR, each of the PDSP instances is extended by the definition of mandatory requests. Six different ratios are considered: 50 %, 60 %, 70 %, 80 %, 90 %, and 100 %. In contrast to the previous ratios, these ratios describe for each single request the probability to be selected as a mandatory request. These ratios do not ensure that $p_r$ % of all requests are mandatory requests and $(1 - p_r)$ % are exchangeable requests. They define that a request is selected as a mandatory request with a probability of $p_r$ %. Another difference is that the PDSP-CR just considers $P1$ requests due to the missing option of using long-term carriers. The described PDSP-CR instances have not been published. That is why these instances are rebuilt according to the description in Schönberger (2005). To achieve a better matching between the instances, five different pairings are generated instead of three as proposed by Schönberger (2005). As an evaluation criterion the relative costs between the PDSP-CR solution $f(EX \cup P1)$ computed by the best heuristic and the best-known VRPTW solution $f^b_{\text{VRPTW}}(P^*)$ are used. The relative costs can be calculated as follows Schönberger (2005):

$$\mathcal{G}_b = \frac{f(EX \cup P1)}{f^b_{VRPTW}(P^*)} \tag{6.8}$$

The applied CGB-heuristic uses the same parameter setting as suggested by Wang et al. (2014). An overview of the applied parameters is outlined in Table 6.1. A detailed explanation of the meaning of the parameters can be found in Section 5.4.

***Table 6.1.:*** *Parameter setting of CGB-heuristic, cf. Ziebuhr and Kopfer (2016a)*

| categories | parameter values |
|---|---|
| iterative procedure | 1. Stopping criterion: 5 iterations |
| | 2. Stopping criterion: 3 iterations ($\leq 5\,\%$ improvement) |
| subproblem | 1. Stopping criterion: 1. Iteration: 10000 rounds (120s) |
| | 2. Stopping criterion: 2.-5. Iteration: 2500 rounds (30s) |
| | 3. Stopping criterion: 1,250 rounds ($0\,\%$ improvement) |
| master problem | Solver: time limit of 120s and MIP-Gap of 0.01 |
| neighborhood | $\max(4,\ 10\,\%$ of $P) \leq$ q $\leq \min(40\,\%$ of $P,\ 100)$ |
| Shaw removal | $\phi = 9$, $\chi = 3$, $\psi = 2$, $\omega =$ not used, $p = 6$ |
| worst removal | $p_{\text{worst}} = 3$ |
| scores | $\sigma_1 = 33$, $\sigma_2 = 9$, $\sigma_3 = 13$, $r_z = 0.1$ |
| temperature | $w = 0.01$, $c$ is set to $5\,\%$ of $T_{\text{start}}$, $\zeta = 0.5$ |
| noise | $\eta = 0.025$ |
| no. of recorded solutions | 1,000 best solutions per iteration |

The applied CGB-heuristic with the different procedures is implemented in C++ (Visual Studio 2012) and the computational studies are executed on a Windows 7 PC with Intel Core i7-2600 processor (3.4 GHz and 16 GB of main memory). To determine the dual values of the SPP-LP master problem, the commercial solver CPLEX (version 12.5.1) is applied. In terms of the setting of CPLEX, the standard setting is always used except for the applied time limit and MIP-Gap.

### 6.2.1. Identification of the basic solution approach

In this subsection, the best basic solution approach for the IOTPP-FL has to be determined by a computational study. Three solution approaches are evaluated: the ALNS without the column generation scheme, the CGB-HOM, and the CGB-HER. In terms of the application of the ALNS without the column generation scheme, the heuristic stops after 25,000 iterations or 25,000/3 iterations without any improvement of the best solution, while the other solution approaches use the parameter setting of Table 6.1. To determine the preferable heuristic, each of these solution approaches is extended by the mentioned strict generation procedure of Subsection 6.1.1 for handling mandatory requests.

The following computational studies are based on the IOTPP instances of Wang et al. (2014). Thereby, just two instances per instance class $R1$, $C1$, and $R1$ are used. The instances $c101$, $c102$, $r101$, $r102$, $rc101$, and $rc102$ are used. These IOTPP instances are extended as described by the definition of mandatory requests. For each instance, one sample per ratio of mandatory requests is considered and solved. Each sample is solved five times and the best solution is recorded with their percentaged increase of costs $\mathcal{G}_a$. These percentaged increase of costs are used to generate the mean percentaged increase of costs per instance. In Table 6.2, the mean percentaged increase of costs per instance and the mean computational effort per solution approach are presented. For example, the mean percentage gap of 5.16 % for instance $c101$ computed by the ALNS means that samples with 5 %–20 % mandatory requests lead to about 5.16 % additional costs than the one without mandatory requests.

***Table 6.2.:*** *Mean $\mathcal{G}_a$ for ALNS, CGB-HOM, and CGB-HER, cf. Ziebuhr and Kopfer (2014)*

| requests | approach | mean $\mathcal{G}_a$ | | | | | | time |
|---|---|---|---|---|---|---|---|---|
| | | lc101 | lc102 | lr101 | lr102 | lrc101 | lrc102 | (s) |
| $P2$ | ALNS | 5.16 | 2.06 | 4.28 | 6.33 | 5.87 | 5.26 | 305.34 |
| | CGB-HOM | 2.21 | 0.64 | 2.34 | 2.58 | 3.25 | 3.59 | 290.05 |
| | CGB-HER | 2.21 | 0.64 | 2.34 | 2.58 | 3.25 | 3.59 | 261.68 |
| $P1$ | ALNS | 14.42 | 12.22 | 5.75 | 7.61 | 10.11 | 9.30 | 321.26 |
| | CGB-HOM | 14.11 | 11.34 | 5.13 | 6.05 | 9.82 | 8.67 | 263.18 |
| | CGB-HER | 14.21 | 11.34 | 5.13 | 6.05 | 9.82 | 8.67 | 223.07 |

Table 6.2 indicates that the ALNS always leads to solutions which have higher transportation costs than the remaining basic solution approaches. For example, the ALNS computes a mean percentaged increase of costs of 4.82 % for all instances with $P2$ requests, which is obviously higher than 2.43 % for both CGB-heuristics. Furthermore, the computational effort is slightly higher as compared to the other solution approaches. That is why the ALNS cannot be recommended as a basic solution approach. By comparing the CGB-HER with the CGB-HOM, it seems obvious that both heuristics are comparable with respect to their solution quality. One exception is the instance $c101$, where the CGB-HER performs slightly worse than the CGB-HOM. The remaining figures are equal. However, the CGB-HER has a computational effort which is about 18 % less per instance compared to the CGB-HOM. Corresponding to the solution quality and computational effort, the CGB-HER is used as a basic solution approach of this part, where many samples have to be solved.

### 6.2.2. Comparison with a state of the art heuristic

The second computational study compares the performance of the CGB-HER with the strict generation procedure (referred to as CGB-HER for this subsection) to the alternating and converging constraint memetic algorithm (ACC-MA) of Schönberger (2005) for the PDSP-CR. As mentioned in Section 5.3, the idea of the ACC-MA is to use a sophisticated selection procedure, in which the transportation costs are already considered before the first feasible solution is found. The ACC-MA represents the best solution approach of Schönberger (2005) which is why this heuristic is used for the benchmark study. The remaining approaches of Li et al. (2016); Chen (2016) could not be used for this study.

The benchmark study uses the rebuilt PDSP-CR instances of Schönberger (2005). Thereby, it is worth mentioning that the instance class $C1$ is skipped because the ACC-MA could not solve the corresponding instances. In Schönberger (2005), instances of class $R1$ and $RC1$ are solved three times and the mean relative costs are reported. Here, the CGB-HER with the strict generation procedure solves the rebuilt instances just once instead of three times. In Table 6.3, the relative costs $\mathcal{G}_b$ for each ratio and instance class are presented. As mentioned, this study just considers $P1$ requests.

***Table 6.3.:*** *Comparison of CGB-HER with ACC-MA in case of relative costs $\mathcal{G}_b$, cf. Ziebuhr and Kopfer (2014)*

| mean $\mathcal{G}_b$ | CGB-HER | | | | | | ACC-MA | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 50 % | 60 % | 70 % | 80 % | 90 % | 100 % | 50 % | 60 % | 70 % | 80 % | 90 % | 100 % |
| R1 | 0.971 | 0.981 | 0.986 | 0.998 | 1.005 | - | 1.04 | 1.05 | 1.06 | 1.09 | 1.09 | 1.10 |
| RC1 | 0.979 | 0.983 | 0.995 | 1.009 | - | - | 1.02 | 1.00 | 1.02 | 1.05 | 1.05 | 1.06 |

As can be seen from Table 6.3, under the given test conditions, the CGB-HER seems to be superior compared to the ACC-MA for all ratios and instance classes in terms of the solution quality. It is even observed that the relative costs of the CGB-HER are sometimes even lower than one, which means that the identified solution values of the CGB-HER are lower than the reference value. This observation is unexpected because a reference value usually represents the lower or upper bound in a test case. Schönberger (2005) uses the so far best-known VRPTW solution as a reference value. By considering the results of this table, it is assumed that these solution values do not represent the best solution values anymore. This observation might be explained by the option of employing CCs. Due to this additional fulfillment mode, it might be possible to identify transportation plans with lower transportation costs. To verify this assumption, the PDSP instances with the CC option and without mandatory requests are solved by the CGB heuristic. The generated solutions are used as reference values. Then, the CGB-HER is applied a second time for the PDSP-CR instances. Thereby, relative costs between 1.06 to 1.11 are identified, which means additional costs and verifies the previous assumption. In terms of the computational effort, the CGB-HER needs about 156 seconds, while the ACC-MA needs about 270 seconds per PDSP-CR instance. Thereby, it is worth mentioning that both solution approaches run on different machines. One disadvantage of the CGB-HER compared to the ACC-MA is that the CGB-HER is not able to identify feasible solutions for the highest ratios of mandatory requests. Nevertheless, the CGB-HER seems to be superior for solving the PDSP-CR.

In terms of the conduction of findings regarding the impact of mandatory requests, it is refereed to the following chapters. However, it can be summarized that there is an increase of transportation costs by considering mandatory requests and that the impact increases with the ratio of mandatory requests. Furthermore, high ratios of mandatory requests lead to instances which cannot be solved.

In additional computational studies, it is also investigated if there is room for improvements of the CGB-heuristic. Thereby, combinations of different strategies were tested. For example, it was tested to integrate the known PWR procedure or to use the CGB-heuristic with a homogenous vehicle fleet with the remaining solution strategies. However, none of these modifications led to better solutions or less computational effort. That is why the solution approaches are used as described.

Here, an IOTPP with different types of mandatory requests is analyzed. To solve it, a CGB-heuristic is used and extended by a strict generation procedure. The idea of this extension is to overcome invalid solutions in terms of the ALNS by using a procedure which accepts only valid solutions by the SA and uses a preinsertion phase for mandatory requests. Furthermore, the solution strategies strict composition and repair procedure are introduced for handling mandatory request which are addressed in the following chapter. Two computational studies are presented. First, different basic solution approaches are compared with each other regarding their solution quality and their computational effort. It is identified that the CGB-HER outperforms the other solution approaches. Second, the CGB-HER with the strict generation procedure is used in a benchmark study with the solution approach of Schönberger (2005). It is demonstrated that the CGB-HER with the strict generation procedure outperforms the existing algorithm of Schönberger (2005).

# 7. Integrated Operational Transportation Planning

This chapter deals with a TPP with pickup and delivery requests, where a freight forwarder can choose between self-fulfillment and subcontracting for fulfilling a request. This problem is known as IOTPP. IOTP is getting more complex when the choice of the fulfillment mode is limited for some requests. This problem is known as IOTPP-FL. In the previous chapter, the CGB-HER has been identified as preferable solution approach in case of a TPP with exchangeable and mandatory requests. Three different solution approaches were presented for handling mandatory requests. However, just the strict generation procedure has been applied in a computational study. That is why a comparison between the different solution strategies is still missing. In this chapter, all solution strategies are compared with each other in a computational study by solving an IOTPP-FL. These studies indicate that one of the extended versions of the heuristic outperforms all related solution approaches in literature. Furthermore, the goal of this thesis is to identity the increase of transportation costs. That is why detailed computational studies are presented.

This chapter is divided into two sections. First, the considered IOTPP-FL and the applied solution strategies are summarized in Section 7.1. In both cases, descriptions can be found in the previous chapters. Furthermore, an IOTPP-FL is described based on an example. Second, the computational studies are presented in Section 7.2 by two subsections. In Subsection 7.2.1, the best solution strategy has to be identified by comparing the mentioned solution strategies regarding their solution quality, number of feasible solutions, and computational effort. Then, the best solution strategy is used in Subsection 7.2.2, where the impact of mandatory requests is analyzed based on different location structures and private fleet sizes. This chapter is based on Ziebuhr and Kopfer (2016a).

## 7.1. Mathematical formulation and solution approach

This chapter considers an IOTPP-FL with two different types of mandatory requests. The considered IOTPP is based on the PDPTW, which is extended by the option of using rented vehicles on mode DB or RB and employing CCs for fulfilling requests. This IOTPP is extended by transportation services, which implicate forwarding limitations. The following transportation services with their corresponding fulfillment modes are proposed:

- $P1$ service: self-fulfillment,
- $P2$ service: self-fulfillment, long-term carrier,
- $EX$ service: self-fulfillment, subcontracting (i.e., CC and long-term carrier).

The corresponding requests are denoted as $P1$ request, $P2$ request, and $EX$ request. A $EX$ request can be fulfilled by any fulfillment mode, a $P2$ request can be fulfilled by the private or rented vehicle fleet, and a $P1$ request can only be fulfilled by the private fleet. $P1$ and $P2$ requests represent mandatory requests, which have in common that they cannot be fulfilled by a CC.

In Figure 7.1, an IOTPP-FL is illustrated. Here, a freight forwarder is in charge of a request portfolio with eleven requests. To simplify the illustration, a VRPTW is used as a basic vehicle routing problem instead of a PDPTW. The request portfolio of the freight forwarder contains: five $EX$ requests, four $P2$ requests, and two $P1$ requests. The freight forwarder is in charge of one rented vehicle on mode RB, one rented vehicle on mode DB, and two private vehicles for fulfilling these requests. Each of these vehicle types has a limited capacity and different cost rates. Furthermore, the freight forwarder is able to employ a CC for fulfilling a request. The goal is to identify this transportation plan which minimizes the total transportation costs by ensuring that the mandatory requests are fulfilled by their corresponding fulfillment mode. In Figure 7.1, the TPP on the left side as well as a solution for the TPP on the right side are presented.

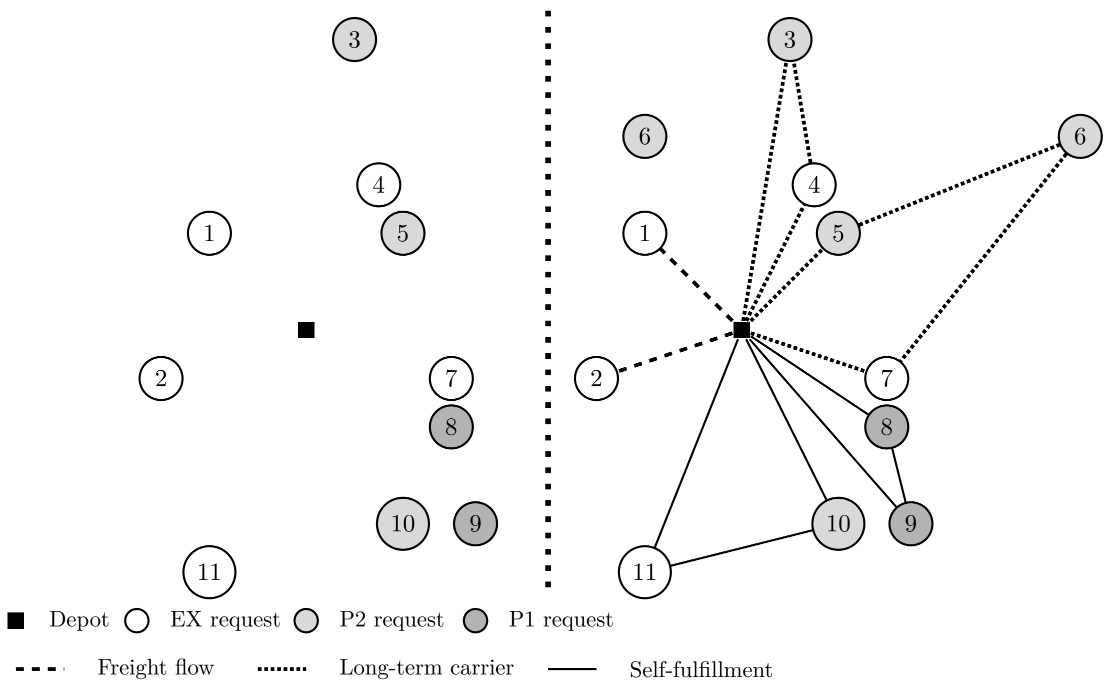


***Figure 7.1.:*** *IOTP problem with exchangeable and mandatory requests*

As can be seen, $EX$ requests can be fulfilled by any fulfillment mode. For example, request one and two are fulfilled by a CC, while request four and seven are fulfilled by a rented vehicle and request eleven is fulfilled by the private fleet. For these requests the fulfillment mode is selected which leads to lower transportation costs. The remaining mandatory requests three, five, six, eight, nine, and ten are fulfilled either by a rented or private vehicle by observing first the requested service and second the transportation costs.

The CGB-HER is applied to solve the IOTPP-FL, which is either extended by the strict generation, strict composition, or repair procedure for handling mandatory requests. Each of these solution strategies is described more in detail in Section 6.1. In the following, it is described which TPP is solved by the subproblem or master problem of the CGB-HER in terms of the different solution strategies.

**Strict generation procedure:** The strict generation procedure solves the IOTPP-FL as subproblem by forwarding just feasible vehicle routes to the master problem. The master problem can be solved as IOTPP.

- Subproblem: The objective function (5.16), the IOTPP constraints (5.18)–(5.34), and the forwarding limitation constraints (6.1)–(6.3) are considered.
- SPP master problem: The objective function (5.38) and the constraints (5.39)–(5.41) are considered.

**Strict composition procedure:** The strict composition procedure mainly solves the IOTPP as subproblem and the IOTPP-FL as master problem. Due to this procedure, feasible solutions for the IOTPP-FL are generated by using the dual values of the master problem. As known, the strict generation procedure is applied in the first round of the iterative procedure of the CGB-HER.

- Subproblem: The objective function (5.16) and the IOTPP constraints (5.17)–(5.34) are considered.
- SPP master problem: The objective function (5.38), the constraints (5.40)–(5.41) and the forwarding limitation constraints (6.4)–(6.6) are considered.

**Repair procedure:** The repair procedure solves the IOTPP as subproblem and master problem and generates a feasible solution as soon as the original CGB-HER is terminated by rearranging mandatory requests.

- Subproblem: The objective function (5.16) and the IOTPP constraints (5.17)–(5.34) are considered.
- SPP master problem: The objective function (5.38) and the constraints (5.39)–(5.41) are considered.

## 7.2. Computational studies

This section presents two computational studies. In Subsection 7.2.1, it is determined which solution strategy combined with the CGB-HER performance superior; i.e., the performance of the strict generation procedure, the strict composition procedure, and the repair procedure are compared. In Subsection 7.2.2, the best solution strategy is used to quantify the increase of costs by considering mandatory requests in diverse transportation scenarios. These scenarios differ regarding the fleet size, the ratio and type of mandatory requests, and the location structure. Based on the results of the second study, general conclusions are derived. These findings can be used as recommendations and guidelines for scheduling and pricing strategies in TPPs with forwarding limitations.

