– DISSERTATION –

Decentralized Code-Share Revenue Management in Airline Alliances

Max Gerlach

Department of Information Systems

Freie Universität Berlin

Berlin, October 2013

Decentralized Code-Share Revenue Management in Airline Alliances

Inaugural-Dissertation zur Erlangung des akademischen Grades eines Doktors der Wirtschaftswissenschaft am Lehrstuhl für Wirtschaftsinformatik des Fachbereichs Wirtschaftswissenschaften der Freien Universität Berlin

vorgelegt von

Max Oliver Gerlach

Master of Science in Operations Research

Universität Maastricht, Niederlande, 2009

aus Düsseldorf.

Berlin, im Oktober 2013

Erstgutachterin

Prof. Dr. Natalia Kliewer

Freie Universität Berlin

Zweitgutachterin

Prof. Dr. Catherine Cleophas

Freie Universität Berlin

Tag der Disputation

15. November 2013

Publikationen

Gerlach, M., Cleophas, C. und Kliewer, N. (2013). Codeshare-Allianzen bei Fluggesellschaften – Ein Segen für das Marketing und eine Herausforderung für Revenue-Management-Systeme, *WIRTSCHAFTSINFORMATIK* 55(3): 147-158

Sowie in englischer Sprache, da der Artikel zweisprachig erschienen ist:

Gerlach, M., Cleophas, C. and Kliewer, N. (2013). Airline Codeshare Alliances – Marketing Boon and Revenue Management Information Systems Challenge, *Business & Information Systems Engineering* 5(3): 153-163.

Arbeitspapiere

Gerlach, M., Cleophas, C. und Kliewer, N. (2013). Code-Share Revenue Management in Practice. Working Paper, Department of Information Systems, Freie Universität Berlin.

Gerlach, M., Cleophas, C. and Kliewer, N. (2013). Adaptive Code-Share Valuations for Airline Alliances. Working Paper, Department of Information Systems, Freie Universität Berlin.

Deutscher Abstract

Allianzen helfen Fluggesellschaften Synergien zu nutzen und zusätzliche Erträge zu generieren. Ein zentraler Baustein ist dabei die gegenseitige Vermarktung von Flügen, bekannt unter der Bezeichnung Codesharing. Wie Daten der Deutschen Lufthansa belegen, gewinnen Allianzen zunehmend an Bedeutung und die Nutzung von Codesharing wird konsequent ausgebaut. Dieser Trend sorgt für neue Herausforderungen bei der Ertragsoptimierung und dabei insbesondere in Bezug auf die Bewertung von Codeshare-Produkten in den internen Informationssystemen der einzelnen Fluggesellschaften.

Im Rahmen dieser Arbeit untersuchen wir wie Codeshare-Verbindungen ertragsoptimal gesteuern werden können. Wir entwicklen ein praxistaugliches Revenue-Management-Modell und leiten notwendige sowie hinreichende Bedingungen für die lokale Implementierung der zentralen Lösung her. Sollten diese nicht erfüllt sein, müssen neue Optimierungsparameter berechnet werde, wofür wir zusätzliche Update-Regeln definieren. Diese passen die Bewertungen der einzelnen Codeshare-Segmente an, so dass die lokalen Lösungen zum Optimum konvergieren. Wir zeigen die Wirkung der adaptiven Updates sowohl mit Monte Carlo Experimenten als auch mit stochastischen Simulationen.

Die Ergebnisse zeigen, dass adaptives Updaten besser funktioniert als andere statische und dynamische Bewertungsmethoden. Bei den Monte Carlo Experimenten erzielt es den höchsten Prozentsatz an optimalen Lösungen und minimiert den durchschnittlichen Fehler zwischen der lokalen und der optimalen Lösung. Zur Berechnung der stochastischen Simulationen nutzen wir den von Lufthansa entwickelten Revenue-Management-Simulator REMATE. Er bildet den gesamten Steuerungsprozesses samt stochastischen Kundenanfragen ab. Auch hier erzielt der adaptive Ansatz die besten Ergebnisse mit annähernd optimalen Erträgen.