As test instances the already described IOTPP-FL instances of Section 6.2 are used. In contrast to the previous chapter, the computational studies contain all 29 IOTPP instances of the instance classes $R1$, $C1$, and $RC1$. Several samples with different distributions of mandatory requests are generated for each instance. As in the previous chapter, each sample is solved once for $P2$ and once for $P1$ requests. Thereby, ratios of 5 %, 10 %, 15 %, and 20 % mandatory requests are considered. In Subsection 7.2.2, the IOTPP-FL instances are modified in terms of the total fleet size. In this computational study, the private fleet size is either reduced to 25 % or increased up to 55 % or 70 % of the total fleet size. The remaining ratio of the total fleet size is split equally between the rented vehicles on mode RB and DB. An equal distribution is sometimes not available, then rented vehicles on mode RB are preferred as fulfillment mode.

As an evaluation criterion, the percentaged increase of costs between the IOTPP-FL solution computed by the CGB-HER with the best solution strategy and the best-known IOTPP solution as presented by Wang et al. (2014) is used. It means that the same evaluation criterion as in the previous chapter is used, which is defined by the variable $\mathcal{G}_a$. The best-known IOTPP solution values for instances with modified private fleet sizes are not available from Wang et al. (2014). That is why the solution values of these instances are determined by executing the CGB-heuristic for the IOTPP.

The CGB-heuristic is implemented in C++ (Visual Studio 2012) and the computational studies are executed on a Windows 7 PC with Intel Core i7-2600 processor (3.4 GHz and 16 GB of main memory). The dual values are computed by using CPLEX (version 12.5.1). The same parameter setting is used as in Chapter 6.

### 7.2.1. Identification of the best solution strategy

Here, the three proposed solution strategies (i.e., strict generation procedure, strict composition procedure, and repair procedure) are compared in terms of their feasibility quota and quality of the generated solutions. For the 29 IOTPP instances, ten samples are generated per the mentioned ratios of mandatory requests, which are solved once for $P2$ requests and once for $P1$ requests. In total, 80 samples are analyzed per IOTPP instance. It means that 2,320 samples are solved per solution strategy.

First, the feasibility quota is analyzed. A feasibility quota of 60 % means that 60 % of the samples are solved successfully by the heuristic, while it is not possible to find feasible solutions for the remaining 40 % of the samples. Obviously, all samples can be solved successfully in case that forwarding limitations are ignored. Consequently, the feasibility quota measures a procedure's capability to handle scenarios with mandatory requests. Table 7.1 gives an overview about the results.

Table 7.1 indicates that every solution strategy is able to solve the IOTPP-FL with $P2$ requests. However, by considering a ratio above 10 % $P1$ requests, some of the samples cannot be solved. In general, the feasibility quota decreases from the clustered to random to randomly clustered instances. Corresponding to the feasibility quota, the strict generation and strict composition procedure are similar. Furthermore, both procedures outperform the repair procedure. For example, the feasibility quota for solving the instance class $RC1$ with 20 % $P1$ requests is about 25 % lower when using the repair procedure instead of applying one of the other procedures. The result is reasonable because the repair procedure

**Table 7.1.:** *Strategies: feasibility quotas (in %), cf. Ziebuhr and Kopfer (2016a)*

| class | P2 requests | | | | mean | P1 requests | | | | mean |
|---|---|---|---|---|---|---|---|---|---|---|
| | 5 % | 10 % | 15 % | 20 % | | 5 % | 10 % | 15 % | 20 % | |
| | | | | | strict generation procedure | | | | | |
| C1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 99 | 92 | 98 |
| R1 | 100 | 100 | 100 | 100 | 100 | 100 | 99 | 97 | 71 | 92 |
| RC1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 95 | 63 | 89 |
| | | | | | strict composition procedure | | | | | |
| C1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 99 | 92 | 98 |
| R1 | 100 | 100 | 100 | 100 | 100 | 100 | 99 | 97 | 71 | 92 |
| RC1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 96 | 60 | 89 |
| | | | | | repair procedure | | | | | |
| C1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 99 | 83 | 96 |
| R1 | 100 | 100 | 100 | 100 | 100 | 100 | 98 | 91 | 43 | 83 |
| RC1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 90 | 35 | 81 |

is an end-of-pipe approach, where forwarding limitations are included at the end of the algorithm. Consequently, for achieving a high ratio of feasible solutions, it is favorable to include conditions for forwarding limitations early, either by considering them within the subproblem or the master problem of the extended CGB-heuristic. With respect to the feasibility quota, it is worth mentioning that this solution approach is a heuristic which is why it is unknown if a sample is truly infeasible or not.

Next, the percentaged increase of costs $\mathcal{G}_a$ per ratio of mandatory requests is analyzed. Table 7.2 presents the percentaged increase of costs $\mathcal{G}_a$ for every procedure and instance set depending on the ratio and type of mandatory requests.

**Table 7.2.:** *Strategies: mean percentaged increase of costs per ratio of mandatory requests, cf. Ziebuhr and Kopfer (2016a)*

| class | P2 requests | | | | mean | P1 requests | | | | mean |
|---|---|---|---|---|---|---|---|---|---|---|
| | 5 % | 10 % | 15 % | 20 % | | 5 % | 10 % | 15 % | 20 % | |
| | | | | | strict generation procedure | | | | | |
| C1 | 0.83 | 1.25 | 1.85 | 2.13 | 1.51 | 6.83 | 10.00 | 15.90 | 22.03 | 13.69 |
| R1 | 0.68 | 1.21 | 1.76 | 2.38 | 1.51 | 2.76 | 5.28 | 10.14 | 15.09 | 8.32 |
| RC1 | 1.13 | 1.87 | 2.41 | 4.13 | 2.38 | 2.21 | 4.20 | 8.66 | 15.18 | 7.56 |
| | | | | | strict composition procedure | | | | | |
| C1 | **0.61** | **1.04** | **1.60** | **1.93** | **1.29** | **4.98** | **8.25** | **14.84** | **21.34** | **12.35** |
| R1 | **0.67** | **1.08** | **1.51** | **2.18** | **1.36** | **2.68** | **5.09** | **9.86** | **14.43** | **8.01** |
| RC1 | **1.12** | **1.77** | **2.29** | **4.01** | **2.30** | **2.20** | **4.14** | **8.58** | **14.71** | **7.41** |
| | | | | | repair procedure | | | | | |
| C1 | 0.83 | 1.58 | 2.19 | 2.95 | 1.89 | 7.68 | 11.21 | 19.63 | 25.06 | 15.89 |
| R1 | 1.15 | 1.76 | 2.73 | 3.64 | 2.32 | 5.18 | 7.96 | 13.59 | 18.05 | 11.20 |
| RC1 | 1.85 | 3.08 | 3.96 | 6.62 | 3.88 | 3.99 | 6.85 | 11.23 | 16.50 | 9.64 |

In Table 7.2, the figures with the lowest additional costs are marked in bold. Corresponding to this benchmark study, the strict composition procedure turns out to be the best solution strategy and the table indicates that the best IOTPP solutions are completely different from the IOTPP-FL solutions. Thus, the repair procedure is outperformed significantly by the other two procedures. It seems that it is difficult to identify high quality IOTPP-FL solutions by repairing IOTPP solutions. The strict composition procedure outperforms the strict generation procedure in all instances in terms of solution quality. By considering $P1$ requests, the achieved cost reduction is 1.34 % for clustered instances, 0.31 % for random instances, and 0.15 % for randomly clustered instances compared to the results of the strict generation procedure. The superiority of the strict composition procedure can be explained by its ability to generate vehicle routes which are out of scope by applying the strict generation procedure, where mandatory requests are handled by the dual values of the SPP-LP master problem. Consequently, in order to achieve high quality IOTPP-FL solutions, it is not favorable to install the conditions for forwarding limitations at the earliest possible stage (subproblem), but to consider them at a later stage when transportation plans are being constructed (master problem). In terms of the computational effort, on average the strict generation and strict composition procedure need about 200 seconds per sample, while the repair procedure requires about 600 seconds per sample.

The fourth strategy of combining the strict composition procedure with the repair procedure as described in Subsection 6.1.3 is also evaluated. The result of testing the fourth strategy is that the quality of the solutions can be improved compared to the original repair procedure. However, the quality of the solutions is slightly worse in comparison to the application of the strict composition procedure. This can be explained by the time consuming reparation of incompatible vehicle routes which does not compensate the reduced time for investigating the solution space by the ALNS.

Some insights on the effects of forwarding limitations on the IOTPP can be derived from Table 7.2. One finding is that $P2$ requests have a much lower effect of cost escalation than $P1$ requests. The results in Table 7.2 with the strict composition procedure demonstrate that for $P2$ requests the costs increase by about 1.7 % for all problem instances while the increase for $P1$ requests amounts to 9.3 % additional costs. Focusing on the results achieved by the strict composition procedure on clustered instances with 5 %–20 % $P2$ requests, on average a freight forwarder has about 1.29 % supplementary costs, while 5 %–20 % $P1$ requests result in about 12.35 % additional costs. Table 7.2 additionally enables findings on the effect of growing ratios of mandatory requests. It is observed that growing ratios lead to higher additional costs. For example, by applying the strict composition procedure a ratio of 5 %, 10 %, 15 % or 20 % $P2$ requests amounts to 0.80 %, 1.30 %, 1.80 % or 2.70 % additional costs, while it amounts to 3.28 %, 5.83 %, 11.09 % or 16.83 % for $P1$ requests. It means that the increase of costs is less than linearly growing within the range of 5 %–20 % $P2$ requests while it is more than linearly growing for $P1$ requests.

### 7.2.2. Impact of mandatory requests

The second computational study analyzes the impact of mandatory requests based on different location structures (clustered, random, and randomly clustered), private fleet sizes (25 %, 40 %, 55 %, and 70 % of the total vehicle fleet), and ratios and types of mandatory requests. For the 29 instances in the IOTPP test set, 30 samples are generated per ratio of mandatory requests. The samples are solved once for $P1$ requests and once for $P2$ requests. In total, 240 samples are solved per IOTPP instance. The results of this subsection are based on the solution of about 28,000 samples. The CGB-HER with the strict composition procedure is applied to solve the samples.

In scenarios without forwarding limitations, cost advantages might be achieved by reducing the size of the private fleet, i.e., by increasing the amount of subcontracting (Krajewska and Kopfer, 2009). If $P1$ requests exist, the reduction of the private fleet will be problematic because it is not only a matter of costs but also of feasibility. In Table 7.3, the feasibility quotas are presented for the IOTPP-FL with $P1$ requests and private fleet sizes of 25 % and 40 % of the total fleet size. The results for the scenarios with a private fleet above 40 % are not presented, because for those scenarios all samples can be solved.

***Table 7.3.:*** *Fleet size study: feasibility quotas (in %), cf. Ziebuhr and Kopfer (2016a)*

| class | 25 % private fleet | | | | mean | 40 % private fleet | | | | mean |
|---|---|---|---|---|---|---|---|---|---|---|
| | 5 % | 10 % | 15 % | 20 % | | 5 % | 10 % | 15 % | 20 % | |
| C1 | 100 | 99 | 80 | 41 | 80 | 100 | 100 | 96 | 89 | 96 |
| R1 | 100 | 93 | 44 | 13 | 62 | 100 | 100 | 96 | 71 | 92 |
| RC1 | 100 | 94 | 48 | 11 | 63 | 100 | 100 | 93 | 63 | 89 |

In general, it is observed that the feasibility decreases from the clustered to the random and randomly clustered instances. By taking both private fleet size scenarios into account, it seems that random and randomly clustered location structures lead to similar feasibility quotas. Furthermore, Table 7.3 indicates that in some cases a reduced private vehicle fleet leads to problem instances which cannot be solved in agreement with forwarding limitations. For example, it is observed that on average less than 22 % of the samples can be solved in case of a ratio of 20 % $P1$ requests and a total vehicle fleet with 25 % private vehicles. Especially the feasibility quota of the randomly clustered instances is quite low with 11 % feasible and 89 % infeasible solutions. That is why the instances with 20 % $P1$ requests and a 25 % private fleet are skipped for an evaluation of the percentaged increase of costs.

Table 7.4 indicates the influence of forwarding limitations on the increase of costs by considering different private fleet sizes and location structures. The listed percentaged increase of costs are defined per mandatory request instead of ratios of mandatory requests as before. The percentaged increase of costs per mandatory request $\mathcal{G}^*$ is defined by formula (7.1). To determine the value of $\mathcal{G}^*$ for an IOTPP-FL instance, the percentaged increase of costs per ratio of mandatory requests $\mathcal{G}_a$ is divided by the number of mandatory requests $No_{\text{CR}}$.

$$\mathcal{G}^* = \frac{\mathcal{G}_a}{No_{\text{CR}}} \tag{7.1}$$

The idea of this computational study, including the metric change, is to provide transportation planners with a calculated pattern. Thereby, it is preferable to use percentaged increase of costs per mandatory requests because transportation planners can reduce the computational effort for cost estimations by just multiplying the identified additional costs per mandatory request by the number of mandatory requests. In Table 7.4, the figures with lowest additional costs are marked in bold for every ratio and type of mandatory requests. In total, twelve instance classes are considered with three different location structures ($C1$, $R1$, and $RC1$) and four different private fleet sizes (25 %–70 %). The instance class C1-25 % means that the instances with location structure $C1$ are considered where the vehicle fleet is divided into 25 % private vehicles, 37.25 % rented vehicles on mode RB, and 37.25 % rented vehicles on mode DB.

***Table 7.4.:*** *Fleet size study: mean percentaged increase of costs per mandatory request, cf. Ziebuhr and Kopfer (2016a)*

| class - | $P2$ requests | | | | mean | $P1$ requests | | | | mean |
|---|---|---|---|---|---|---|---|---|---|---|
| fleet size | 5 % | 10 % | 15 % | 20 % | | 5 % | 10 % | 15 % | 20 % | |
| C1-25% | **0.19** | 0.25 | **0.22** | **0.20** | **0.22** | 1.71 | 2.05 | 2.51 | - | 2.09 |
| C1-40% | **0.20** | **0.20** | **0.20** | **0.19** | **0.20** | 1.60 | 1.69 | 1.84 | 1.93 | 1.76 |
| C1-55% | **0.04** | **0.06** | **0.04** | **0.04** | **0.05** | 0.48 | 0.66 | 0.74 | 0.88 | 0.69 |
| C1-70% | **0.02** | **0.03** | **0.03** | **0.03** | **0.03** | **0.22** | 0.31 | 0.42 | 0.52 | 0.37 |
| R1-25% | 0.22 | **0.22** | **0.22** | 0.21 | **0.22** | 1.15 | 1.34 | **1.74** | - | 1.41 |
| R1-40% | 0.21 | 0.22 | 0.20 | 0.21 | 0.21 | 0.86 | 0.98 | **1.23** | **1.41** | **1.12** |
| R1-55% | 0.16 | 0.16 | 0.14 | 0.15 | 0.15 | **0.44** | 0.45 | **0.49** | **0.56** | **0.48** |
| R1-70% | 0.09 | 0.09 | 0.09 | 0.10 | 0.09 | 0.25 | 0.34 | **0.33** | **0.37** | 0.32 |
| RC1-25% | 0.49 | 0.36 | 0.50 | 0.49 | 0.46 | **1.00** | **1.12** | 1.83 | - | **1.32** |
| RC1-40% | 0.37 | 0.30 | 0.36 | 0.38 | 0.35 | **0.72** | **0.66** | 1.12 | 1.19 | 0.92 |
| RC1-55% | 0.29 | 0.22 | 0.29 | 0.28 | 0.27 | 0.45 | **0.36** | 0.52 | 0.63 | 0.49 |
| RC1-70% | 0.22 | 0.18 | 0.22 | 0.24 | 0.22 | 0.31 | **0.25** | **0.33** | **0.37** | **0.31** |

Corresponding to Table 7.4, it is observed for $P2$ requests that randomly clustered instances always (i.e., independently of the fleet size and ratio of mandatory requests) lead to the highest percentaged increase of costs, while clustered instances have the lowest percentaged increase of costs. For example, on average a freight forwarder has about 0.12 % supplementary costs for clustered location structures compared to 0.33 % supplementary costs for randomly clustered location structures. The perspective changes in case of $P1$ requests, where the highest supplementary costs appear in clustered location structures, while the lowest additional costs appear in randomly clustered instances. In clustered location structures, requests are located in different clusters and between these clusters there are no requests. There it is often expensive to fulfill mandatory requests for a vehicle fleet with limited availability because mandatory requests might be located in different clusters. However, clustered location structures are promising in case of an unlimited fleet size. In such scenarios, the IOTPP solutions are similar to the IOTPP-FL solutions because the mentioned request clusters represent an advantage. Due to this observation, it is assumed that for clustered instances a kind of turnover point exists in terms of the fleet size. As soon as this point is reached, it seems to be inexpensive to include additional mandatory requests into a transportation plan. It is worth mentioning in terms of the turnover point that $P2$ requests and $P1$ requests differ with respect to the vehicles which are available for vehicle routing. The available fleet for fulfilling mandatory requests consists of private and rented vehicles while it consists of only private vehicles for $P1$ requests. By taking the results in Table 7.4 and the latter remarks into account, it is obvious that the available fleet size is always 100 % for $P2$ requests while the size differs between 25 %–70 % for $P1$ requests. That is why it is expected that the turnover point seems to be between an available fleet size of 70 %–100 %. Table 7.4 also indicates that increasing the ratio of $P1$ requests results in a moderate increase of fulfillment costs for all kinds of location structures and private fleet sizes while there are mixed results for $P2$ requests.

The results in Table 7.4 indicate the impact of forwarding limitations on the percentaged increase of costs per mandatory request. The test sets will now be analyzed through a new evaluation criterion $\mathcal{G}_t$ that integrates both the impact of forwarding limitations and modified fleet sizes.

$$\mathcal{G}_t = \mathcal{G}_a + \left( \frac{f^b(P^*_{\text{new}}) \cdot 100}{f^b(P^*_{\text{old}})} - 100 \right) \% \tag{7.2}$$

The total percentaged increase of costs adds up to the percentaged increase of costs per ratio of mandatory requests $\mathcal{G}_a$ and the percentaged increase of costs caused by a modified private fleet size. In formula (7.2), the term $f^b(P^*_{\text{old}})$ represents the IOTPP solution with the existing private fleet size and $f^b(P^*_{new})$ represents the IOTPP solution with the modified private fleet size. By taking the evaluation criterion $\mathcal{G}_t$ into account, freight forwarders can decide if it is promising to act on a market with mandatory requests by modifying their private fleet size. The first term of formula (7.2) can be obtained from Table 7.4 while the second term is missing. That is why Table 7.5 is presented. Table 7.5 takes the solution values of the IOTPP with a 70 % private fleet as reference value (i.e., $f^b(P^*_{\text{old}})$) and shows the relative cost deviations in case of a reduced private fleet.