Schlagworte: Luftfahrtallianzen, Codesharing, Informationssysteme, Optimierung, Ertragssteuerung, Lineare Programmierung, Simulation

English Abstract

Airlines form alliances to benefit from synergies and to raise additional revenues. A central tool in this strategy is the shared marketing of flights, referred to as code-sharing. As industry data from Lufthansa shows, alliances become increasingly important and code-share sales grow constantly. This trend imposes new challenges on the revenue management process and in particular on the valuation of the code-share fragments in the internal information systems of the independent airlines.

This thesis investigates how alliance partners can control code-share products such that the overall revenue is maximized. We propose a decentralized code-share revenue management model with limited information exchange, as it could be implemented in practice, and examine coordination strategies. In the process, we derive necessary and sufficient conditions for central optimality. The local optimization results must satisfy these conditions in order to implement the central optimal solution. Otherwise the code-share valuations must be updated and we specify rules for this such that the local solutions converge to the central one. We refer to this concept as adaptive valuation and test its performance with Monte Carlo Experiments as well as stochastic simulations.

The results show that adaptive valuation outperforms other common static and dynamic schemes. In the Monte Carlo Experiments, it provides the highest percentage of central optimal solutions as well as minimizes the average error between the local and the central solution. The large-scale simulations are evaluated in REMATE, a revenue management simulator developed at Lufthansa. It models the entire revenue management process and the system's interaction with stochastic customer requests. Also in this more complex environment the adaptation schemes give the best results and the revenues are close to the central ones.

Key Words: Airline Alliances, Code-Sharing, Information Systems, Optimization, Revenue Management, Linear Programming, Simulation

Contents

List of Figures

List of Tables

Part I.

Literature Review and Research Gap

1. Introduction

Collaboration is a common strategy for airlines to build competitive networks, to generate incremental revenues and to save costs (de la Torre, 1999; Brueckner, 2001; Morrish and Hamilton, 2002). The best-known type of collaboration are *airline alliances*. They experienced strong growth and increasing popularity over the last years. The number of members grew by 60 percent between 2003 and 2010 (Hu et al., 2013) and most large and mid-sized network carriers are engaged in one of the three big alliances, namely Star Alliance, Oneworld and SkyTeam. In 2010 the member airlines transported about 1,539 million passengers − 56 percent of the total air travel − and generated 61 percent of the industry-wide revenues. Chakravorty (2010) notices that competition among alliances has more and more replaced the competition among airlines and that each alliance strives for the best customer service and the most extensive network. The key facts about the three big alliances are summarized in Table 1.1.

Table 1.1.: Summary of Airline Alliances (Data Accessed October 18, 2012)

	Members	Revenue (bn.US$)	Passengers (mil.)	Aircrafts
Star Alliance[1]	27	182.24	678.98	4386
SkyTeam[2]	18	97.90	537.00	2644
Oneworld[3]	16	105.51	324.43	2381
Non-Affiliated[4]	—	249.95	1197.59	—

[1] http://www.staralliance.com/de/about/member_airlines/
[2] http://www.skyteam.com/en/About-us/Press/Facts-and-Figures/
[3] http://www.oneworld.com/news-information/oneworld-fact-sheets/oneworld-at-a-glance/
[4] http://www.airlines.org/Pages/Annual-Results-World-Airlines.aspx

The focus of this thesis is on the impact of alliances on internal information systems and in particular the revenue management process. *Revenue management* describes the art of selling the right seats to the right customers for the right prices at the right times (compare Smith et al., 1992). It is a central tool in most airlines' sales and ticketing processes and consists of three consecutive steps: *forecasting*, *optimization* and *inventory control*. Forecasting estimates the expected demand for every product using historic booking data. The optimization determines revenue-maximizing steering parameters. The inventory controls the sales process.