***Table 7.5.:*** *Increase of transportation costs compared to a ratio of 70 % private fleet, cf. Ziebuhr and Kopfer (2016a)*

| class | size of the private fleet | | |
|---|---|---|---|
| | 55 % | 40 % | 25 % |
| C1 | +1.14 % | +2.77 % | +5.20 % |
| R1 | +2.70 % | +8.67 % | +12.43 % |
| RC1 | +4.46 % | +10.84 % | +16.45 % |

It is worth mentioning that a high ratio of private vehicles can be a disadvantage in situations with high demand fluctuations and in case of low freight tariffs for subcontracting. Flexibility with respect to the private fleet size and the ability to adapt the size as a reaction to unstable market situations is a great competitive advantage for freight forwarders. However, analyzing changing conditions and prices on transportation markets with different freight tariffs is beyond the scope of this thesis. In these studies, parameter values for freight charges are not varied, because the computational study focuses on the effects of varying the fleet size for different location structures. Due to the parameter values defined in this thesis for calculating freight charges, it can be observed that reducing the private fleet size below 70 % always leads to increased costs (see Table 7.5). Table 7.5 shows that the costs of the clustered instances increase by about 5.2 % when reducing the private fleet from 70 % to 25 %. Furthermore, it can be conducted that the costs increase from the clustered to the random and even more to the randomly clustered instances. By considering Table 7.4 and Table 7.5, it can be derived that freight forwarders, who operate in scenarios with a clustered structure and few $P1$ requests, are in a better situation with respect to their flexibility for reducing the fleet size. In their situation, the total percentaged increase of costs $\mathcal{G}_t$ due to forwarding limitations and a reduced private fleet is relatively low compared to other location structures. For example, by assuming a freight forwarder who is reducing the private fleet to 25 % by considering 5 % $P1$ requests (three requests) and 95 % $EX$ requests (52 requests) then there is a total percentaged cost increase of $(1.71\,\% \cdot 3) + 5.20\,\% = 10.33\,\%$ in a clustered structure. By applying the same setting for a random structure, it results in $(1.15\,\% \cdot 3) + 12.43\,\% = 15.88\,\%$ additional costs. The used figures are given in Table 7.4 and Table 7.5.

In this chapter, a point estimate is used for evaluating the samples. A point estimate is more practical in terms of deriving conclusions. However, an interval estimate has the advantage of meeting the real mean with a certain confidence. In the case that the population is normal and the standard deviation of the population is unknown, the so called t-test can be applied to compute confidence intervals. Thereby, the central limit theorem ensures that a population is normal if the sample size is at least 30 units (Jones, 2012). In Figure 7.2, four confidence intervals with a confidence coefficient of 95 % are given for the instance $c102$ with $P1$ requests and a private fleet of 40 % (see Table 7.4).

Each confidence interval is represented by a lower and an upper bound. The intervals are ordered in ascending order in terms of the ratio of mandatory requests. The confidence intervals can be interpreted as follows. By solving 30 samples with a ratio of 5 % $P1$ requests, it can be derived that the real mean ($\mathcal{G}_a$) is between 4.62 % and 4.99 % with 95 % confidence. The detailed confidence intervals of instance $c102$, $c105$, $c107$, $r102$, $r105$, $r107$, $rc102$, $rc105$, and $rc107$ are given for the private fleet size of 40 % in Table 7.6, 55 % in Table 7.7, and 70 % in Table 7.8.

As a general conclusion from analyzing the relationship between the location structures and the fleet sizes is that freight forwarders with clustered instances need a larger ratio of vehicles, which can be used for mandatory requests, compared to competitors with random or randomly clustered location structures. Otherwise, it might be difficult to offer competitive prices for mandatory requests. When a freight forwarder has a large

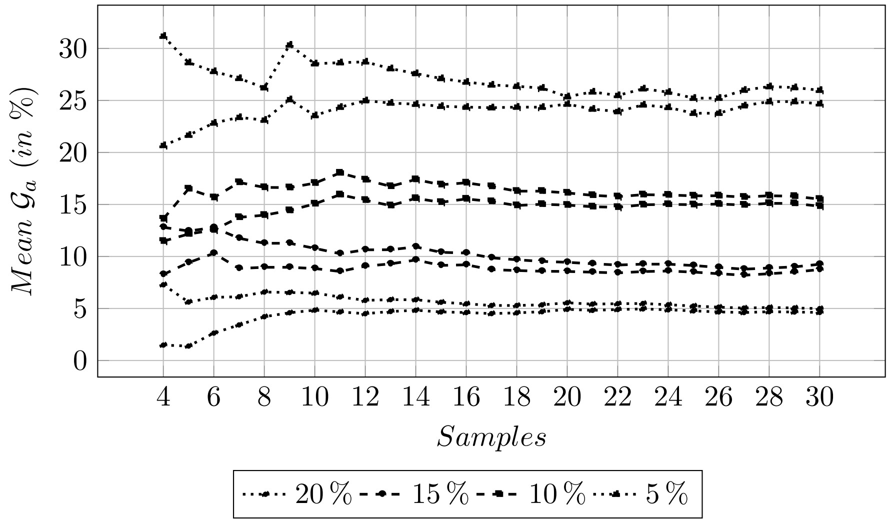


***Figure 7.2.:*** *Development of confidence intervals (instance c102 with 40 % private fleet), cf. Ziebuhr and Kopfer (2016a)*

fleet size for fulfilling mandatory requests, operating on a market with clustered requests might be a competitive advantage. Otherwise, operating on a transportation market with a random or randomly clustered location structure is favorable if the mentioned fleet size is lower than 70 %. This chapter reports on investigations of an IOTPP-FL. To solve

***Table 7.6.:*** *Interval estimate by 40 % private fleet (confidence coefficient 95 %)*

| private fleet | | *P2* requests | | | | *P1* requests | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 40 % | 5 % | 10 % | 15 % | 20 % | 5 % | 10 % | 15 % | 20 % |
| c102 | mean | 0.11 | 0.27 | 0.35 | 1.04 | 4.31 | 7.12 | 15.01 | 22.00 |
| | lower bound | 0.10 | 0.24 | 0.32 | 0.99 | 4.14 | 6.89 | 14.69 | 21.53 |
| | upper bound | 0.13 | 0.30 | 0.39 | 1.08 | 4.48 | 7.36 | 15.34 | 22.46 |
| c105 | mean | 0.94 | 1.29 | 1.89 | 2.55 | 4.81 | 8.99 | 15.17 | 25.33 |
| | lower bound | 0.85 | 1.19 | 1.79 | 2.46 | 4.62 | 8.74 | 14.83 | 24.68 |
| | upper bound | 1.02 | 1.38 | 1.99 | 2.64 | 4.99 | 9.25 | 15.51 | 25.98 |
| c107 | mean | 0.55 | 1.25 | 2.13 | 1.99 | 5.42 | 9.49 | 13.72 | 20.59 |
| | lower bound | 0.48 | 1.16 | 2.03 | 1.88 | 5.28 | 9.25 | 13.45 | 20.25 |
| | upper bound | 0.62 | 1.35 | 2.24 | 2.09 | 5.56 | 9.74 | 14.00 | 20.93 |
| r102 | mean | 0.62 | 1.48 | 1.48 | 2.76 | 1.57 | 3.86 | 5.34 | 8.13 |
| | lower bound | 0.57 | 1.39 | 1.39 | 2.65 | 1.49 | 3.73 | 5.17 | 7.95 |
| | upper bound | 0.68 | 1.58 | 1.58 | 2.87 | 1.64 | 4.00 | 5.51 | 8.31 |
| r105 | mean | 1.48 | 2.24 | 3.03 | 4.35 | 2.89 | 4.25 | 7.95 | 13.69 |
| | lower bound | 1.37 | 2.13 | 2.91 | 4.23 | 2.77 | 4.12 | 7.76 | 13.23 |
| | upper bound | 1.58 | 2.35 | 3.14 | 4.48 | 3.00 | 4.39 | 8.15 | 14.14 |
| r107 | mean | 0.29 | 0.25 | 0.43 | 0.80 | 2.32 | 5.15 | 11.29 | 15.69 |
| | lower bound | 0.23 | 0.23 | 0.40 | 0.73 | 2.22 | 5.01 | 10.99 | 15.39 |
| | upper bound | 0.36 | 0.27 | 0.45 | 0.87 | 2.42 | 5.29 | 11.59 | 15.98 |
| rc102 | mean | 0.92 | 1.42 | 2.26 | 3.92 | 1.25 | 2.15 | 5.55 | 13.99 |
| | lower bound | 0.84 | 1.33 | 2.16 | 3.80 | 1.17 | 2.06 | 5.33 | 13.06 |
| | upper bound | 0.99 | 1.51 | 2.36 | 4.03 | 1.34 | 2.24 | 5.76 | 14.92 |
| rc105 | mean | 1.21 | 1.73 | 2.32 | 4.42 | 2.13 | 3.30 | 6.61 | 14.26 |
| | lower bound | 1.11 | 1.60 | 2.21 | 4.22 | 2.01 | 3.15 | 6.43 | 13.47 |
| | upper bound | 1.31 | 1.86 | 2.43 | 4.62 | 2.25 | 3.45 | 6.78 | 15.05 |
| rc107 | mean | 1.36 | 1.48 | 4.52 | 5.10 | 2.68 | 4.77 | 14.57 | 17.88 |
| | lower bound | 1.25 | 1.37 | 4.38 | 4.93 | 2.56 | 4.60 | 13.62 | 17.30 |
| | upper bound | 1.46 | 1.59 | 4.67 | 5.27 | 2.81 | 4.94 | 15.52 | 18.47 |

the problem, the CGB-HER is alternatively applied with an end-of-pipe procedure or one of the two integrated procedures for considering forwarding limitations. This chapter claims to contribute to the following three topics: (a) development of new solution strategies for problems with forwarding limitations, (b) outperforming the best-known heuristic approach for problems with forwarding limitations, and (c) presenting a detailed

**Table 7.7.:** *Interval estimate by 55 % private fleet (confidence coefficient 95 %)*

| private fleet | | $P2$ requests | | | | $P1$ requests | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 55 % | 5 % | 10 % | 15 % | 20 % | 5 % | 10 % | 15 % | 20 % |
| c102 | mean | 0.00 | 0.00 | 0.00 | 0.00 | 1.52 | 2.82 | 7.66 | 9.92 |
| | lower bound | 0.00 | 0.00 | 0.00 | 0.00 | 1.41 | 2.67 | 7.43 | 9.73 |
| | upper bound | 0.00 | 0.00 | 0.00 | 0.00 | 1.62 | 2.97 | 7.88 | 10.11 |
| c105 | mean | 0.12 | 0.10 | 0.14 | 0.24 | 0.80 | 2.88 | 4.95 | 9.89 |
| | lower bound | 0.11 | 0.09 | 0.13 | 0.23 | 0.73 | 2.70 | 4.73 | 9.62 |
| | upper bound | 0.13 | 0.11 | 0.15 | 0.25 | 0.87 | 3.06 | 5.16 | 10.17 |
| c107 | mean | 0.07 | 0.14 | 0.20 | 0.19 | 0.88 | 3.07 | 5.17 | 8.95 |
| | lower bound | 0.06 | 0.13 | 0.19 | 0.18 | 0.82 | 2.91 | 4.95 | 8.74 |
| | upper bound | 0.08 | 0.15 | 0.21 | 0.20 | 0.94 | 3.23 | 5.39 | 9.17 |
| r102 | mean | 0.71 | 1.50 | 2.01 | 2.68 | 1.00 | 2.34 | 2.95 | 4.44 |
| | lower bound | 0.65 | 1.42 | 1.92 | 2.59 | 0.93 | 2.23 | 2.83 | 4.31 |
| | upper bound | 0.77 | 1.59 | 2.10 | 2.78 | 1.07 | 2.44 | 3.08 | 4.56 |
| r105 | mean | 0.91 | 1.33 | 2.04 | 3.00 | 0.91 | 1.33 | 2.04 | 3.00 |
| | lower bound | 0.84 | 1.26 | 1.96 | 2.91 | 0.84 | 1.26 | 1.96 | 2.91 |
| | upper bound | 0.98 | 1.39 | 2.13 | 3.09 | 0.98 | 1.39 | 2.13 | 3.09 |
| r107 | mean | 0.16 | 0.25 | 0.48 | 0.59 | 0.16 | 0.25 | 0.48 | 0.59 |
| | lower bound | 0.14 | 0.22 | 0.45 | 0.56 | 0.14 | 0.22 | 0.45 | 0.56 |
| | upper bound | 0.18 | 0.28 | 0.51 | 0.62 | 0.18 | 0.28 | 0.51 | 0.62 |
| rc102 | mean | 0.74 | 1.14 | 1.69 | 2.80 | 0.80 | 1.33 | 2.38 | 4.67 |
| | lower bound | 0.68 | 1.05 | 1.61 | 2.70 | 0.73 | 1.25 | 2.29 | 4.52 |
| | upper bound | 0.80 | 1.23 | 1.78 | 2.89 | 0.86 | 1.42 | 2.47 | 4.82 |
| rc105 | mean | 1.37 | 1.86 | 2.54 | 4.41 | 2.09 | 2.86 | 4.48 | 8.22 |
| | lower bound | 1.27 | 1.73 | 2.42 | 4.22 | 1.97 | 2.70 | 4.32 | 7.93 |
| | upper bound | 1.48 | 2.00 | 2.65 | 4.59 | 2.21 | 3.01 | 4.65 | 8.50 |
| rc107 | mean | 1.14 | 1.12 | 3.13 | 3.59 | 2.01 | 2.73 | 5.89 | 9.17 |
| | lower bound | 1.05 | 1.04 | 3.01 | 3.46 | 1.89 | 2.59 | 5.71 | 8.89 |
| | upper bound | 1.22 | 1.21 | 3.25 | 3.73 | 2.12 | 2.88 | 6.07 | 9.45 |

**Table 7.8.:** *Interval estimate by 70 % private fleet (confidence coefficient 95 %)*

| private fleet | | $P2$ requests | | | | $P1$ requests | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 70 % | 5 % | 10 % | 15 % | 20 % | 5 % | 10 % | 15 % | 20 % |
| c102 | mean | 0.00 | 0.00 | 0.00 | 0.00 | 0.65 | 1.58 | 4.67 | 5.96 |
| | lower bound | 0.00 | 0.00 | 0.00 | 0.00 | 0.58 | 1.47 | 4.50 | 5.81 |
| | upper bound | 0.00 | 0.00 | 0.00 | 0.00 | 0.71 | 1.69 | 4.84 | 6.11 |
| c105 | mean | 0.00 | 0.00 | 0.00 | 0.00 | 0.44 | 1.40 | 2.27 | 5.54 |
| | lower bound | 0.00 | 0.00 | 0.00 | 0.00 | 0.40 | 1.28 | 2.13 | 5.35 |
| | upper bound | 0.00 | 0.00 | 0.00 | 0.00 | 0.48 | 1.51 | 2.42 | 5.74 |
| c107 | mean | 0.00 | 0.00 | 0.00 | 0.00 | 0.46 | 1.23 | 2.51 | 5.16 |
| | lower bound | 0.00 | 0.00 | 0.00 | 0.00 | 0.43 | 1.11 | 2.34 | 4.98 |
| | upper bound | 0.00 | 0.00 | 0.00 | 0.00 | 0.49 | 1.34 | 2.67 | 5.35 |
| r102 | mean | 0.43 | 0.91 | 1.34 | 1.64 | 0.46 | 1.11 | 1.45 | 2.08 |
| | lower bound | 0.40 | 0.85 | 1.27 | 1.57 | 0.42 | 1.03 | 1.37 | 1.98 |
| | upper bound | 0.47 | 0.97 | 1.42 | 1.72 | 0.50 | 1.19 | 1.54 | 2.17 |
| r105 | mean | 0.55 | 0.71 | 1.28 | 2.08 | 0.61 | 0.75 | 1.44 | 2.50 |
| | lower bound | 0.49 | 0.66 | 1.21 | 2.01 | 0.55 | 0.70 | 1.36 | 2.41 |
| | upper bound | 0.61 | 0.75 | 1.36 | 2.16 | 0.66 | 0.80 | 1.53 | 2.59 |
| r107 | mean | 0.00 | 0.00 | 0.00 | 0.00 | 0.68 | 1.96 | 2.80 | 3.51 |
| | lower bound | 0.00 | 0.00 | 0.00 | 0.00 | 0.60 | 1.84 | 2.64 | 3.37 |
| | upper bound | 0.00 | 0.00 | 0.00 | 0.00 | 0.77 | 2.07 | 2.95 | 3.65 |
| rc102 | mean | 0.75 | 1.16 | 1.71 | 2.79 | 0.79 | 1.27 | 1.94 | 3.26 |
| | lower bound | 0.69 | 1.07 | 1.62 | 2.69 | 0.73 | 1.18 | 1.85 | 3.16 |
| | upper bound | 0.82 | 1.25 | 1.80 | 2.89 | 0.86 | 1.36 | 2.02 | 3.37 |
| rc105 | mean | 0.92 | 1.31 | 1.93 | 3.21 | 1.13 | 1.53 | 2.62 | 4.76 |
| | lower bound | 0.85 | 1.22 | 1.83 | 3.07 | 1.05 | 1.41 | 2.49 | 4.55 |
| | upper bound | 0.99 | 1.41 | 2.02 | 3.36 | 1.21 | 1.65 | 2.75 | 4.98 |
| rc107 | mean | 0.90 | 0.92 | 2.17 | 2.59 | 1.39 | 2.00 | 3.53 | 4.94 |
| | lower bound | 0.84 | 0.85 | 2.09 | 2.50 | 1.29 | 1.87 | 3.38 | 4.75 |
| | upper bound | 0.97 | 0.99 | 2.25 | 2.68 | 1.50 | 2.13 | 3.68 | 5.13 |

computational study demonstrating how much the costs of a freight forwarder increase due to forwarding limitations under different transportation scenarios. All solution strategies lead to appending additional restrictions resulting from forwarding limitations to the CGB-HER. Three ways of installing the additional restrictions in the CGB-heuristic are compared. The results of the computational studies demonstrate that integrating the rules for respecting the additional restrictions to the master problem (route selection) of the solution approach is the best of the three methods. Installing approaches for integrating these restrictions during the subproblem (route generation) is the second best solution strategy, while enforcing the compliance to these restrictions at the end (after the entire process of the CGB-HER) is by far the worst alternative. Furthermore, the impact of the fleet size and the location structure is analyzed. Besides a calculated pattern, which can be used as a decision instrument for transportation planning, several findings are derived. For example, two preferable scenarios are identified. In the case that the available fleet size for mandatory requests is less than or equal to 70 % of the total vehicle fleet, random and randomly clustered location structures are preferable. On other hand clustered location structures are preferable in the case that the available fleet size is about 100 % of the total vehicle fleet. The available fleet contains all vehicles which can be used for fulfilling mandatory requests. In case of $P1$ requests, the available fleet contains private vehicles while in case of $P2$ requests the available fleet contains private and rented vehicles.

# 8. Collaborative Transportation Planning

This chapter deals with a TPP with pickup and delivery requests, where a freight forwarder can choose between self-fulfillment and collaboration for fulfilling a request. This problem is known as CTP problem (CTPP). In CTP, independent freight forwarders align their transportation plans by exchanging requests within a horizontal coalition. The goal of the freight forwarders is to increase their profitability and flexibility in competitive markets with high demand fluctuations. In recent publications, it is assumed that each request can be fulfilled by any freight forwarder. The cherry-picking procedure is often applied for identifying which request should be fulfilled by the private fleet and which should be fulfilled by a freight forwarder. This assumption represents a simplification because there are requests in practice which are prohibited to be outsourced due to contractual agreements.

The contribution of this chapter is to identify the increase of costs for solving a CTPP with exchangeable and mandatory requests. The problem is denoted as CTPP with forwarding limitations (CTPP-FL). To analyze the impact of mandatory requests for the CTPP-FL, the CGB-HER with two solution strategies for handling mandatory requests is applied and investigated.