The traditional *network revenue management* problem monopolistically maximizes the revenue over the resources and products of a single airline. The resources are seats on the flights in the network. The products are itineraries – combinations of one or multiple flights that enable a customer to travel from some origin to some destination. Itineraries only including flights from a single airline are called *intraline itineraries*. Itineraries traversing flights from different airlines are referred to as *interline itineraries*. In this thesis, we consider a special kind of interline itineraries, named *code-share itineraries*. *Code-sharing* plays an important aspect in the formation of alliances and describes an arrangement where two or more carriers share the same inventory by assigning their designators to each others flights. It enables airlines to market their partner's inventory under their own name, to serve new markets and to feed more passengers into the own network without utilizing additional resources (Oum et al., 1996).

Since separate airlines control the flights constituting a code-share itinerary, the sales process becomes dependent on the decisions in their respective revenue management systems. The coordination of such control decisions is essential for the successful integration of alliances in the revenue management process and the central theme of this thesis.

1.1. Background

With the use of code-sharing we distinguish between the *marketing carrier* and the *operating carrier*. The former issues the ticket. The latter operates one or more of the flights. In order to sell a code-share ticket, all operating carriers must accept the code-share passenger from the marketing carrier and provide a seat on their flights. In the process, the carriers exchange their current inventory levels whenever a code-share request arrives. Since airlines typically share the availability of seats, this step is often referred to as *availability exchange*. The marketing carrier decides on the basis of the aggregated inventory information which of its products become available and hence, which customers are accepted and rejected.

Once a code-share passenger was accepted, the marketing carrier collects the money, issues the ticket and compensates the other carriers for using their capacity. Special *revenue sharing schemes* govern the compensation payments and distribute the realized code-share revenues among the airlines.

Availability exchange and revenue sharing are the two central levers in code-share management. It has been recognized that the choice of these schemes significantly impacts the revenue management process and may distort control decisions as well as degrade the revenue of the individual airlines and the alliance as a whole (Darot, 2001; Wright, 2010; Jain, 2011; Hu et al., 2013). The explanation to this observation is twofold: First, improper availability exchange may lead to inaccurate availabilities and revenue losses on code-share itineraries (direct or *first-order* effect). In addition, code-share customers occupy seats that can no longer be used for intraline passengers, leading to a sub-optimal traffic mix in the entire network (indirect or *second-order* effect; Wright, 2010; Gerlach et al., 2013). Second, improper revenue sharing may not compensate an airline adequately for the displacement of potential intraline customers. If the revenue allocation does not exceed what each carrier could earn in its intraline network, the alliance might not be stable and its members not willing to cooperate.

In this context it is important to recognize that airlines act on two levels: On the one hand, they sell own products – intraline itineraries – and aim at maximizing the profit from them. Intraline itineraries are fully controlled by the private revenue management systems, their availability depends solely on own resources and the carriers keep the complete revenue such that there is no interaction with external partners. On the other hand, airlines supply capacity to combined products – code-share itineraries – to generate incremental revenues through alliances. Code-share itineraries lie partially outside the own networks and are affected by the control decisions of the partners. The revenue earned by each airline depends on the fare charged by the marketing carrier to the customer as well as the revenue sharing scheme.

The coordination of both levels is important to achieve the full potential of the alliance, but difficult to realize in practice. Interviews with practitioners have revealed that alliance members operate distinct and often heterogeneous systems, requiring consistent availability exchange methods. Moreover, most current revenue management methodologies do not explicitly model code-share itineraries and cannot capture their unique value. Practitioners report further that alliance partners hardly share information. The information exchange is mostly limited to local availabilities. Other information, e.g. about future expected demand and prices, remains private. Finally, revenue sharing is often based on the distance flown by each carrier. The distance is used as a measure for the relative operating cost, but does not account for the potential value of a flight.