This chapter contains two sections. Section 8.1 describes how the existing CTPP of Chapter 5 can be extended to the CTPP-FL. Furthermore, it is described how the CGB-HER with the corresponding solution strategies has to be modified in order to solve the CTPP-FL. The CTPP-FL is presented based on an example to get a better idea of the underlying TPP. Section 8.2 presents the computational studies of this chapter. In Subsection 8.2.1, two studies are presented. One study calculates reference values for CTPP, while the second one is used to determine the best solution strategy for handling mandatory requests. In Subsection 8.2.2, the CGB-HER with the best solution strategy is applied with the goal of determining the impact of mandatory requests for the CTPP-FL. This chapter is based on Ziebuhr and Kopfer (2015).

## 8.1. Mathematical formulation and solution approach

The considered CTPP-FL is based on the CTPP of Chapter 5, which is extended by two different types of mandatory requests. The underlying problem of the CTPP-FL is a PDPTW, where transportation capacities can be exchanged within a horizontal coalition of freight forwarders. It is proposed to consider the following transportation services in the CTPP-FL with their corresponding fulfillment modes:

- $P1$ service: self-fulfillment,
- $P3$ service: self-fulfillment, collaboration,

- *EX* service: self-fulfillment, collaboration, CC.

The corresponding requests are denoted as *P*1 request, *P*3 request, and *EX* request. A *EX* request can be fulfilled by any fulfillment mode. A *P*3 request can be fulfilled by the private fleet of any freight forwarder and a *P*1 request can only be fulfilled by the private fleet of the request owner. In this case, *P*1 and *P*3 requests represent mandatory requests, which have in common that they cannot be fulfilled by a CC. The option of employing CCs is still used in this scenario. Otherwise it would be impossible to identify feasible solutions for some test cases. For example, in case that a freight forwarder wins more transportation requests through the request exchange than he or she can fulfill with his or her private fleet.

In Figure 8.1, an example of the CTPP-FL is illustrated. In this scenario, two freight forwarders are in charge of an individual request portfolio with in total 13 requests. As in the previous chapter, a VRPTW is used as a basic vehicle routing problem instead of a PDPTW.

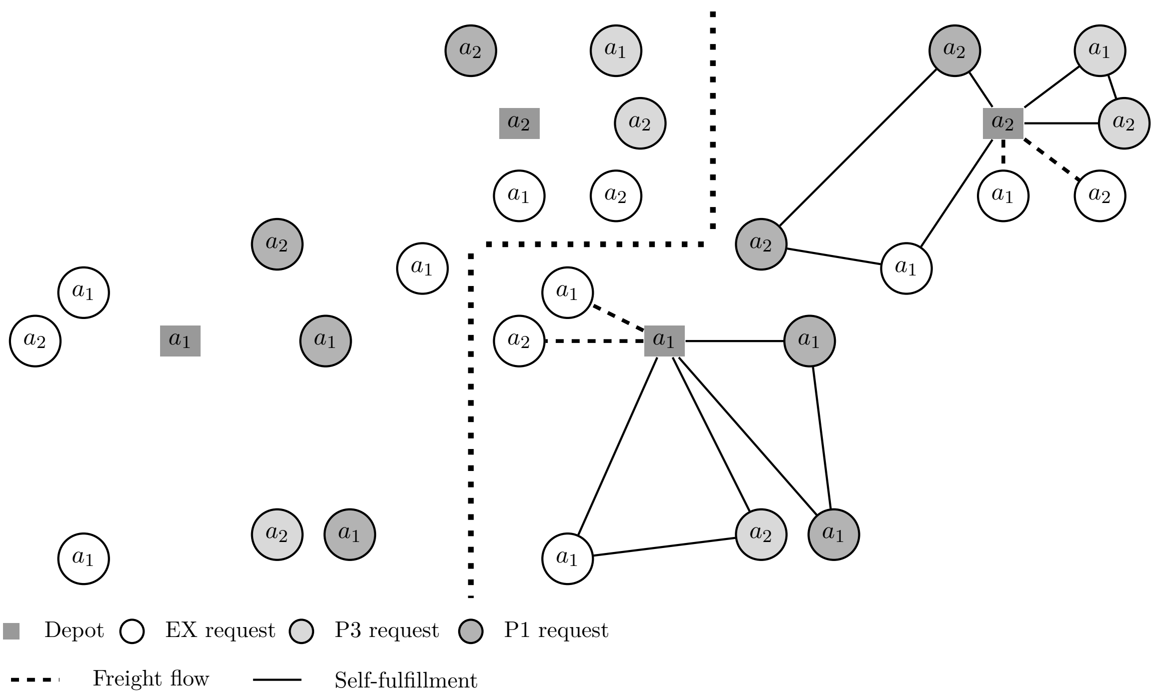


***Figure 8.1.:*** *CTP problem with exchangeable and mandatory requests*

The request portfolio of the freight forwarder $a_1$ contains: four *EX* requests, one *P*3 request, and two *P*1 requests, while the request portfolio of freight forwarder $a_2$ contains: two *EX* requests, two *P*3 requests, and two *P*1 requests. Both freight forwarders are in charge of two private vehicles for fulfilling these requests. Each of these vehicle types has a limited capacity and different cost rates. Furthermore, both freight forwarders are able to employ a CC for fulfilling a request. Thus, the goal is to identify this transportation plan which minimizes the individual transportation costs of both freight forwarders by ensuring that mandatory requests are fulfilled by their corresponding fulfillment mode. In Figure 8.1, the TPP on the left side as well as a solution for the TPP on the right side are presented. By considering the locations of the depots and requests, it can be observed that some requests within the coalition are located close to the depot of a different freight forwarder. It means that it could be promising to exchange requests with each other. This assumption is verified by considering the solution of this CTPP-FL, where freight forwarder $a_1$ gets two requests from freight forwarder $a_2$ (one $EX$ request and one $P3$ request) while freight forwarder $a_2$ gets three requests from freight forwarder $a_1$ (two $EX$ requests and one $P3$ request). As can be seen, $EX$ requests can be fulfilled by any fulfillment mode, while $P3$ requests can be fulfilled by the private vehicle fleet of any freight forwarder and $P1$ requests can only be fulfilled by the private fleet of the request owner.

The CGB-HER is applied to solve the CTPP-FL, which is either extended by the strict generation or strict composition procedure for handling mandatory requests. The repair procedure is obsolete due to their performance in the previous chapter. Each of these solution strategies is described in detail in Section 6.1. In the following, it is described which mathematical model is solved by the subproblem or master problem of the CGB-HER in terms of the different solution strategies.

This chapter looks at the described PDPTW from a collaborative perspective, where $m$ freight forwarders within coalition $F$ align their transportation plans by exchanging pickup and delivery requests. During the collaboration, each freight forwarder $c \in F$ offers his or her entire request portfolio $P_c$ ($P = \sum_{c=1}^{m} P_c$) for exchange and receives a new portfolio $P'_c$ after the allocation process is completed. Depending on the transportation plans of the freight forwarders, each freight forwarder $c \in F$ transfers a set of requests ($P_c^-$) to the coalition and receives a set of requests ($P_c^+$) from the coalition. It means that each freight forwarder $c \in F$ solves a PDPTW with the individual request portfolio $P'_c$. A freight forwarder has a private fleet, which is sufficient for fulfilling all of their requests of their original request portfolio, but due to the collaboration it is possible that a freight forwarder receives more requests than he or she can fulfill. Thus, each freight forwarder $c \in F$ has to add the term $+\sum_{i \in P} \gamma_i y_i^{\mathrm{CC}}$ in the objective function (5.1) and constraints (5.2) have to be modified by adding the term $+y_i^{\mathrm{CC}}$, which enables the option of employing a CC on a spot market. Some requests would remain unassigned without this modification of the subproblem. This CTPP has to be extended to handle mandatory requests. One option is by modifying the subproblem, while the other one relies on the modification of the master problem. In case of the collaborative perspective, the mathematical formulation with objective function (5.35) and the constraints (5.36)–(5.37) do not have to be changed in terms of both solution strategies.

**Strict generation procedure:** The strict generation procedure solves a CTPP-FL as a subproblem by forwarding just feasible vehicle routes to the master problem, the master problem can be solved as a CTPP. First of all, it is proposed to separate the request portfolio of the coalition $P$ into three disjoint sets: $EX$ requests, $P1$ requests, and $P3$ requests with $P = EX \cup P1 \cup P3$. Thereby, it is ensured that each freight forwarder $c \in F$ does not offer his or her set of $P1$ requests for exchange. It means that each freight forwarder $c \in F$ receives a new portfolio $P'_c$ with winning $EX'_c$ and $P3'_c$ request sets after the allocation process is completed and is supplemented by the individual set of $P1$ requests $P1_c$, with $P1 = \cup_{c=1}^{m} P1_c$. The individual request portfolio of the freight forwarder $c$ contains the request sets $EX'_c \cup P1_c \cup P3'_c$. Then, the subproblem with mandatory requests can be generated by replacing constraints (5.2) of the known PDPTW formulation of Subsection 5.2.1 by the following constraints, which define the applicable fulfillment modes for each type of requests. Obviously, the known PDPTW sets have to be considered for each freight forwarder $c \in F$ separately.

$$\sum_{k \in K_c} \sum_{j \in V_c} x_{ijk} + y_i^{CC} = 1\,, \qquad \forall i \in EX'_c, \tag{8.1}$$

$$\sum_{k \in K_c^1} \sum_{j \in V_c} x_{ijk} = 1\,, \qquad \forall i \in P1_c \cup P3'_c. \tag{8.2}$$

Constraints (8.1) ensure that a $EX$ request can be fulfilled either by the private fleet or by employing a CC, while constraints (8.2) ensure that a $P1$ or $P3$ request can only be fulfilled by the private fleet $K_c^1$. In case of the collaborative perspective, the set of offered requests has to be modified as mentioned in order that the individual $P1$ requests are not offered for exchange. In summary, the strict generation procedure solves the following subproblem and SPP master problem.

- Subproblem: The objective function (5.1) with freight charges, the constraints (5.3)–(5.15), and the forwarding limitation constraints (8.1)–(8.2) are considered.
- SPP master problem: The objective function (5.38) without freight charges and the constraints (5.39)–(5.41) are considered.

In case of the strict generation procedure for the CTPP-FL, it is observed that each freight forwarder always generates vehicle routes where all $P3$ requests are served by self-fulfillment in order that similar routes are generated. This issue is solved by eliminating the penalty costs in case that all requests are visible for each freight forwarder.

**Strict composition procedure:** The strict composition procedure mainly solves the CTPP as subproblem and the CTPP-FL as master problem. Feasible solutions for the CTPP-FL are generated by using the dual values, which are generated by solving the master problem. As known, the strict generation procedure is applied in the first round of the iterative procedure of the CGB-HER.

Some modifications are necessary for handling mandatory requests by the master problem. Similar to the strict generation procedure, the request portfolio of the coalition $P$ is also separated into three disjoint sets corresponding to the request type, $P'_c = EX'_c \cup P1_c \cup P3'_c$. It is proposed to introduce two new parameters in the master problem. The parameter $b_r^{\text{FC}}$ indicates for each submitted vehicle route $r \in R$ the original freight forwarder, while the parameter $b_i^C$ indicates for each request with pickup node $i \in P$ the requested freight forwarder in terms of the attached service. The corresponding SPP master problem of the CTPP-FL can be defined as follows:

$$\min C_c = \sum_{r \in R_c} c_r u_r \tag{8.3}$$

$$\sum_{r \in R_c} a_{ir} u_r + u_i^{\mathrm{CC}} = 1\,, \qquad \forall i \in EX'_c, \tag{8.4}$$

$$\sum_{r \in R_c} a_{ir} u_r = 1\,, \qquad \forall i \in P1_c \cup P3'_c, \tag{8.5}$$

$$\sum_{r \in R_c} a_{ir} u_r b_r^{\mathrm{FC}} = b_i^C\,, \qquad \forall i \in P1_c, \tag{8.6}$$

$$\sum_{r \in R_c} f_{er} u_r \leq |K_c^e|\,, \qquad \forall e \in E, \tag{8.7}$$

$$u_r \in \{0, 1\}\,, \qquad \forall r \in R_c. \tag{8.8}$$

The goal of the master problem is to minimize the transportation costs defined by objective function (8.3) by observing different constraints. Constraints (8.4) define that each fulfillment mode can be used for $EX$ requests, while $P1$ and $P3$ requests have to be fulfilled by a private vehicle. Constraints (8.5) ensure the application of a vehicle instead of using a CC. constraints (8.6) are used to ensure that a private vehicle of the original freight forwarder $c$ is used for a $P1$ request. The remaining constraints are known from the previous chapters. In summary, the strict composition procedure solves the following subproblem and SPP master problem.

- Subproblem: The objective function (5.1) with freight charges and the constraints (5.2)–(5.15) are considered, where constraint (5.2) is extended by the option of employing a CC.
- SPP master problem: The objective function (8.3) and the constraints (8.4)–(8.8) are considered.

## 8.2. Computational studies

A CTPP-FL with two types of mandatory requests is solved in this section. The results of two computational studies are presented. First, reference values are generated. These reference values are required because the existing solution approach for the CTPP of Wang and Kopfer (2014) uses a different LNS. Then, it is determined which solution strategy is preferred for the CTPP-FL. Second, the main computational study is presented, where the increase of transportation costs in terms of mandatory requests is determined.

The applied instances differ from the previous chapter. Here, the CTPP instances of Wang and Kopfer (2014) are used, where two to five PDPTW instances with the same characteristic ($C$, $R$, $RC$) and size (100 customers) are combined to one instance. Thereby, it is ensured that each PDPTW instance represents one freight forwarder with the corresponding requests (referred to as region). CTPP instances are generated by the consolidation of regions where all nodes of one region are adjusted with the same amount. The applied fleet sizes of each freight forwarder are set to the number of used vehicles in the best-known solutions, while the fixed costs are set to zero and the variable costs are set to one per distance unit. In total, 24 CTPP instances are generated. It is worth mentioning that the CTPP instances with five freight forwarders are skipped in this chapter due to the high computational effort. Obviously, it is necessary to extend the CTPP instances by the definition of mandatory requests. As in the previous chapter, different ratios of mandatory requests are generated. However, this chapter uses higher ratios (10 %, 20 %, and 30 % mandatory requests) as the previous chapter due to the fact that these samples are still feasible. A ratio of 20 % means that 20 % of all requests are mandatory requests and 80 % are exchangeable requests. The decision, which request is chosen as a mandatory request, is executed randomly. The number of samples is reduced to 15 samples for each ratio and instance. As in the previous chapters, scenarios with exchangeable requests combined with one type of mandatory requests are investigated. It means that the different types of mandatory requests are not mixed.

As an evaluation criterion, the percentaged increase of costs between the CTPP-FL solution $f(EX \cup P1 \text{ or } P3)$ computed by the CGB-HER with the best solution strategy and the best-known CTPP solution $f^b(P^*)$ is used. It means that the same evaluation criterion as in the previous chapter is used, which is defined by the variable $\mathcal{G}_a$.

The CGB-heuristic is implemented in C++ (Visual Studio 2012) and the computational studies are executed on a Windows 7 PC with Intel Core i7-2600 processor (3.4 GHz and 16 GB of main memory). The dual values are computed by using CPLEX (version 12.5.1). The same parameter setting is used as in Chapter 6 except for the extended time limit of the commercial solver (300s) and reduced iterations during the ALNS (5,000 iterations).

### 8.2.1. Setup of computational studies

Before the main study can be performed, reference values and best solution strategy for handling mandatory requests have to be determined. The reference values are calculated by using all of the mentioned CTPP instances are used, while the best solution strategy is identified by a subset of the CTPP instances.

First, the solution quality of the CGB-HER has to be verified for the CTPP instances of Wang and Kopfer (2014). Thereby, three scenarios have to be distinguished: IP, CP, and collaborative planning. The best-known IP solution values are reported by SINTEF and do not have to be calculated. Therefore, the individual PDPTW solution values are summarized. The solutions of the remaining scenarios have to be calculated. Wang and Kopfer (2014) use a CGB-heuristic with an LNS for the collaborative planning approach and an LNS is used for the CP approach. The former solution approach uses ten rounds in the iterative procedure with 5,000 iterations and is applied once per instance, while the latter one uses 15,000 iterations and is applied three times. In Table 8.1, the best-known IP solution values reported by SINTEF, CP and CTP solution values computed by the CGB-heuristic from Wang and Kopfer (2014), and CP and CTP solution values computed by the CGB-HER are presented. The CGB-HER is applied three times and the best solution values are listed. In general, it is expected that IP leads to the highest costs, CP leads to the lowest costs, and collaborative planning is between these bounds. The lowest solution values in case of CP and collaborative planning are marked in bold.

***Table 8.1.:*** *Reference values for IP, CP, and collaborative planning, cf. Ziebuhr and Kopfer (2015)*

| instance | | | IP | CP | | collaborative planning | |
|---|---|---|---|---|---|---|---|
| id | m | n | SINTEF | LNS | ALNS | LNS | ALNS |
| c101 | 2 | 105 | 1,864.29 | 1,699.08 | **1,669.87** | 1,691.63 | **1,669.87** |
| c102 | 2 | 106 | 1,655.38 | **1,539.82** | **1,539.82** | **1,539.82** | **1,539.82** |
| c103 | 3 | 159 | 2,658.48 | 2,414.99 | **2,372.23** | 2,391.34 | **2,372.23** |
| c104 | 3 | 159 | 2,517.89 | 2,198.10 | **2,190.70** | 2,194.32 | **2,190.70** |
| c105 | 4 | 212 | 3,518.49 | 3,027.86 | **2,984.95** | 3,041.51 | **2,984.35** |
| c106 | 4 | 211 | 3,519.67 | 2,882.58 | **2,850.53** | 2,867.63 | **2,857.05** |
| r101 | 2 | 104 | 2,452.03 | **2,263.21** | 2,276.33 | **2,269.32** | 2,276.33 |
| r102 | 2 | 104 | 2,363.93 | 2,311.98 | **2,262.78** | 2,311.77 | **2,262.78** |
| r103 | 3 | 160 | 3,600.24 | 3,394.24 | **3,382.52** | 3,444.36 | **3,375.60** |
| r104 | 3 | 154 | 3,239.63 | 3,014.98 | **2,998.63** | 3,041.63 | **3,005.42** |
| r105 | 4 | 208 | 4,434.32 | **3,906.01** | 3,907.74 | 3,960.58 | **3,959.87** |
| r106 | 4 | 215 | 5,624.38 | 5,038.12 | **5,016.92** | 5,112.16 | **5,027.05** |
| rc101 | 2 | 106 | 2,488.89 | 2,386.78 | **2,378.99** | **2,378.99** | **2,378.99** |
| rc102 | 2 | 107 | 2,867.77 | 2,696.04 | **2,690.49** | **2,690.49** | **2,690.49** |
| rc103 | 3 | 160 | 3,945.21 | 3,511.99 | **3,457.16** | 3,544.73 | **3,450.44** |
| rc104 | 3 | 161 | 4,190.75 | 3,650.29 | **3,641.66** | 3,693.64 | **3,641.20** |
| rc105 | 4 | 211 | 5,345.12 | 4,539.76 | **4,489.57** | 4,680.19 | **4,489.57** |
| rc106 | 4 | 213 | 5,341.35 | 4,777.83 | **4,740.34** | 4,881.84 | **4,740.34** |

In this study, 29 new best-known solutions for the CP and collaborative planning are identified. 15 new best-known solutions for CP and 14 new best-known solutions for collaborative planning are identified. It means that it is preferable to use an ALNS instead of an LNS within the column generation approach. Even if the instances are solved just once by the CGB-HER, the solution approach of Wang and Kopfer (2014) is still outperformed by identifying 14 new best-known solutions for collaborative planning. The best collaborative planning solution values are used to determine the percentaged increase of costs per mandatory request in Subsection 8.2.2.