1.2. Motivation and Goals

The motivation for this thesis stems from the importance of code-sharing in airline alliances and the resulting problems on revenue management. In order to show how the use of code-sharing has evolved over the last years, we conducted a small data analysis using data from Lufthansa German Airlines. Lufthansa is the largest European carrier and one of the founding members of Star Alliance, the world's leading alliance network. Based on the data provided, we find that the amount of flights used for or impacted by code-sharing has increased tremendously over the last 10 years. While the number of flights operated by Lufthansa remained relatively constant, the share of flights exclusively marketed by Lufthansa dropped from 60 percent in 2000 to less than 20 percent in 2010. In the same time, the number of flights marketed by Lufthansa but operated by at least one other airline (OAL), increased by more than 80 percent. Figure 1.1 depicts the complete development of the average number of Lufthansa marketing flights per week over the last 17 years. They are divided into three categories: flights (1) operated and marketed by Lufthansa, (2) operated by Lufthansa and marketed by at least one other airline and (3) operated by another airline and marketed through Lufthansa.

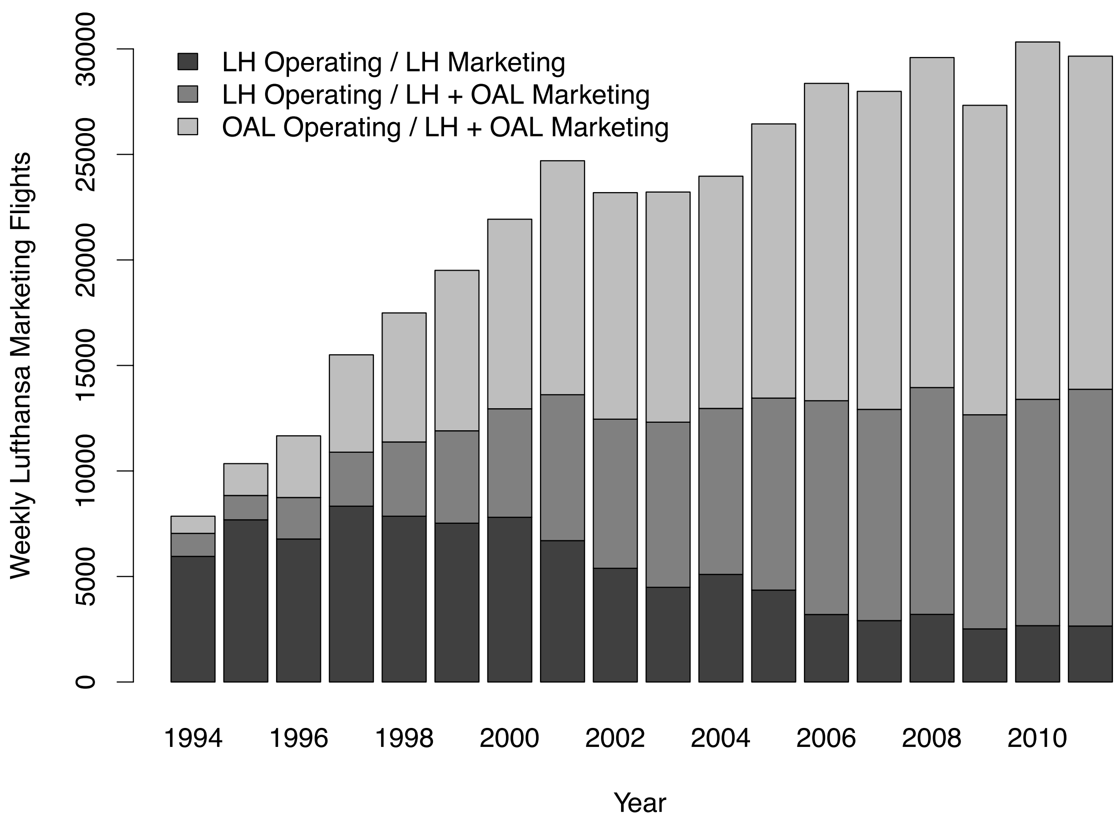

Figure 1.1.: Development of Lufthansa Marketing Flights

In terms of bookings, we find that code-sharing accounts for around seven to nine percent of the total Lufthansa bookings[5]. On the one hand, this is low compared to the number of flights used for code-sharing. On the other hand, the share of code-share bookings increases significantly on *hub-to-hub routes*, as depicted in Figure 1.2, and is estimated to reach up to 65 percent on individual flights (Netessine and Shumsky, 2005).