Second, the best solution strategy for handling mandatory requests in the CTPP-FL has to be determined. The strict generation procedure is compared with the strict composition procedure for all instances with 20 % $P3$ or $P1$ requests. It is identified that on average a freight forwarder has to charge about 0.27 % additional costs per 20 % $P3$ requests and about 8.25 % additional costs per 20 % $P1$ requests by using the strict composition procedure and about 6.13% additional costs per 20 % $P3$ requests and about 5.60 % additional costs per 20 % $P1$ requests by using the strict generation procedure. As can be seen, the strict composition procedure is preferable for $P3$ requests, while the strict generation procedure is preferable for $P1$ requests. The strict composition procedure seems to be superior in case that several fulfillment modes are applicable for request fulfillment, which occurs for $P3$ requests and not for $P1$ requests. The computational effort of both solution strategies is comparable and is not used as an evaluation criterion.

### 8.2.2. Impact of mandatory requests

The impact of mandatory requests in case of the CTPP-FL is analyzed by solving all instances with two to four freight forwarders by the strict generation procedure for $P1$ requests or the strict generation procedure for $P3$ requests. To determine the percentaged increase of costs per mandatory request, the best CTPP solution values of Subsection 8.2.1 are used. The results are listed in Table 8.2 in column four to nine. Each CTPP-FL instance is solved once.

***Table 8.2.:*** *Mean percentaged increase of costs per mandatory request in CTPP-FL, cf. Ziebuhr and Kopfer (2015)*

| instance | | | $P1$ requests | | | $P3$ requests | | |
|---|---|---|---|---|---|---|---|---|
| id | m | n | 10 % | 20 % | 30 % | 10 % | 20 % | 30 % |
| c101 | 2 | 105 | 0.28 | 0.20 | 0.21 | 0.00 | 0.01 | 0.01 |
| c102 | 2 | 106 | 0.36 | 0.27 | 0.21 | 0.00 | 0.00 | 0.00 |
| c103 | 3 | 159 | 0.33 | 0.19 | 0.18 | 0.00 | 0.00 | 0.00 |
| c104 | 3 | 159 | 0.49 | 0.28 | 0.29 | 0.00 | 0.00 | 0.00 |
| c105 | 4 | 212 | 0.20 | 0.21 | 0.17 | 0.00 | 0.01 | 0.02 |
| c106 | 4 | 211 | 0.34 | 0.27 | 0.27 | 0.01 | 0.01 | 0.02 |
| r101 | 2 | 104 | 0.29 | 0.22 | 0.18 | 0.08 | 0.04 | 0.02 |
| r102 | 2 | 104 | 0.07 | 0.09 | 0.06 | 0.01 | 0.02 | 0.01 |
| r103 | 3 | 160 | 0.08 | 0.09 | 0.08 | 0.01 | 0.01 | 0.00 |
| r104 | 3 | 154 | 0.14 | 0.11 | 0.12 | 0.01 | 0.00 | 0.00 |
| r105 | 4 | 208 | 0.13 | 0.14 | 0.13 | 0.02 | 0.01 | 0.01 |
| r106 | 4 | 215 | 0.13 | 0.11 | 0.10 | 0.01 | 0.01 | 0.00 |
| rc101 | 2 | 106 | 0.13 | 0.09 | 0.08 | 0.02 | 0.00 | 0.00 |
| rc102 | 2 | 107 | 0.12 | 0.14 | 0.10 | 0.00 | 0.00 | 0.00 |
| rc103 | 3 | 160 | 0.18 | 0.17 | 0.16 | 0.01 | 0.01 | 0.01 |
| rc104 | 3 | 161 | 0.25 | 0.23 | 0.21 | 0.01 | 0.01 | 0.00 |
| rc105 | 4 | 211 | 0.22 | 0.18 | 0.17 | 0.02 | 0.01 | 0.00 |
| rc106 | 4 | 213 | 0.16 | 0.11 | 0.12 | 0.03 | 0.02 | 0.02 |

By solving the CTPP-FL with $P3$ requests, it is observed that the impact of these mandatory requests is not significant, while $P1$ requests lead to measurable additional costs. These additional costs depend on the location structure, number of freight forwarders, and ratios of $P1$ requests. It is observed that random location structures lead to the lowest additional costs, while clustered structures lead to the highest additional costs. This observation can be explained by the distribution of the requests. Thereby, it is more expensive to fulfill $P1$ requests in different request clusters instead of considering equally distributed requests in random structures. In terms of the ratios of mandatory requests, it is observed that the additional costs per $P1$ request decrease by increasing the ratio of $P1$ requests. The mentioned behavior can be explained by the first application of the ALNS, where it is preferable when several mandatory requests are available for route building. By increasing the number of freight forwarders $m$, different results are identified depending on the location structures. It seems that in case of random location structures additional freight forwarders reduce the impact of $P1$ requests, while additional freight forwarders for the remaining location structures with clustered requests lead to higher or lower additional costs depending on the positions of the request clusters.

In this chapter, a CTPP-FL is introduced, which represents the first problem formulation in literature where mandatory requests are considered in a CTP approach. The mentioned CGB-HER with two solution strategies is applied to solve a CTPP-FL. The best solution strategy depends on the request type. It is identified that $P1$ requests should be solved by the strict generation procedure, while the strict composition procedure should be used for $P3$ requests. In terms of the financial impact of mandatory requests, several findings could be derived like that $P1$ requests have a much higher impact on the additional costs than $P3$ requests and that random location structures lead to the lowest additional costs in this scenario.

# 9. Collaborative Integrated Operational Transportation Planning

This chapter deals with a TPP with pickup and delivery requests, where a freight forwarder can choose between self-fulfillment, subcontracting, and collaboration for fulfilling a request. This problem is known as CIOTP problem (CIOTPP). The CIOTPP combines the TPPs of Chapter 7 and 8. CIOTP is especially relevant for small and medium sized freight forwarders. These freight forwrdes are confronted with thin margins and high demand fluctuations in competitive transportation markets, where they are trying to improve their planning situation by using different external resources besides their own resources. These external resources might belong to closely related long-term carriers, CCs, or cooperating freight forwarders within a horizontal coalition. In this scenario, mandatory requests are introduced. Due to the consideration of all these fulfillment modes, it is possible to introduce new transportation services corresponding to mandatory requests. The known CGB-HER is applied to solve the CIOTPP with forwarding limitations (CIOTPP-FL). The heuristic uses either a strict generation procedure or a strict composition procedure for handling mandatory requests. Two computational studies are presented based on the CIOTPP-FL. One study analyzes the increase of transportation costs by considering each type of mandatory requests separately, while a second one determines the increase of costs by considering mandatory request combinations (i.e., mixed mandatory request types). At this stage, there is no comparable solution approach in literature for CIOTP.

First, the considered CIOTPP-FL and the solution approach are briefly described in Section 9.1. The TPP is explained based on an example, which gives a better overview about the optimization problem. Then, it is described how mandatory requests can be integrated within the CIOTPP. Second, the impact of mandatory requests is investigated for CIOTPP-FL in Section 9.2. Two computational studies are presented, which are based on the CIOTPP instances from literature. Both studies solve the CIOTPP-FL but differ in terms of the consideration of mandatory requests. A CIOTPP-FL with one type of mandatory request per instance is solved in Subsection 9.2.1, while a CIOTPP-FL with several types of mandatory requests per instance is determined in Subsection 9.2.2. This chapter is based on Ziebuhr and Kopfer (2016b) and Ziebuhr and Kopfer (2017).

## 9.1. Mathematical formulation and solution approach

The considered CIOTPP-FL contains four different types of mandatory requests. In the previous Chapters 7 and 8, scenarios with exchangeable requests and two types of mandatory requests are presented. Both chapters investigate $P1$ requests with either $P2$ or $P3$ requests. Here, these three types of mandatory requests as well as a new type are considered in the CIOTPP-FL. The following listing of transportation services with their applicable fulfillment modes is proposed for the CIOTPP-FL:

- $P1$ service: self-fulfillment,
- $P2$ service: self-fulfillment, long-term carrier,
- $P3$ service: self-fulfillment, collaboration,
- $P4$ service: self-fulfillment, long-term carrier, collaboration,
- $EX$ service: self-fulfillment, subcontracting, collaboration.

The corresponding requests of the mentioned services are denoted as $P1$ request, $P2$ request, $P3$ request, $P4$ request, and $EX$ requests. All types of requests are already known from the previous chapters except for the $P4$ requests. A $P4$ request can be fulfilled by the private fleet or long-term carrier of any freight forwarder. It means that just the employment of a CC is permitted for a $P4$ request.

A CIOTPP-FL is illustrated in Figure 9.1. There two freight forwarders are in charge of an individual request portfolio with in total 13 requests and different types of vehicles. As in the previous chapters, a VRPTW is used as a basic vehicle routing problem. The request portfolio of the freight forwarder $a_1$ contains: four $EX$ requests, one $P2$ request, and two $P4$ requests, while the request portfolio of freight forwarder $B$ contains: two $EX$ requests, one $P1$ request, two $P2$ requests, and one $P3$ request.

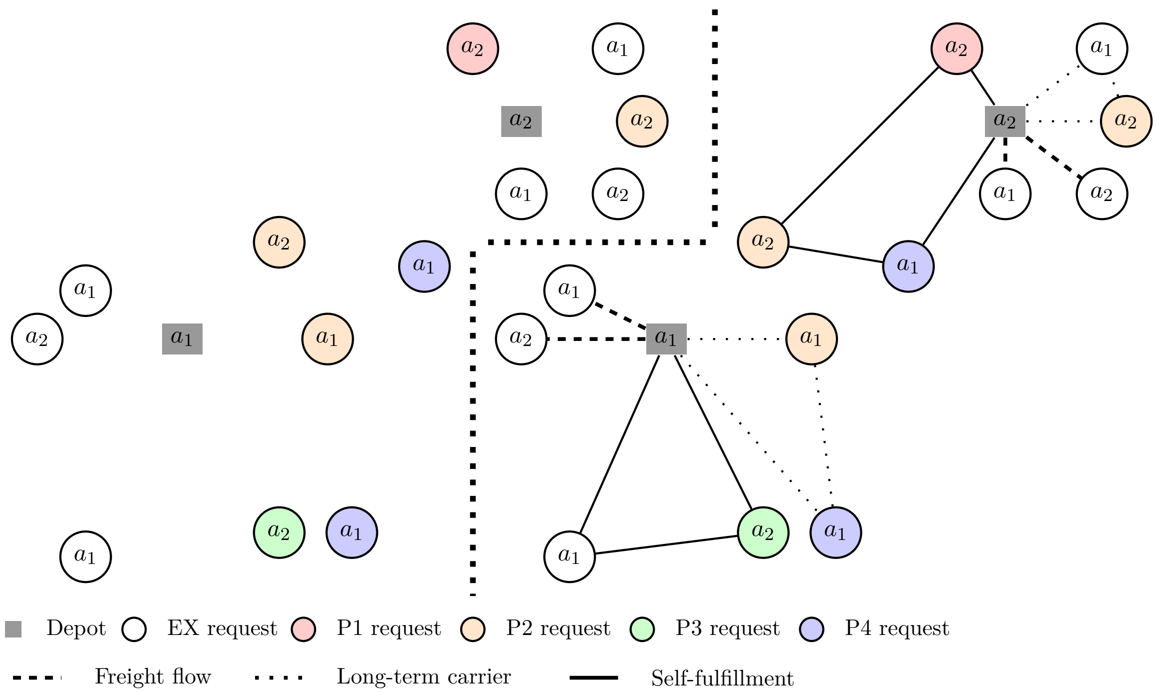


***Figure 9.1.:*** *CIOTP problem with exchangeable and mandatory requests*

Both freight forwarders are in charge of one private vehicle and one rented vehicle paid on mode DB or RB for fulfilling these requests. Each vehicle type has a limited capacity and different cost rates. Furthermore, both freight forwarders can employ a CC for fulfilling a request. The goal is to identify the transportation plan which minimizes the individual transportation costs of both freight forwarders by ensuring that the mandatory requests are fulfilled by their corresponding fulfillment mode. In Figure 9.1, the TPP is presented on the left side, while a solution is given on the right side of this figure. As can be seen, the TPP is getting more complex by considering four types of mandatory requests. Thereby, each request type has to be fulfilled by a suitable fulfillment mode corresponding to the attached service. For example, a $P3$ request can be fulfilled by the private fleet of any freight forwarder, while a $P4$ request can be fulfilled by any vehicle of the coalition.

The CGB-HER is applied and extended either by the strict generation or strict composition procedure for solving the CIOTPP-FL. In general, the solution strategies work as described in Section 6.1. In the following, it is described, which problem is solved by the subproblem or master problem of the CGB-HER in terms of the different solution strategies.

In a CIOTPP, $m$ freight forwarders align their individual transportation plans by exchanging requests with each other within a horizontal coalition. On a single decision level, each freight forwarder $c \in F$ faces an IOTPP, where a request can be fulfilled by self-fulfillment or subcontracting. The request exchange represents a group decision problem, where each freight forwarder $c \in F$ offers his or her entire request portfolio $P_c$ for exchange and receives a new portfolio $P'_c$ after the allocation process is completed. Each freight forwarder $c \in F$ solves his or her IOTPP based on the received request portfolio. Then, on a group decision level, it is observed to reduce the individual transportation costs and to fulfill each request corresponding to the attached service. Obviously, the CIOTPP has to be extended in order to be suitable for the consideration of forwarding limitations. By considering forwarding limitations, some fulfillment modes are prohibited for certain requests. In case of the collaborative perspective, the mathematical formulation with objective function (5.35) and the constraints (5.36)–(5.37) do not have to be changed in terms of both solution strategies.

**Strict generation procedure:** The strict generation procedure solves the CIOTPP-FL as subproblem by forwarding just feasible vehicle routes to the master problem, while the master problem can be solved as CIOTPP. As in the previous chapters, it is recommended to separate the set of requests $P$ into disjoint sets: $EX$ requests, $P1$ requests, $P2$ requests, $P3$ requests, and $P4$ requests, i.e., $P = EX \cup P1 \cup P2 \cup P3 \cup P4$. Thereby, each freight forwarder $c \in F$ keeps his or her set of $P1$ and $P2$ requests private. The individual request portfolio of freight forwarder $c$ contains the request sets $EX'_c \cup P1_c \cup P2_c \cup P3'_c \cup P4'_c$. In this case, $EX'_c, P3'_c$, and $P4'_c$ request sets represent the individual request sets after the request exchange. Based on this modification and by replacing constraints (5.17) with the following constraints, it is possible to generate the CIOTPP-FL:

$$\sum_{k \in K_c} \sum_{j \in V_c} x_{ijk} + y_i^{CC} = 1\,, \qquad \forall i \in EX'_c, \tag{9.1}$$

$$\sum_{k \in K_c} \sum_{j \in V_c} x_{ijk} = 1\,, \qquad \forall i \in P2_c, \forall i \in P4'_c, \tag{9.2}$$

$$\sum_{k \in K_c^1} \sum_{j \in V_c} x_{ijk} = 1\,, \qquad \forall i \in P1_c, \forall i \in P3'_c. \tag{9.3}$$

The forwarding limitation constraints are (9.1)–(9.3). Constraints (9.1) ensure that either a private vehicle or a rented vehicle or a CC is used for fulfilling a request in $EX'_c$. For a request in $P2_c$ or $P4'_c$, the private fleet and rented fleet can be used (constraints (9.2)), while a request in $P1_c$ or $P3'_c$ can only be served by the private fleet (constraints (9.3)). In case of the collaborative perspective, the set of offered requests has to be modified as mentioned in order that the individual $P1_c$ and $P2_c$ request sets are not offered for exchange. In summary, the strict generation procedure solves the following subproblem and SPP master problem.

- Subproblem: The objective function (5.16), constraints (5.18)–(5.34), and the forwarding limitation constraints (9.1)–(9.3) are considered.
- SPP master problem: The objective function (5.38) and constraints (5.39)–(5.41) are considered.

**Strict composition procedure** The strict composition procedure mainly solves the CIOTPP as subproblem and the CIOTPP-FL as master problem. Thus, feasible solutions for the CIOTPP-FL are generated by using the dual values of the master problem. As known, the strict generation procedure is applied in the first round of the iterative procedure of the CGB-HER.

For handling mandatory requests by the master problem, the set of offered requests contains all requests and is separated into five disjoint sets $P'_c = EX'_c \cup P1_c \cup P2_c \cup P3'_c \cup P4'_c$. Five parameters are added in terms of the master problem related to the CIOTPP-FL. On the one hand, the parameters $b_i^{u1}$ and $b_i^{o1}$ indicate the upper or lower bound of $P1_c$, $P2_c$, and $P4'_c$ requests in terms of the applicable fulfillment mode, respectively. On the other hand, the parameters $b_{ic}^{u2}$ and $b_{ic}^{o2}$ indicate the upper or lower bound of $P3'_c$ requests in terms of the applicable fulfillment mode for each freight forwarder $c \in F$. Furthermore, the parameter $b_r^M$ is introduced, which contains the information about the number of the vehicle. Thereby, it is always ensured that the set of vehicles $K$ is ordered in an ascending order corresponding to the vehicle fleets of the freight forwarders. This means that all vehicles of freight forwarder one are listed, which are followed by the vehicles of freight forwarder two and so on. It is also ensured that the set of individual vehicles starts by private vehicles and ends by rented vehicles. To handle mandatory requests, the CIOTPP has to be extended by replacing constraints (8.4) and (8.5) of the master problem in Section 8.1 by the following ones:

$$\sum_{r \in R_c} a_{ir} u_r = 1\,, \qquad \forall i \in P1_c \cup P2_c \cup P3'_c \cup P4'_c, \tag{9.4}$$

$$b_i^{u1} \leq \sum_{r \in R_c} a_{ir} u_r b_r^M \leq b_i^{o1}\,, \qquad \forall i \in P1_c \cup P2_c \cup P4'_c, \tag{9.5}$$

$$b_{ic}^{u2} \leq \sum_{r \in R_c} a_{ir} u_r b_r^M \leq b_{ic}^{o2}\,, \qquad \forall i \in P3'_c, \forall c \in F. \tag{9.6}$$

The goal of the master problem is to minimize the transportation costs defined by objective function (8.3) and to fulfill requests corresponding to their attached service.

Constraints (9.4) define that each mandatory request is served by exactly one vehicle route. Thereby, it is not ensured that a specific vehicle is used for a mandatory request. That is why constraints (9.5) are proposed, which observe the applicable range of vehicles for $P1$ requests, $P2$ requests, and $P4$ requests, while constraints (9.6) do that for $P3$ requests. For example, a scenario with one freight forwarder is considered, who has three private vehicles, three rented vehicles on mode DB and RB. Then, the applicable range for a $P2$ request would be one ($b_i^{u1}$) to nine ($b_i^{o1}$). Otherwise, in a scenario with two freight forwarders, where both freight forwarders (numbered by 1 and 2) are in charge of three vehicles per fulfillment mode (self-fulfillment, mode DB, and mode RB), a $P3$ request has to be served by a vehicle with the number one ($b_{i1}^{u2}$) to three ($b_{i1}^{o2}$) or ten ($b_{i2}^{u2}$) to twelve ($b_{i2}^{o2}$). In summary, the strict composition procedure solves the following subproblem and SPP master problem.