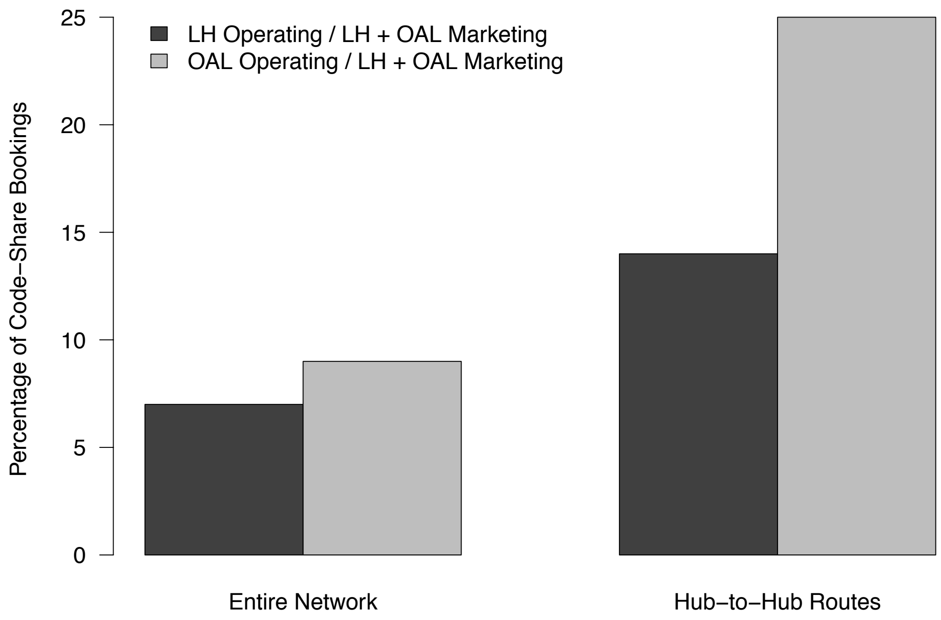

Figure 1.2.: Share of Lufthansa Code-Share Bookings in 2011

The high number of code-share flights as well as the significant amount of code-share bookings indicate the relevance of alliances in today's airline business and call for well-developed code-share revenue management techniques. From a theoretical point of view, the best solution is central forecasting and optimization across the entire inventory. The monopolistic revenue management problem is well documented, maximizes the total revenue and resolves the problems occurring under decentralized control (Boyd, 1998). However, there are several reasons why centralized control may not be feasible in practice. The three main ones are legal, technical and organizational constraints.

[5]Practitioners from other airlines report numbers between six to 16 percent.

- **Legal:** Antitrust law restricts the exchange of private information among independent carriers, which is necessary to merge revenue management systems. While full immunity can only be achieved through a merger, more and more alliance members are granted partial immunity (Gagnepain and Marín, 2010), including joint scheduling or pricing rights as well as the limited exchange of private information. The additional information can be used to improve alliance coordination, but is not sufficient for the implementation of central inventory control.

- **Technical:** Switching to a single system requires the harmonization of individual price and product structures and the agreement on a common revenue management methodology. Such standardization might not be beneficial if the carriers have different business requirements, for example one carrier focuses on low-cost point-to-point traffic, while the other operates a hub-and-spoke network. Moreover, the technical implementation of a common system is costly and takes a long time. The complete information systems infrastructure and many business processes need to be adapted, sunk costs for the old systems occur and the revenue management departments need to be trained on the new system (Chakravorty, 2010; Randall, 2010; Maruna and Morrell, 2010; Feldman, 1998). Even when theoretically possible, this process may become a nightmare in practice as Boyd (1998) points out.