- Subproblem: The objective function (5.16) and constraints (5.17)–(5.34) are considered.
- SPP master problem: The objective function (8.3), constraints (8.6)–(8.8), and the forwarding limitation constraints (9.5)–(9.6) are considered.

In terms of the strict generation procedure, the fulfillment of $P1$ and $P2$ requests by freight forwarders as well as the fulfillment of $P1$ and $P3$ requests by long-term carriers are prohibited by infeasible insertion positions during the ALNS. To ensure that vehicle routes are selected in the strict composition procedure, which are feasible for the considered CIOTPP-FL, the master problem is extended by new constraints. These constraints observe the applied fulfillment modes for mandatory requests. Therefore, the vehicles are numbered in an ascending order and for each mandatory request a certain range is determined. To handle multiple types of mandatory requests simultaneously, the described solution approach can mainly be used as described. It is just necessary to introduce new parameters in the CGB-HER, which define the request types by a specific number for determining the applicable vehicle sets.

## 9.2. Computational studies

A CIOTPP-FL is solved in this section. Once it is solved by considering each type of mandatory request separately and once by considering all of them simultaneously. In both cases, the impact of mandatory requests is calculated.

In Subsection 9.2.1, the existing CIOTPP instances of Wang et al. (2014) are used. Wang et al. (2014) present 24 CIOTPP instances, where two to five IOTPP instances with the same location structure ($R1$, $C1$, and $RC1$) are combined to one CIOTPP instance with modified coordinates of the nodes. To consider the characteristic of the CIOTPP-FL, it is necessary to define different request types. Here, instances with 5 %, 10 %, and 15 % mandatory requests are considered. Higher ratios are skipped because they lead to infeasible solutions, which cannot be used for an evaluation. 15 samples are generated for each ratio and instance, which are solved once for $P1$, $P2$, $P3$, and $P4$ requests.

Subsection 9.2.2 uses the instances of Subsection 9.2.1. To be precise, the instances with 15 % mandatory requests and 85 % exchangeable requests are considered. This instance set is selected because the other instance sets consider only few mandatory requests. Regarding these 15 % mandatory requests, eleven different combinations are analyzed: ($P1$, $P2$), ($P1$, $P3$), ($P1$, $P4$), ($P2$, $P3$), ($P2$, $P4$), ($P3$, $P4$), ($P1$, $P2$, $P3$), ($P1$, $P2$, $P4$), ($P1$, $P3$, $P4$), ($P2$, $P3$, $P4$), and ($P1$, $P2$, $P3$, $P4$). For example, the combination ($P1$, $P2$) means that 50 % of the mandatory requests are $P1$ requests and the remaining ones are $P2$ requests. In the existing instances, the type of mandatory request is not defined. That it is why the number of mandatory requests in an instance is divided by the number of request types in a combination. Then, each request type receives the same number of requests in an ascending order based on the existing request order. In case that an equal distribution is not available, the last request type within a combination receives the remaining mandatory requests. 15 samples are generated for each instance and combination.

As an evaluation criterion, the percentaged increase of costs between the CIOTPP-FL solution computed by the CGB-HER with the best solution strategy and the best-known CIOTPP solution values as presented by Wang et al. (2014) is used. It means that the same evaluation criterion as in the previous chapters is applied, which is defined by the variable $\mathcal{G}_a$.

The CGB-heuristic is implemented in C++ (Visual Studio 2012) and the computational studies are executed on a Windows 7 PC with Intel Core i7-2600 processor (3.4 GHz and 16 GB of main memory). The dual values are computed by using CPLEX (version 12.5.1). The same parameter setting is used as in Chapter 6.

### 9.2.1. Impact of one mandatory request type

This subsection focuses on the increase of transportation costs for the CIOTPP-FL, where different types of mandatory requests are not mixed with each other. First, it has to be identified which solution strategy should be used for the CIOTPP-FL. Second, the financial impact has to be determined.

All CIOTPP-FL instances with four freight forwarders including a ratio of 10 % mandatory requests are solved once by the strict generation and once by the strict composition procedure in order to identify the best solution strategy. As in the previous chapters, both solution strategies perform similar in terms of the computational effort, while the results regarding the solution quality depend on the mandatory request type. It is observed that the strict generation procedure is preferable for $P1$ and $P3$ requests, while the strict composition procedure is preferable for $P2$ and $P4$ requests. It is assumed that the strict generation procedure has performance issues in terms of identifying promising vehicle routes in case that the solution space is widely extended by additional fulfillment modes. This is the case when the long-term carrier represents a feasible fulfillment mode.

In a second study, the impact of mandatory requests is analyzed. Thereby, the CIOTPP-FL is solved by the CGB-HER combined with the best solution strategy for each mandatory request type. The increase of transportation costs per mandatory request as well as the best CIOTPP solution values are presented in Table 9.1.

***Table 9.1.:*** *Mean percentaged increase of costs per mandatory request in CIOTPP-FL, cf. Ziebuhr and Kopfer (2016b)*

| instance | | | best | $P1$ requests | | | $P2$ requests | | | $P3$ requests | | | $P4$ requests | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| id | m | N | CIOTPP | 5 % | 10 % | 15 % | 5 % | 10 % | 15 % | 5 % | 10 % | 15 % | 5 % | 10 % | 15 % |
| c101 | 2 | 105 | 5,348 | 0.97 | 1.00 | 1.15 | 0.37 | 0.30 | 0.29 | 0.83 | 0.86 | 1.08 | 0.03 | 0.02 | 0.06 |
| c102 | 2 | 106 | 5,340 | 1.08 | 0.86 | 0.99 | 0.31 | 0.22 | 0.18 | 0.93 | 0.75 | 0.84 | 0.00 | 0.00 | 0.00 |
| c103 | 3 | 159 | 7,910 | 0.61 | 0.60 | 0.63 | 0.23 | 0.26 | 0.23 | 0.50 | 0.46 | 0.57 | 0.07 | 0.06 | 0.06 |
| c104 | 3 | 159 | 7,335 | 0.60 | 0.67 | 0.72 | 0.27 | 0.24 | 0.20 | 0.44 | 0.55 | 0.68 | 0.02 | 0.01 | 0.02 |
| c105 | 4 | 212 | 9,912 | 0.37 | 0.38 | 0.48 | 0.19 | 0.14 | 0.20 | 0.28 | 0.31 | 0.45 | 0.03 | 0.03 | 0.03 |
| c106 | 4 | 211 | 9,721 | 0.56 | 0.56 | 0.58 | 0.32 | 0.30 | 0.29 | 0.47 | 0.43 | 0.56 | 0.05 | 0.04 | 0.03 |
| c107 | 5 | 264 | 12,296 | 0.38 | 0.35 | 0.47 | 0.21 | 0.16 | 0.17 | 0.32 | 0.30 | 0.44 | 0.02 | 0.01 | 0.02 |
| c108 | 5 | 264 | 12,386 | 0.33 | 0.39 | 0.47 | 0.16 | 0.17 | 0.19 | 0.31 | 0.32 | 0.47 | 0.00 | 0.00 | 0.01 |
| r101 | 2 | 104 | 7,052 | 0.34 | 0.41 | 0.48 | 0.25 | 0.23 | 0.21 | 0.27 | 0.31 | 0.39 | 0.05 | 0.08 | 0.06 |
| r102 | 2 | 104 | 7,504 | 0.53 | 0.60 | 0.80 | 0.10 | 0.13 | 0.12 | 0.51 | 0.59 | 0.75 | 0.03 | 0.03 | 0.03 |
| r103 | 3 | 160 | 10,678 | 0.32 | 0.35 | 0.40 | 0.19 | 0.18 | 0.20 | 0.24 | 0.25 | 0.35 | 0.06 | 0.06 | 0.05 |
| r104 | 3 | 154 | 9,854 | 0.33 | 0.39 | 0.48 | 0.14 | 0.16 | 0.14 | 0.27 | 0.30 | 0.47 | 0.02 | 0.01 | 0.03 |
| r105 | 4 | 208 | 13,333 | 0.31 | 0.31 | 0.41 | 0.18 | 0.17 | 0.16 | 0.21 | 0.24 | 0.43 | 0.06 | 0.05 | 0.04 |
| r106 | 4 | 215 | 15,508 | 0.23 | 0.23 | 0.27 | 0.14 | 0.16 | 0.14 | 0.15 | 0.18 | 0.24 | 0.02 | 0.05 | 0.03 |
| r107 | 5 | 265 | 16,222 | 0.20 | 0.24 | 0.27 | 0.16 | 0.15 | 0.18 | 0.14 | 0.19 | 0.26 | 0.00 | 0.04 | 0.03 |
| r108 | 5 | 262 | 16,908 | 0.24 | 0.24 | 0.27 | 0.21 | 0.15 | 0.11 | 0.17 | 0.18 | 0.30 | 0.12 | 0.08 | 0.07 |
| rc101 | 2 | 106 | 7,415 | 0.52 | 0.46 | 0.61 | 0.35 | 0.27 | 0.25 | 0.37 | 0.39 | 0.55 | 0.19 | 0.18 | 0.13 |
| rc102 | 2 | 107 | 8,058 | 0.54 | 0.46 | 0.62 | 0.35 | 0.26 | 0.24 | 0.48 | 0.41 | 0.58 | 0.26 | 0.18 | 0.14 |
| rc103 | 3 | 160 | 10,801 | 0.35 | 0.37 | 0.40 | 0.26 | 0.25 | 0.23 | 0.20 | 0.24 | 0.40 | 0.07 | 0.07 | 0.06 |
| rc104 | 3 | 161 | 11,485 | 0.38 | 0.40 | 0.45 | 0.25 | 0.27 | 0.26 | 0.22 | 0.26 | 0.45 | 0.04 | 0.05 | 0.06 |
| rc105 | 4 | 211 | 13,783 | 0.32 | 0.36 | 0.42 | 0.26 | 0.26 | 0.24 | 0.18 | 0.28 | 0.40 | 0.08 | 0.06 | 0.06 |
| rc106 | 4 | 213 | 14,711 | 0.29 | 0.31 | 0.37 | 0.24 | 0.22 | 0.20 | 0.18 | 0.23 | 0.36 | 0.06 | 0.05 | 0.06 |
| rc107 | 5 | 265 | 17,231 | 0.29 | 0.28 | 0.35 | 0.25 | 0.22 | 0.21 | 0.17 | 0.25 | 0.34 | 0.06 | 0.05 | 0.03 |
| rc108 | 5 | 266 | 18,339 | 0.25 | 0.27 | 0.31 | 0.22 | 0.21 | 0.21 | 0.17 | 0.21 | 0.30 | 0.07 | 0.06 | 0.06 |
| mean | | | 11,214 | 0.43 | 0.44 | 0.52 | 0.23 | 0.21 | 0.20 | 0.33 | 0.35 | 0.49 | 0.06 | 0.05 | 0.05 |

Several findings can be derived by observing this table. As already identified, the increase of costs depends on the location structure, ratio of mandatory requests, mandatory request type, and number of freight forwarders. First, it is observed that on average $P1$ requests lead to the highest additional costs, $P3$ requests the second highest, $P2$ requests the third highest and $P4$ requests lead to the lowest additional costs. Concerning the $P1$ and $P4$ requests, the result is expected due to the different sets of applicable external resources, while the result is not obvious in case of $P2$ and $P3$ requests. It is worth mentioning that in scenarios with low ratios and randomly clustered location structures $P3$ requests have often lower additional costs than $P2$ requests. In such scenarios, the benefit of using the private fleet of a freight forwarder is higher than the benefit of using long-term carriers. A third observation is that random location structures lead to the lowest additional costs for $P1$, $P2$, and $P3$ requests, while clustered structures are preferable for $P4$ requests. By increasing the number of freight forwarders, the mean impact of mandatory requests can be reduced. Another observation is that higher ratios lead to different results depending of the type of mandatory requests. While $P1$ and $P3$ requests get higher additional costs, the impact on $P2$ and $P4$ requests can be reduced by these higher ratios. Thereby, it is worth mentioning that ratios higher than 15 % are analyzed. Therefore, it is observed that with higher ratios, it is difficult to identify feasible solutions for the CIOTPP with $P1$ or $P3$ requests. That is why higher ratios are skipped. Actually, in terms of a ratio of 15 % it is observed that on average with the CGB-HER 10 % of the CIOTPP instances with $P1$ requests cannot be solved while 7 % of the CIOTPP instances with $P3$ requests cannot be solved. However, all considered CIOTPP instances with $P2$ or $P3$ requests can be solved.

### 9.2.2. Impact of multiple mandatory request types

This subsection focuses on the increase of transportation costs for a CIOTPP-FL, where different types of mandatory requests are considered simultaneously. First, it has to be identified, which solution strategy is suitable for this CIOTPP-FL. Second, the financial impact has to be determined.

In a first study, the strict generation procedure is compared with the strict composition procedure for all mandatory request combinations and instances with three freight forwarders. In this case it is known that the strict generation procedure is preferable for $P1$ and $P3$ requests, while the strict composition procedure is preferable for $P2$ and $P4$ requests corresponding to Subsection 9.2.1. Corresponding to this observation, it is assumed that the strict generation procedure is preferable when most of the mandatory requests are $P1$ and $P3$ requests within a combination, while the strict composition procedure is preferable within combinations with many $P2$ and $P4$ requests. In a computational study, this assumption can often be verified by identifying that every time when $P3$ requests are considered the strict generation procedure leads to better results.

In a second study, the increase of transportation costs is analyzed by considering different combinations of mandatory requests. Thereby, the CIOTPP-FL is solved by the CGB-HER combined with the best strategy for each mandatory request combination. The mean percentaged increase of costs per mandatory request as well as the best CIOTPP solution values are presented in Table 9.2.

***Table 9.2.:*** *Mean percentaged increase of costs per mandatory request combination in CIOTPP-FL, cf. Ziebuhr and Kopfer (2017)*

| instance | best | combinations (increase of costs in %) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| id | CIOTPP | (1,2) | (1,3) | (1,4) | (2,3) | (2,4) | (3,4) | (1,2,3) | (1,2,4) | (1,3,4) | (2,3,4) | (1,2,3,4) |
| c101 | 5,348 | 0.72 | 1.11 | 0.60 | 0.69 | 0.18 | 0.57 | 0.84 | 0.50 | 0.76 | 0.48 | 0.64 |
| c102 | 5,340 | 0.59 | 0.91 | 0.50 | 0.51 | 0.09 | 0.42 | 0.67 | 0.39 | 0.61 | 0.34 | 0.50 |
| c103 | 7,910 | 0.43 | 0.60 | 0.34 | 0.40 | 0.14 | 0.31 | 0.48 | 0.30 | 0.42 | 0.29 | 0.37 |
| c104 | 7,335 | 0.46 | 0.70 | 0.37 | 0.44 | 0.11 | 0.35 | 0.53 | 0.31 | 0.47 | 0.30 | 0.40 |
| c105 | 9,912 | 0.34 | 0.47 | 0.26 | 0.32 | 0.11 | 0.24 | 0.38 | 0.24 | 0.32 | 0.22 | 0.29 |
| c106 | 9,721 | 0.43 | 0.57 | 0.30 | 0.42 | 0.16 | 0.30 | 0.47 | 0.30 | 0.39 | 0.29 | 0.36 |
| c107 | 12,296 | 0.32 | 0.45 | 0.24 | 0.30 | 0.09 | 0.23 | 0.36 | 0.22 | 0.31 | 0.21 | 0.27 |
| c108 | 12,386 | 0.33 | 0.47 | 0.24 | 0.33 | 0.10 | 0.24 | 0.38 | 0.22 | 0.31 | 0.22 | 0.28 |
| r101 | 7,052 | 0.35 | 0.44 | 0.27 | 0.30 | 0.14 | 0.23 | 0.36 | 0.25 | 0.31 | 0.22 | 0.29 |
| r102 | 7,504 | 0.46 | 0.78 | 0.41 | 0.43 | 0.07 | 0.39 | 0.56 | 0.31 | 0.53 | 0.30 | 0.42 |
| r103 | 10,678 | 0.30 | 0.38 | 0.22 | 0.27 | 0.12 | 0.20 | 0.32 | 0.21 | 0.27 | 0.20 | 0.25 |
| r104 | 9,854 | 0.31 | 0.47 | 0.25 | 0.31 | 0.09 | 0.25 | 0.36 | 0.22 | 0.33 | 0.22 | 0.28 |
| r105 | 13,333 | 0.28 | 0.42 | 0.22 | 0.29 | 0.10 | 0.23 | 0.33 | 0.20 | 0.29 | 0.21 | 0.26 |
| r106 | 15,508 | 0.20 | 0.25 | 0.15 | 0.19 | 0.08 | 0.14 | 0.21 | 0.14 | 0.18 | 0.14 | 0.17 |
| r107 | 16,222 | 0.23 | 0.27 | 0.15 | 0.22 | 0.10 | 0.14 | 0.24 | 0.16 | 0.19 | 0.16 | 0.18 |
| r108 | 16,908 | 0.19 | 0.28 | 0.17 | 0.20 | 0.09 | 0.18 | 0.23 | 0.15 | 0.21 | 0.16 | 0.19 |
| rc101 | 7,415 | 0.43 | 0.58 | 0.37 | 0.40 | 0.19 | 0.34 | 0.47 | 0.33 | 0.43 | 0.31 | 0.39 |
| rc102 | 8,058 | 0.43 | 0.60 | 0.38 | 0.41 | 0.19 | 0.36 | 0.48 | 0.33 | 0.45 | 0.32 | 0.40 |
| rc103 | 10,801 | 0.31 | 0.40 | 0.23 | 0.31 | 0.15 | 0.23 | 0.34 | 0.23 | 0.29 | 0.23 | 0.27 |
| rc104 | 11,485 | 0.35 | 0.45 | 0.25 | 0.36 | 0.16 | 0.25 | 0.39 | 0.26 | 0.32 | 0.26 | 0.30 |
| rc105 | 13,783 | 0.33 | 0.41 | 0.24 | 0.32 | 0.15 | 0.23 | 0.35 | 0.24 | 0.29 | 0.23 | 0.28 |
| rc106 | 14,711 | 0.28 | 0.36 | 0.21 | 0.28 | 0.13 | 0.21 | 0.31 | 0.21 | 0.26 | 0.21 | 0.25 |
| rc107 | 17,231 | 0.28 | 0.35 | 0.19 | 0.28 | 0.12 | 0.19 | 0.30 | 0.20 | 0.24 | 0.19 | 0.23 |
| rc108 | 18,339 | 0.26 | 0.30 | 0.18 | 0.26 | 0.14 | 0.18 | 0.27 | 0.19 | 0.22 | 0.19 | 0.22 |

A mean percentaged increase of costs of 0.72 % means that one artificial mandatory within a certain request combination leads to this percentage of additional costs. To evaluate the results of Table 9.2, it is necessary to consider the findings of Subsection 9.2.1. There it is observed that on average $P1$ requests lead to the highest, $P3$ requests to the second highest, $P2$ requests to the third highest, and $P4$ requests to the fourth highest additional costs. Corresponding to this finding, the following order in terms of additional costs is expected for combinations with two request types: ($P1$, $P3$), ($P1$, $P2$), ($P2$, $P3$), ($P1$, $P4$), ($P3$, $P4$), and ($P2$, $P4$) as well as for combinations with three request types: ($P1$, $P2$, $P3$), ($P1$, $P3$, $P4$), ($P1$, $P2$, $P4$), and ($P2$, $P3$, $P4$) (descending orders). This assumption is verified in this computational study. Furthermore, it is also obvious that the highest additional costs over all combinations can always be observed for the combination ($P1$, $P3$), while the lowest ones can be observed for the combination ($P2$, $P4$). The remaining combinations differ slightly more regarding their additional costs depending on the location structure and the number of freight forwarders. It is also observed that there is no significant cost reduction by considering different types of mandatory requests simultaneously due to the fact that the identified figures are just slightly lower (on average about 7 %) than the aggregated figures of Subsection 9.2.1. It is assumed that the procedure of distributing different mandatory requests within the instance generation is mainly responsible for this observation.

This chapter introduces a CIOTPP-FL, where four different mandatory request types are analyzed in terms of their increase of costs compared to a solution without forwarding limitations. In previous chapters, the topic of mandatory requests is analyzed for a TPP with either self-fulfillment and subcontracting or self-fulfillment and collaboration as fulfillment modes. Here, an extensive approach is presented where self-fulfillment, subcontracting, and collaboration are applicable as fulfillment modes. It is also introduced how mandatory request combinations can be solved and evaluated. The CGB-HER with two different strategies for handling mandatory requests is applied as solution approach. In a computational study, it is identified that the strict generation procedure is preferable for $P1$ and $P3$ requests and the strict composition procedure is preferable for $P2$ and $P4$ requests, while the strict composition procedure is always preferred in case of mandatory request combinations without $P3$ requests. Furthermore, the impact of mandatory requests is analyzed. Several findings can be derived. For example, random location structures are preferable in terms of additional costs and $P1$ requests lead to the highest additional costs. It is also observed that mandatory request combinations with $P1$ and $P3$ requests lead to the highest additional costs.

# 10. Conclusions and Outlook

Lot-sizing problems of manufacturers and transportation planning problems (i.e., TPPs) of freight forwarders are presented and analyzed in this thesis. These problems represent common and crucial planning tasks in supply chain management. Due to high demand fluctuations and competitive markets, companies within supply chains use internal and external resources for the fulfillment of tasks. The thesis claims to contribute to the following topics: (1) introducing mandatory tasks for the DULR, IOTPP, CTPP, and CIOTPP as well as (2) presenting detailed computational studies that demonstrate how much the costs of companies increase due to mandatory tasks. Mandatory tasks are tasks, which have to be fulfilled by appointed resources due to contractual obligations. A lack of research is identified in terms of this topic. It is usually assumed in literature that a task can be fulfilled by any internal or external resources. The thesis gives an overview how these planning tasks with mandatory tasks can be solved by using operations research. Therefore, existing mathematical models and solution approaches have to be extended in order to take care of mandatory tasks. The thesis focuses on the determination of the impact of mandatory tasks based on extensive computational studies. The results of these studies can be used by production and transportation planners in order to optimize their production and transportation plans in case of the existence of mandatory tasks.

## 10.1. Summary

The thesis is divided into two separated parts. Part I gives an overview about lot-sizing problems with mandatory tasks, while Part II presents TPPs with mandatory tasks. A comprehensive approach was not possible due to different premisses in lot size planning and transportation planning.

Lot size planning belongs to the short-term production planning within supply chains, where the goal is is to identify promising lot sizes under the consideration of different cost rates. Different lot-sizing problems are known in the literature. The thesis focuses on uncapacitated MLLS problems, where final items (i.e., tasks of the final customers) are produced based on a multi-level product structure. Thereby, it is usually assumed in publications that items cannot be produced by external resources. An overview about lot size planning in production planning is presented in Chapter 2. It is identified that lot size planning represents one planning phase in material requirements planning (i.e., MRP) of supply chains. Chapter 2 focuses on single and group decision making lot-sizing problems. A known solution approach (i.e., NBM) is described to solve the presented lot-sizing problems (i.e., MLULSP and DMLULSP). The NBM is a heuristic, which uses an indirect problem representation and a simulated annealing (i.e., SA) as an acceptance criterion.

The mentioned NBM is rebuilt from scratch and extended by a so called PWR procedure (i.e., NBM-PWR) in Chapter 3. This PWR procedure allows to overcome dead locks during the local search more easily by using reset points. The assumption is verified by a benchmark study, where the NBM-PWR is compared with six state of the art heuristics. Thereby, the NBM-PWR is identified as one of the best heuristics for the DMLULSP in case of small and medium instances.

Due to the development of a promising solution approach in Chapter 3, it is possible to extend the DMULSP by the option of using external resources for items. In this case, the application of external resources means that a second manufacturer within the coalition can produce a certain item (i.e., exchangeable item) as well. Chapter 4 explains the development of a DMLULSP to a DMLULSP with exchangeable items (i.e., DULR) and the development to a DMULSP with exchangeable and mandatory items (i.e., DULR-PL). External resources were not considered in previous publications regarding these kind of lot-sizing problems. In Section 4.1, a DULR with sole-item production is proposed, where manufacturers compete with each other regarding the whole production volume of certain items. The problem is solved by an extended NBM-PWR (i.e., NBM-PWR-1). The solution approach is extended by an assignment and side payment procedure. The assignment procedure is used for the production of exchangeable items, while side payments are used to enhance a stable coalition. In computational studies, the efficiency of the NBM-PWR-1 is verified based on 272 instances. Then, the mathematical model of Section 4.1 is extended by new cost parameters in Section 4.2. A multiple-item production can be realized due to this procedure, where the production volume of an item can be produced by more than one manufacturer. The NBM-PWR is modified (i.e., NBM-2) by an updated assignment and side payment procedure as well as by eliminating the PWR procedure. The PWR procedure is obsolete for this lot-sizing problem due to the updated side payment procedure. Besides the formulation of a DULR with multiple item-production, the goal of Section 4.1 is to present three different ways of computing side payments and to analyze their impact on the solution quality and computational effort by extensive computational studies. Thus, it is identified that the NBM-2 performs better in case that side payments are applied. In Section 4.3, a DULR-PL is presented, where some items have to be produced by an appointed agent of the coalition. These items are denoted as mandatory items. The NBM-3 is applied to solve this problem, which extends the NBM-2 by a new assignment procedure for identifying a suitable shared production among the agents of a coalition and a procedure for handling mandatory items. In a benchmark study, the NBM-3 is identified as a suitable solution approach for a DULR with multiple item-production. Based on this result, the NBM-3 is used for solving the DULR-PL, where several findings could be derived. The computational studies indicate that mandatory items always lead to higher production costs and that items on a higher level of the product structure (e.g., final items) have a higher impact on the production costs than items on a lower level of the product structure (e.g., raw materials).

Transportation planning is responsible for determining transportation plans for the distribution of goods. Different decision problems have to be solved. For example, it has to be determined which fulfillment should be selected for which transportation request. This thesis focuses on TPPs with pickup and delivery locations, where goods have to be picked at a location and delivered to another location. Internal and external resources are available to transport these goods. In Chapter 5, an overview about transportation planning is given. It is identified that transportation planning represents a planning task in short-term distribution planning of a supply chain. In this scenario, different TPPs with up to three fulfillment modes might be considered. Common fulfillment modes for a request are: self-fulfillment, subcontracting, and collaboration. Thereby, it is usually assumed that a request can be fulfilled by any fulfillment mode. In contrast to lot size planning, there are some publications in transportation planning which consider requests with forwarding limitations. These requests, which have to be fulfilled by appointed resources, are denoted as mandatory requests. A literature overview regarding mandatory requests in TPPs is given in Chapter 5. There it can be seen for example that none of the existing publications consider different mandatory request types, because usually they just consider subcontracting or collaboration as external resources and not both of them. This thesis considers up to four mandatory request types regarding forwarding limitations (e.g., $P1$, $P2$, $P3$, and $P4$ requests) in different TPPs. For example, a $P3$ request can only be fulfilled by a private vehicle of any coalition member. At the end of Chapter 5, the basic solution approach (i.e., CGB-heuristic) of this part is explained. The applied CGB-heuristic uses a column generation approach, where the subproblem is solved by an ALNS and the master problem is solved by a commercial solver.

Chapter 6 describes how the CGB-heuristic has to be extended in order to solve an IOTPP-FL with two types of mandatory requests ($P1$ and $P2$ requests). In contrast to Part I with the NBM, the CGB-heuristic is not rebuilt from scratch. An existing code of Wang and Kopfer (2015) was available and modified for this thesis by different solution strategies for handling mandatory requests. Three solution strategies are presented: strict generation, strict composition, and repair procedure. All of these solution strategies append additional restrictions resulting from forwarding limitations to the CGB-heuristic. Three ways of installing the additional restrictions in the CGB-heuristic are compared. In Chapter 6, two computational studies are presented. First, different basic solution approaches (i.e., ALNS, CGB-HER, and CGB-HOM) are compared with each other regarding their solution quality and their computational effort for the IOTPP-FL. Thereby, it is identified that the CGB-HER outperforms the other solution approaches. Second, the CGB-HER with the strict generation procedure is used for a benchmark study, where it is demonstrated that the CGB-HER with the strict generation procedure outperforms the existing heuristic of Schönberger (2005) for a different TPP with exchangeable and mandatory requests significantly. There a freight forwarder can choose between self-fulfillment and employing a CC on a sport market for fulfilling a request.

In Chapter 6, two issues are not addressed. First, two of the three solution strategies for handling mandatory requests are not evaluated. Second, detailed computational studies regarding the impact of mandatory requests for the IOTPP-FL are missing. In Chapter 7, the IOTPP-FL is considered and the CGB-HER is alternatively applied with an end-of-pipe procedure or one of the two integrated procedures for handling mandatory requests. The results of the computational studies demonstrate that integrating the rules for respecting the additional restrictions to the master problem of the solution approach is the best of the three solution strategies. Installing approaches for integrating these restrictions during the subproblem is the second best solution strategy, while enforcing the compliance to these restrictions at the end (after the entire process of the CGB-HER) is by far the worst alternative. Furthermore, extensive computational studies are presented, where the impact of the fleet size and the location structure are analyzed in terms of the IOTPP-FL. Thereby, it is verified that mandatory requests always lead to additional costs in TPPs because mandatory requests limit the fulfillment options for a requests, which results in additional costs in terms of the considered scenario with limited resources. Besides a calculated pattern, which can be used as a decision instrument for transportation planning, several findings are derived based on these computational studies. For example, two preferable scenarios are identified. On the one hand, random and randomly clustered location structures are preferable, when the available fleet size for mandatory requests is less than or equal to 70 % of the total vehicle fleet. On the other hand, clustered location structures are preferable, when the available fleet size is about 100 % of the total vehicle fleet. The available fleet contains all vehicles which can be used for fulfilling a specific mandatory request type.

Chapter 8 contributes by introducing a CTPP-FL with two types of mandatory requests ($P1$ and $P3$ requests), which is the first problem formulation in the literature where mandatory requests are considered in a CTP scenario. The mentioned CGB-HER with two solution strategies is applied. The repair procedure is omitted due to performance issues. In a computational study, it is identified that the best solution strategy depends on the request type. It is observed that the CTPP-FL with $P1$ requests should be solved by the strict generation procedure, while the strict composition procedure should be used for the CTPP-FL with $P3$ requests. In terms of the impact of mandatory requests, several findings could be derived like that $P1$ requests have a much higher impact on the additional costs than $P3$ requests and that the lowest additional costs occur for random location structures.

Finally, Chapter 9 introduces a CIOTPP-FL with four types of mandatory requests ($P1$, $P2$, $P3$, and $P4$ requests). In previous chapters, the topic of mandatory requests is analyzed for TPPs either with self-fulfillment and subcontracting or with self-fulfillment and collaboration as fulfillment modes. An extensive approach is presented, where self-fulfillment, subcontracting, and collaboration are applicable as fulfillment modes. The same solution approach as presented in Chapter 8 is applied, which is slightly modified. Two test settings are investigated in Chapter 9. First, a computational study is presented, where the impact of each mandatory request type is investigated separately for the CIOTTP-FL. Second, a computational study is presented, where the impact of mandatory request combinations (i.e., consideration of mixed mandatory request types) is investigated for the CIOTTP-FL. It is identified that the strict generation procedure is preferable for the CIOTPP-FL with $P1$ and $P3$ requests and for all combinations with $P3$ requests, while the strict composition procedure is preferable for the CIOTPP-FL with $P2$ and $P4$ requests and for all combinations without $P3$ requests. Furthermore, it is observed for the CIOTPP-FL that on average $P1$ requests lead to the highest additional costs, $P3$ requests the second highest, $P2$ requests the third highest and $P4$ requests lead to the lowest additional costs. Corresponding to this observation, it is verified that the highest additional costs for the CIOTPP-FL occur for combinations with $P1$ and $P3$ requests and the lowest ones for combinations with $P2$ and $P4$ requests.

One finding of the thesis is that mandatory items or requests lead to additional costs in production and transportation scenarios with limited resources. It is even possible that high ratios of mandatory items or requests lead to infeasible solutions.

## 10.2. Future research

This thesis contributes by analyzing the impact of mandatory tasks for common lot-sizing problems and TPPs. In both research fields, a lack of research was identified. This research gap could be reduced by introducing modified mathematical models and heuristic approaches, which are able to deal with mandatory tasks. Furthermore, extensive computational studies are presented. However, future research is still necessary.

In Part I, a collaborative uncapacitated lot-sizing problem with exchangeable and mandatory items is introduced. However, this is just one problem formulation based on new cost parameters, where at most two agents compete with each other regarding the production volume of an exchangeable item. Furthermore, the option of using subcontracting is still not addressed in this problem formulation. It is also proposed to develop solution approaches which are also able to solve larger instances due to the fact that the NBM just solves small and medium instances.

In Part II, different TPPs with exchangeable and mandatory requests are presented. Thereby, three common TPPs are extended by mandatory requests. In general, the topic represents a recent research field, where new mathematical models and solution approaches are developed by several departments during the last two years. Thereby, it is observed that a benchmark study between these existing solution approaches is still missing, because common instances are not available. This research gap has to be closed. Furthermore, it might be interesting to apply the existing solution approaches on real life data sets.

Obviously, it is also highly recommended to develop a comprehensive approach, where the impact of mandatory tasks is determined for a supply chain instead of just investigating it for certain planning parts of the supply chain. It was not possible to develop such an approach instantly due to missing foundations in both research fields. However, the thesis should help to develop such a comprehensive approach, where the impact of mandatory tasks can be analyzed.

# Bibliography


Afentakis, P. and B. Gavish (1986). Optimal lot-sizing algorithms for complex product structures. *Operations Research 34*(2), 237–249.

Afentakis, P., B. Gavish, and U. Karmarkar (1984). Computationally efficient optimal solutions to the lot-sizing problem in multistage assembly systems. *Management Science 30 (2)*, 222–239.

Agarwal, R. and O. Ergun (2010). Network design and allocation mechanisms for carrier alliances in liner shipping. *Operations Research 58*(6), 1726–1742.

Agarwal, Y. (1985). Vehicle routing with limited fleet and common carrier option. In *TIMS/ORSA Joint National Meeting.*

Applegate, D. L., R. E. Bixby, V. Chvatal, and W. J. Cook (2007). *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton, NJ, USA: Princeton University Press.

Arndt, H. (2008). *Supply Chain Management: Optimierung logistischer Prozesse* (4., aktualisierte und überarbeitete Auflage ed.). Wiesbaden: Gabler Verlag / GWV Fachverlage GmbH, Wiesbaden. Online-Ressource : v.: digital.

Baldacci, R., M. Battarra, and D. Vigo (2008). *Routing a Heterogeneous Fleet of Vehicles*, pp. 3–27. Boston, MA: Springer US.

Ball, M., B. Golden, A. Assad, and L. Bodin (1983). Planning for truck fleet size in the presence of a common-carrier option. *Decision Sciences 14*(1), 103–120.

Ballou, R. and M. Chowdhury (1980). An extended computer model for transport mode selection and vehicle scheduling. *Logistics and Transportation Review 16 (4)*, 325–338.

Bent, R. and P. V. Hentenryck (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers and Operations Research 33*(4), 875–893.

Benton, W. and R. Srivastava (1985). Product structure complexity and multilevel lot sizing using alternative costing policies. *Decision Sciences 16*, 357–369.

Berger, S. and C. Bierwirth (2010). Solutions to the request reassignment problem in collaborative carrier networks. *Transportation Research Part E: Logistics and Transportation Review 46*(5), 627–638.

Bolduc, M.-C., J. Renaud, and F. Boctor (2007). A heuristic for the routing and carrier selection problem. *European Journal of Operational Research 183*(2), 926–932.

Bolduc, M.-C., J. Renaud, F. Boctor, and G. Laporte (2008). A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of the Operational Research Society 59*(6), 776–787.

Bookbinder, J. and L. Koch (1990). Production planning for mixed assembly/arborescent systems. *Journal of Operations Management 9*, 7–23.

Buer, T., J. Homberger, and H. Gehring (2013). A collaborative ant colony metaheuristic for distributed multi-level uncapacitated lot-sizing. *International Journal of Production Research 51*(17), 5253–5270.

Buer, T., M. Ziebuhr, and H. Kopfer (2015). A coordination mechanism for a collaborative lot-sizing problem with rivaling agents. In *Dethloff, J.; Haasis, H.-D.; Kopfer, H.; Kotzab, H.; Schönberger, J. (eds) Proceedings of LM 2013*, pp. 325–337. Springer.

Burn, J. D. B. and R. A. Millen (1985). An evaluation of heuristic performance in multi-stage lot-sizing systems. *International Journal of Production Research 23*(5), 857–866.

Bussieck, M. R., A. Fink, and M. E. Lübbecke (1998). Yet another note on an efficient zero-one formulation of the multilevel lot-sizing problem. *Working Paper*.

Ceschia, S., L. Di Gaspero, and A. Schaerf (2011). Tabu search techniques for the heterogeneous vehicle routing problem with time windows and carrier-dependent costs. *Journal of Scheduling 14*(6), 601–615.

Chen, H. (2016). Combinatorial clock-proxy exchange for carrier collaboration in less than truck load transportation. *Transportation Research Part E: Logistics and Transportation Review 91*, 152–172.

Chu, C.-W. (2005). A heuristic algorithm for the truckload and less-than-truckload problem. *European Journal of Operational Research 165*(3), 657–667.

Clifton, C., A. Iyer, R. Cho, W. Jiang, M. Kantarcioglu, and J. Vaidya (2008). An approach to securely identifying beneficial collaboration in decentralized logistics systems. *Manufacturing & Service Operations Management 10*(1), 108–125.

Coleman, B. and M. McKnew (1991). An improved heuristic for multilevel lot sizing in material requirements planning. *Decision Sciences 22*(1), 136–156.

Cordeau, J.-F., G. Laporte, M. W. Savelsbergh, and D. Vigo (2007). Chapter 6 vehicle routing. In C. Barnhart and G. Laporte (Eds.), *Transportation*, Volume 14 of *Handbooks in Operations Research and Management Science*, pp. 367–428. Elsevier.

Côté, J.-F. and J.-V. Potvin (2009). A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. *European Journal of Operational Research 198*(2), 464–469.

Crowston, W. B. and M. H. Wagner (1973). Dynamic lot size models for multi-stage assembly systems. *Management Science 20*(1), 14–21.

Cruijssen, F., O. Bräysy, W. Dullaert, H. Fleuren, and M. Salomon (2007). Joint route planning under varying market conditions. *International Journal of Physical Distribution & Logistics Management 37*(4), 287–304.

Cruijssen, F., W. Dullaert, and H. Fleuren (2007). Horizontal cooperation in transport and logistics: A literature review. *Transportation Journal 46*(3), pp. 22–39.