- **Organizational:** Some international alliances have around 20 or more members that are located in several time zones around the world and have diverse business cultures. Coordinating day-to-day business processes under these circumstances creates organizational burdens. Even individual airlines have built extensive networks with multiple hubs, long- and short-haul services as well as different brands. Managing such a network is a challenging task because the larger the network, the more different interests clash with each other and the harder it becomes to coordinate them. Controlling an alliance spanning several of these networks might be considered impossible.

From a more theoretical perspective Kim (1991) mentions four prevalent disadvantages of centralization. They are complexity, timeliness, vulnerability and inefficiency. Complexity refers to the computational complexity of the underlying optimization problem. Even with the best and fastest algorithms, it might be no longer tractable. Timeliness and vulnerability mean the operational flexibility and stability of a system. A decentralized system has shorter response times and is less vulnerable toward malfunctions. Inefficiencies arise when the central system cannot account for specific attributes of the sub-systems.

Finally, there can be political reasons against centralization, for example when top management considers selling or outsourcing some part of the company. A central system would make such decisions more difficult and costly since the individual parts are no longer independent.

In conclusion, centralized alliance control as the simplest alternative, is unlikely or even impossible to implement in practice. However, also decentralization creates challenges, as we noted in the previous section. Without central revenue management, inventory control is distributed over smaller sub-networks belonging to different airlines. Code-share itineraries interconnect these sub-networks and affect the revenue management decisions throughout the network. As our data analysis shows, the use of code-sharing has constantly increased over the last ten years and code-share bookings account for a significant share of the total bookings. Thereby, the risk of potential distortions as well as revenue losses increases, highlighting the necessity for sophisticated code-share control.

The goal of this thesis is to analyze the alliance revenue management problem from a theoretical as well as practical perspective. We provide a detailed discussion on the challenges related to code-share control and propose a coordination scheme that qualifies for practical implementation. In particular, we assume that the airlines in the alliance operate independent revenue management systems and possess limited information about their partners. They are neither able to implement a central system, nor one carrier can solve the revenue management problem of its partner to gain additional information for its local control decisions.

The algorithm that we develop, compares the decentralized control decisions from different carriers on the same code-share itinerary. Whenever the carriers' decisions do not coincide, we adapt the valuations of code-share itineraries in the optimizer to improve the local decisions. In the process, we derive necessary and sufficient conditions for central optimality and define update rules such that the decentralized solutions converge to the central solution. To demonstrate convergence, we conduct Monte Carlo experiments in a static environment isolating the relevant effects. We compare the outcome of our approach against several other prevalent valuation schemes and get insights into their theoretical performance. Finally, large-scale stochastic simulations show the performance in a more realistic setup. They model the complete revenue management process and capture its full dynamics.

1.3. Structure of this Thesis

This thesis is divided into three parts. Part I provides an introduction to airline alliances and revenue management. It describes how code-sharing is currently handled in the industry and discusses recent theoretical approaches to alliance revenue management. Part II presents the alliance model, introduces the adaptive code-share valuation scheme and analyzes its theoretical performance. Part III further investigates the performance of the newly-developed scheme from Part II. In a large-scale simulation study, we compare and evaluate the adaptation procedure against other commonly used valuation schemes. The thesis closes with a discussion of the findings and an outlook toward future research directions.

Part I: In the first part of Chapter 2, we present a comprehensive overview of airline alliances. We discuss their benefits, relate them to other types of collaboration and emphasize the importance of code-sharing. In the second part, we summarize the state-of-the-art literature on revenue management and outline major advances.

Chapter 3 describes how code-share itineraries are currently controlled in practice and how the resulting challenges are approached from a theoretical side. Our literature review includes the published work on airline alliance revenue management and provides a critical assessment of its relevance and applicability. Following this discussion, we derive research opportunities originating from gaps in the literature.

Part II: In Chapter 5, we develop the decentralized alliance model and discuss its underlying assumptions. We theoretically derive a novel approach to coordinate revenue management decisions in this setting. It is introduced in Chapter 6. The idea is to adapt code-share valuations over time such that the local solutions converge to the central one. In the process, we propose optimality conditions in Section 6.3 and adaptation rules in Section 6.4. Their performance is investigated in Chapter 7 with a series of Monte Carlo Experiments.

Part III: In Chapter 8, we discuss practical implications stemming from the findings in Part II. Afterwards, Chapter 10 analyzes the impact of the valuation schemes using the revenue management simulation environment introduced in Chapter 9. Embedded in the complete revenue management process with stochastic demand, the performance of the methods is tested across varying market conditions and compared against current industry practice as well as other valuation schemes.

Chapter 11 closes this thesis with a summary of our work and a critical assessment of the findings, including a discussion on their practical relevance as well as suggestions for future research directions.

2. Literature Review

This chapter reviews the state-of-the-art literature on the two fundamental concepts used in this thesis: airline alliances and revenue management. Section 2.1 starts with an introduction to alliances and provides an overview of important aspects related to the formation of alliances. Section 2.2 discusses the revenue management process and highlights the crucial steps.

2.1. Airline Alliances

Alliances are the most popular type of collaboration in the airline industry and their relevance has constantly grown over the last decade, as shown in the first chapter. We continue the previous analysis with a more detailed discussion on the benefits of alliances, including a thorough description of code-sharing. Finally, we relate the idea of alliances to similar types of collaboration, list typical network topologies and examine the impact of alliances on consumer welfare, prices and competition.

2.1.1. Benefits of Alliances

De la Torre (1999) identifies five main benefits of alliance. They are increases in network coverage, higher traffic volumes, cost savings from synergies, access to foreign markets and reductions in competition. An important component in achieving these benefits is code-sharing. Code-sharing describes the process of assigning multiple airline designators to a single flight. For example, the Lufthansa flight LH400 from Frankfurt to

New York is also sold by United Airlines as flight UA8841, by Thai Airways as flight TG7700 and by Brussels Airlines as flight SN7230. The airline that operates the flight, i.e. that provides the aircraft and crew, is called operating carrier (here LH). The airline issuing the ticket is called marketing carrier (here LH, UA, TG or SN). Introducing marketing flights allows alliance members to collaboratively sell their inventory and to extend their network reach to foreign markets. They increase market penetration, feed additional passengers into the own network and generate incremental revenues (Gerlach et al., 2013; Shumsky, 2006; Vinod, 2005). Last, code-sharing enables airlines to expand globally since other forms of cooperation such as mergers are restricted by foreign ownership and trade laws (Park, 1997; Oum et al., 1996).

Next to the benefits from code-sharing, carriers in an alliance exploit synergies by the joint usage of infrastructure as for example airport lounges, gates and check-in facilities at foreign airports as well as the consolidation of maintenance and ground operations. On an administrative level potential savings arise through joint purchasing and marketing activities (Morrish and Hamilton, 2002). Further, airlines can harmonize their price and product structures under partial antitrust immunity (Brueckner and Whalen, 2000). KLM–Northwest was the first alliance with partial immunity (1993), followed by others as recently United Airlines–Continental Airlines–ANA (2009), American Airlines–British Airways (2010) and Lufthansa–ANA (2011). Because competition laws restrict the exchange of private information, back-end processes such as reservation systems remain separated (Feldman, 1998).

Access to partners' distribution channels enhances the sales process and increases market penetration through higher visibility in the *global distribution systems* and on the Internet – the two most important sales channels of most airlines (Boyd and Bilegan, 2003). Higher visibility comes from displaying a single flight under several marketing flight numbers. In addition, code-share itineraries appear as online connections, which are ranked higher in the display order than conventional interline itineraries.

Customers benefit from a wider variety of connections and coordinated flight schedules with shorter transfer times and higher frequencies. *Seamless service* increases the convenience on combined itineraries through single ticketing, check-in and baggage drop-off, and provides the impression of traveling on a single airline (Brueckner, 2001). Merged *frequent flyer programs* allow passengers to earn and redeem miles on the entire alliance network and privileges of status customers such as lounge access, priority boarding and late check-in are recognized alliance-wide (Goh and Uncles, 2003; Park, 1997).