Cruijssen, F. and M. Salomon (2004). Empirical study: Order sharing between transportation companies may result in cost reductions between 5 to 15 percent. Discussion Paper 2004-80, Tilburg University, Center for Economic Research.

Dantzig, G. and J. Ramser (1959). The truck dispatching problem. *Management Science 6*(1), 80–91.

Dellaert, N. and J. Jeunet (2000). Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm. *International Journal of Production Research 38 (5)*, 1083–1099.

Dellaert, N. and J. Jeunet (2003). Randomized multi-level lot-sizing heuristics for general product structures. *European Journal of Operational Research 148*(1), 211–228.

Desaulniers, G., J. Desrosiers, A. Erdmann, M. M. Solomon, and F. Soumis (2001). Vrp with pickup and delivery. In P. Toth and D. Vigo (Eds.), *The Vehicle Routing Problem*, pp. 225–242. Philadelphia, USA: Society for Industrial and Applied Mathematics.

Diaby, M. and R. Ramesh (1995). The distribution problem with carrier service: A dual based penalty approach. *ORSA Journal on Computing 7*(1), 24–35.

Drechsel, J. and A. Kimms (2011). Cooperative lot sizing with transshipments and scarce capacities: solutions and fair cost allocations. *International Journal of Production Research 49*(9), 2643–2668.

Dudek, G. and H. Stadtler (2005). Negotiation-based collaborative planning between supply chains partners. *European Journal of Operational Research 163*(3), 668–687. Supply Chain Management and Advanced Planning.

Dudek, G. and H. Stadtler (2007). Negotiation-based collaborative planning in divergent two-tier supply chains. *International Journal of Production Research 45*(2), 465–484.

Dumas, Y., J. Desrosiers, and F. Soumis (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research 54*(1), 7–22.

Ergun, O., G. Kuyzu, and M. Savelsbergh (2007). Reducing truckload transportation costs through collaboration. *Transportation Science 41*(2), 206–221.

Eslikizi, S., M. Ziebuhr, H. Kopfer, and T. Buer (2015). Shapley-based side payments and simulated annealing for distributed lot-sizing. In *Dolgui, A.; Sasiadek, J.; Zaremba, M. (Eds.): 15th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2015*, Volume 48 (3), pp. 1592–1597.

Eurostat (2016, January). Freight transport statistics.

Eyefortransport (2006, July). European 3pl market report.

Farris II, M.T.; Terrance, L. (2008). Evaluating the private fleet. *Transportation Journal 47(4)*, 51–66.

Fink, A. (1997). Untersuchungen zur effizienten lösbarkeit dynamischer, unkapazitierter, mehrstufiger mehrprodukt-losgrößenprobleme. Research Report 97/11, TU Braunschweig, Braunschweig.

Fink, A. (2004). Supply chain coordination by means of automated negotiations. In *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 3 - Volume 3*, HICSS '04, Washington, DC, USA. IEEE Computer Society.

Funke, B., T. Grünert, and S. Irnich (2005). Local search for vehicle routing and scheduling problems: Review and conceptual integration. *Journal of Heuristics 11*(4), 267–306.

Guardiola, L. A., A. Meca, and J. Puerto (2009). Production-inventory games: A new class of totally balanced combinatorial optimization games. *Games and Economic Behavior 65*(1), 205–219.

Heskett, J. L., N. Glaskowsky, and R. M. Ivie (1973). *Business Logistics,*. Ronald, New York.

Hiller, T. (2011). *Analyse ausgewählter Problemstellungen der Organisations- und Personalwirt.* Wiesbaden: Gabler.

Holzbaur, U. D. and G. Hachtel (2010). *Management für Ingenieure: Technisches Management für Ingenieure in Produktion und Logistik.* Wiesbaden: Vieweg + Teubner. Online-Ressource : v.: digital.

Homberger, J. (2006). Ein paralleler evolutionärer restart algorithmus zur lösung des multi-level unconstrained lot-sizing problems. *Logistik Management 1*(1), 54–68.

Homberger, J. (2008). A parallel genetic algorithm for the multilevel unconstrained lot-sizing problem. *INFORMS Journal on Computing 20*(1), 124–132.

Homberger, J. (2010). Decentralized multi-level uncapacitated lot-sizing by automated negotiation. *4OR: A Quarterly Journal of Operations Research 8*, 155–180.

Homberger, J. (2011). A generic coordination mechanism for lot-sizing in supply chains. *Electronic Commerce Research 11 (2)*, 123–149.

Homberger, J. and H. Gehring (2009). An ant colony optimization approach for the multi-level unconstrained lot-sizing problem. In *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, pp. 1–7.

Homberger, J. and H. Gehring (2010). A pheromone-based negotiation mechanism for lot-sizing in supply chains. In *Proceedings of the 44nd Hawaii International Conference on System Sciences*, pp. 1–10.

Homberger, J., H. Gehring, and T. Buer (2015). Integrating side payments into collaborative planning for the distributed multi-level unconstrained lot sizing problem. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, pp. 1068–1077.

Jones, L. (2012). *Conversational Statistics for Business and Economics* (3rd Edition ed.). Kendall Hunt Publishing Company.

Jurczyk, A., H. Kopfer, and M. Krajewska (6/2006). Speditionelle auftragsdisposition eines mittelständischen transportunternehmens. *Internationales Verkehrswesen 1*, 275–279.

Karimi, B., S. F. Ghomi, and J. Wilson (2003). The capacitated lot sizing problem: a review of models and algorithms. *Omega 31*(5), 365–378.

Karp, R. M. (1972). *Reducibility among Combinatorial Problems*, pp. 85–103. Boston, MA: Springer US.

Kirkpatrick, S., C. Gelatt, and M. Vecchi (1983). Optimization by simulated annealing. *Science 220*, 671–680.

Klein, M., P. Faratin, and Y. Bar-Yam (2002). Using an annealing mediator to solve the prisoners' dilemma in the negotiation of complex contracts. In *Revised Papers from the Workshop on Agent Mediated Electronic Commerce on Agent-Mediated Electronic Commerce IV, Designing Mechanisms and Systems*, AAMAS '02, London, pp. 194–202. Springer.

Klein, M., P. Faratin, H. Sayama, and Y. Bar-Yam (2003a). Negotiating complex contracts. *Group Decision and Negotiation 12 (2)*, 111–125.

Klein, M., P. Faratin, H. Sayama, and Y. Bar-Yam (2003b). Negotiating complex contracts. *Group Decision and Negotiation 12*(2), 111–125.

Klein, M., P. Faratin, H. Sayama, and Y. Bar-Yam (2003c). Protocols for negotiating complex contracts. *Intelligent Systems, IEEE 18*(6), 32–38.

Klincewicz, J., H. Luss, and M. Pilcher (1990). Fleet size planning when outside carrier services are available. *Transportation Science 24*(3), 169–182.

Kopfer, H., M. Krajewska, and A. Jurczyk (2006). Kooperation von profit centern im transportwesen endbericht des forschungsprojekts "integrierte auftragsdisposition in eigenständigen profit centern eines speiditionsunternehmens (inapce)". Technical report, Bremer Innovationsagentur.

Krajewska, M. and H. Kopfer (2005). Metaheuristics for the integrated operational transportation planning problem: an overview. In *Proceedings of MLS+CP Workshop "Combination of metaheuristics and local search with Constraint Programming techniques"*.

Krajewska, M. and H. Kopfer (2007). Collaborating freight forwarding enterprises - request allocation and profit sharing. *Kim K. H.; Günther H.-O. (Hrsg.): Container terminals and cargo systems - Design, Operations Management and Logistics Control Issues 1*, 365–381.

Krajewska, M. and H. Kopfer (2009). Transportation planning in freight forwarding companies: Tabu search algorithm for the integrated operational transportation planning problem. *European Journal of Operational Research 197*(2), 741–751.

Krajewska, M., H. Kopfer, G. Laporte, S. Ropke, and G. Zaccour (2008). Horizontal cooperation among freight carriers: request allocation and profit sharing. *Journal of the Operational Research Society 59*(11), 1483–1491.

Kratica, J., T. Kostic, D. Tosic, D. Dugosija, and V. Filipovic (2012). A genetic algorithm for the routing and carrier selection problem. *Computer Science and Information Systems 9*(1), 49–62.

Kuik, R. and M. Salomon (1990). Multi-level lot-sizing problem: Evaluation of a simulated-annealing heuristic. *European Journal of Operational Research 45*(1), 25–37.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research 59*(3), 345–358.

Lau, H. and Z. Liang (2001). Pickup and delivery with time windows: algorithms and test case generation. In *Tools with Artificial Intelligence, Proceedings of the 13th International Conference on*, pp. 333–340.

Li, H. and A. Lim (2001). A metaheuristic for the pickup and delivery problem with time windows. In *Tools with Artificial Intelligence, Proceedings of the 13th International Conference on*, pp. 160–167.

Li, Y., H. Chen, and C. Prins (2016). Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. *European Journal of Operational Research 252*(1), 27–38.

Liu, R., Z. Jiang, X. Liu, and F. Chen (2010). Task selection and routing problems in collaborative truckload transportation. *Transportation Research Part E: Logistics and Transportation Review 46*(6), 1071–1085.

Lu, Q. and M. Dessouky (2004). An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science 38*(4), 503–514.

Lübbecke, M. and J. Desrosiers (2004). Selected topics in column generation. *Operations Research 53*(6), 1007–1023.

Min, H. (1998). A personal-computer assisted decision support system for private versus common carrier selection. *Transportation Research Part E: Logistics and Transportation Review 34*(3), 229–241.

Müller-Merbach, H. (1983). Zweimal travelling salesman. *DGOR-Bulletin 25*, 12–13.

Nanry, W. P. and J. W. Barnes (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological 34*(2), 107–121.

Ohrt, C. (2008). *Tourenplanung im Straßengüterverkehr.* Wiesbaden: Betriebswirtschaftlicher Verlag Dr. Th. Gabler / GWV Fachverlage GmbH, Wiesbaden. Online-Ressource : v.: digital.

Özener, O., O. Ergun, and M. Savelsbergh (2011). Lane-exchange mechanisms for truckload carrier collaboration. *Transportation Science 45*(1), 1–17.

Pisinger, D. and S. Ropke (2010). *Large Neighborhood Search*, pp. 399–419. Boston, MA: Springer US.

Pitakaso, R., C. Almeder, K. F. Doerner, and R. F. Hartl (2007). A max-min ant system for unconstrained multi-level lot-sizing problems. *Journal of Computers and Operations Research 34*, 2533–2552.

Potvin, J.-Y. and M.-A. Naud (2011). Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier. *Journal of the Operational Research Society 62*(2), 326–336.

Ramaekers, K., A. Caris, G. Janssens, and T. Maes (2015). Pickup and delivery selection with compulsory requests. In *Sequeria Goncalves, P.J. (Ed.). Proceedings of the 21st European Concurrent Engineering Conference/11th Future Business Technology Conference/19th Annual Euromedia Conference*, pp. 28–33.

Ramaekers, K., A. Caris, T. Maes, and G. Janssens (2016). Pickup and delivery selection problem formulation and extension to problem variants. *Information Technology and Management Science 18*, 84–90.

Resende, M. and C. Ribeiro (2010). Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In M. Gendreau and J.-Y. Potvin (Eds.), *Handbook of Metaheuristics*, Volume 146 of *International Series in Operations Research & Management Science*, pp. 283–319. Springer US.

Ropke, S., J.-F. Cordeau, and G. Laporte (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks 49*(4), 258–272.

Ropke, S. and D. Pisinger (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science 40*(4), 455–472.

Rothe, J., D. Baumeister, C. Lindner, and I. Rothe (2012). *Einführung in Computational Social Choice: Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen.* Heidelberg: Spektrum Akademischer Verlag. Online-Ressource : v.: digital.

Rushton, A., P. Croucher, and P. Baker (2010). *The handbook of logistics & distribution management.* Kogan Page Limited.

Sahling, F. (2010). *Mehrstufige Losgrößenplanung bei Kapazitätsrestriktionen* (1. Aufl. ed.). Gabler.

Savelsbergh, M. and M. Sol (1995). The General Pickup and Delivery Problem. *Transportation Science 29*(1), 17.

Savelsbergh, M. and M. Sol (1998). Drive: Dynamic routing of independent vehicles. *Operations Research 46*(4), 474–490.

Schmidt, K. (1989). *Die Einzelkosten- und Deckungsbeitragsrechnung als Instrument der Erfolgskontrolle und Fahrzeugeinsatzdisposition im gewerblichen Güterfernverkehr*. Ph. D. thesis, Universität Frankfurt am Main.

Schönberger, J. (2005). *Operational Freight Carrier Planning*. Springer.

Schrijver, A. (2005). On the history of combinatorial optimization (till 1960). In G. N. K. Aardal and R. Weismantel (Eds.), *Discrete Optimization*, Volume 12 of *Handbooks in Operations Research and Management Science*, pp. 1–68. Elsevier.

Schwind, M., O. Gujo, and J. Vykoukal (2009). A combinatorial intra-enterprise exchange for logistics services. *Information Systems and e-Business Management 7*(4), 447–471.

Shapley, L. (1953). A value for n-person games. In *Kuhn, H. W., Tucker, A. W. (eds): Contributions to the Theory of Games*, Volume 2. Princeton University.

Shaw, P. (1997). A new local search algorithm providing high quality solutions to vehicle routing problems. Technical report, Department of Computer Science, University of Strathclyde, Scotland.

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research 35*(2), 254–265.

Song, J. and A. Regan (2003). An auction based collaborative carrier network. Technical report, Institute of Transportation Studies, University of California.

Stadtler, H. (2009). A framework for collaborative planning and state-of-the-art. *OR Spectrum 31*(1), 5–30.

Stadtler, H., C. Kilger, and H. Meyr (2015). *Supply Chain Management and Advanced Planning* (5th ed.)., Chapter 4, pp. 71–95. Springer Publishing Company.

Steinberg, E. and H. Napier (1980). Optimal multi-level lot sizing for requirements planning systems. *Management Science 26 (12)*, 1258–1271.

Stenger, A., D. Vigo, S. Enz, and M. Schwind (2013). An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science 47*(1), 64–80.

Taghipour, A. and J.-M. Frayret (2012). Mutual adjustment search with incentive for supply chain planning coordination. *International Journal of Computer Integrated Manufacturing 25*(10), 946–962.

Taghipour, A. and J.-M. Frayret (2013). Dynamic mutual adjustment search for supply chain operations planning coordination. *International Journal of Production Research 51*(9), 2715–2739.

Tang, O. (2004). Simulated annealing in lot sizing problems. *International Journal of Production Economics 88*(2), 173–181.

Tempelmeier, H. (2006). *Material-Logistik: Modelle und Algorithmen für die Produktionsplanung und -steuerung in Advanced Planning-Systemen : mit 127 Tabellen* (6, neubearb. Aufl. ed.). Berlin: Springer. Online-Ressource : v.: digital.

Thonemann, U. (2005). *Operations Management: [Konzepte, Methoden und Anwendungen].* Wirtschaft, BWL, Produktion & Logistik. München [u.a.]: Pearson Studium.

Vahrenkamp, R. (2015). 25 years city logistics: Why failed the urban consolidation centres? In *Proceedings of LM 2013*.

Veral, E. and R. LaForge (1985). The performance of a simple incremental lot-sizing rule in a multilevel inventory environment. *Decision Sciences 16*(1), 57–72.

Verdonck, L., A. Caris, K. Ramaekers, and G. K. Janssens (2013). Collaborative logistics from the perspective of road transportation companies. *Transport Reviews 33*(6), 700–719.

Vidal, T., M. Nelson, P. Penna, and L. Ochi (2015). Large neighborhoods with implicit customer selection for vehicle routing problems with profits. *Transportation Science 50(2)*, 720–734.

Voß, S. and D. Woodruff (2006). *Introduction to computational optimization models for production planning in a supply chain* (2nd ed.). Springer.

Wang, W. (2014). *Operational Transportation Planning of Modern Freight Forwarding Companies - Vehicle routing under consideration of subcontraction and request exchange.* Ph. D. thesis, University of Bremen.

Wang, X. and H. Kopfer (2014). Collaborative transportation planning of less-than-truckload freight. *OR Spectrum 36*(2), 357–380.

Wang, X. and H. Kopfer (2015). Rolling horizon planning for a dynamic collaborative routing problem with full-truckload pickup and delivery requests. *Flexible Services and Manufacturing Journal 27*, 509–533.

Wang, X., H. Kopfer, and M. Gendreau (2014). Operational transportation planning of freight forwarding companies in horizontal coalitions. *European Journal of Operational Research 237*(3), 1133–1141.

Wenger, W. (2010). *Multikriterielle Tourenplanung.* Ph. D. thesis, Universität Hohenheim.

Werner, H. (2013). *Supply Chain Management: Grundlagen, Strategien, Instrumente und Controlling* (5., ueberarb. u. erw. Aufl. 2013 ed.). Wiesbaden: Springer. Online-Ressource (XXXIV, 502 S.) : Ill.

Wiese, H. (2005). *Kooperative Spieltheorie.* München [u.a.]: Oldenbourg.

Wight, O. W. (1984). *Manufacturing Resource Planning: MRP II - Unlocking America's Productivity Potential.* John Wiley & Sons.

Xiao, Y., I. Kaku, Q. Zhao, and R. Zhang (2011). A reduced variable neighborhood search algorithm for uncapacitated multilevel lot-sizing problems. *European Journal of Operational Research 214*(2), 223–231.

Xiao, Y., I. Kaku, Q. Zhao, and R. Zhang (2012). Neighborhood search techniques for solving uncapacitated multilevel lot-sizing problems. *Computers & Operations Research 39*(3), 647–658.

Yelle, L. (1979). Materials requirements lot sizing: A multilevel approach. *International Journal of Production Research 17 (3)*, 223–232.

Zangwill, W. I. (1969). A backlogging model and a multi-echelon model of a dynamic economic lot size production system - a network approach. *Management Science 15*(9), pp. 506–527.

Ziebuhr, M., T. Buer, and H. Kopfer (2013). Agent-negotiation of lot-sizing contracts by simulated annealing with part-way resets. In *Klusch, M., Thimm, M., Papryzycki, M. (Eds.) Proceedings of MATES 2013*, Volume 8076, pp. 166–179. Springer.

Ziebuhr, M., T. Buer, and H. Kopfer (2015). A collaborative lot-sizing problem with production limitations. In *Computational Intelligence, 2015 IEEE Symposium Series on*, pp. 1005–1012.

Ziebuhr, M. and H. Kopfer (2014). The integrated operational transportation planning problem with complusory requests. In *Gonzalez-Ramirez, R.G. et al. (eds): Proceedings of ICCL 2014*, Volume 8760, pp. 1–15. Springer.

Ziebuhr, M. and H. Kopfer (2015). Collaborative transportation planning with forwarding limitations. In *Proceedings of OR 2015.*

Ziebuhr, M. and H. Kopfer (2016a). Solving an integrated operational transportation planning problem with forwarding limitations. *Transportation Research Part E: Logistics and Transportation Review 87*, 149–166.

Ziebuhr, M. and H. Kopfer (2016b). Transportation planning with forwarding limitations. In *Freitag, M.; Kotzab, H.; Pannek, J. (Eds.): Dynamics in Logistics - Proceedings of the 5th International Conference LDIC*, pp. 225–234. Springer.

Ziebuhr, M. and H. Kopfer (2017). Transportation planning with different forwarding limitations. In *Proceedings of OR 2016.*