Despite the above mentioned benefits, alliances also generate additional costs. Costs occur for monitoring and managing the alliance as well as adapting business practices (de la Torre, 1999). Using empirical data from several US-based airlines, Chua et al. (2005) examine the effects on costs and find that most alliances have only a small negative effect on costs. In a more general context, Kleymann and Seristö (2001) analyze the risks and benefits of alliances. They argue that carriers should carefully balance the investments they take to improve alliance performance, and the risks of loosing flexibility and sovereignty. Nicolaou and Christ (2011) examine the effects of risk sharing and information systems integration on the perceived risk and performance of an alliance. They find that higher integration increases the perceived risk, while risk sharing decreases it. Furthermore, perceived risk negatively relates to the perceived performance, suggesting that risks associated with an alliance may reduce its expected performance. For a further discussion, we refer to Tomkins (2001), who studies the interaction of risk, trust and information.

To ensure the stability of the alliance, it is important that benefits and costs are fairly distributed. As carriers remain financially independent, most of their decisions are likely to aim at the individual businesses without considering externalities on other members. It is the task of the alliance management to prevent non-cooperative behavior by implementing sophisticated control mechanisms and fostering trust among the partners (Goerzen, 2005; Ireland et al., 2002). At the same time, the selfish nature of each member should be respected. Discussions on alliance management in general are provided in Osborn and Hagedoorn (1997) and Gulati (1998). The role of organizational structures in corporate alliances is stressed in Killing (1988).

2.1.2. Related Types of Collaboration

Next to alliances, there are two further types of collaboration that promise similar benefits, but also raise similar problems.

Mergers and Acquisitions

As in many other industries, mergers and acquisitions are also frequenly observed in the airline industry. Oberstar (2008, p. 1) writes that today most carriers "are the product of one or more mergers over the past three decades". This trend continues and a series of mergers among major carriers has occured during the last years. In Europe, it started with the merger between KLM and Air France in 2004. It was followed by Lufthansa's acquisition of Swiss Airlines (2005), Brussels Airlines (2008), bmi (2009; sold again 2012) and Austrian Airlines (2009). Finally, in spring 2010, the third-largest European network carrier, British Airways, signed a merger with Spain's Iberia. In the U.S., there were 18 big mergers between 1978 and 2005 (Maruna and Morrell, 2010), followed by the ones between Delta Airlines and Northwest Airlines (2008), Continental Airlines and United Airlines (2010) as well as American Airlines and US Airways (2013).

Just as alliances, mergers or acquisitions are a popular instrument for carriers to create economies of scale and scope, save costs, improve profitability and to reduce competition (Maruna and Morrell, 2010). In addition, merged airlines enjoy antitrust immunity and have the possibility to exchange private information or to join business processes, allowing them to exploit additional benefits from their cooperation.

In practice, however, we observe different levels of integration. In some cases, they remain separate companies with distinct brands and collaboration is restricted to an operational level – similar to alliances. In other cases, both partners are fully integrated, one brand disappears and all backend and administrative processes are merged. An example for the former is the Lufthansa Group. An example for the latter is Delta Airlines.

Without full integration, the coordination problem faced by the merged airlines is similar to the one faced by alliances. The key difference is that the airlines are not financially independent. While it is important for alliances to distribute the benefits and costs in a fair manner, merged carriers can primarily focus on maximizing their overall performance. The question of how benefits and costs are internally shared is secondary, though not less important, because each airline still operates independently based on the financial compensation it receives. Finally, antitrust immunity simplifies the information exchange and the realization of potential benefits.

Subsidiaries and Divisions

The third type of collaboration are interdependent divisions or subsidiaries that belong to the same company. They take own decisions in order to maximize their performance. Nevertheless, they are financially dependent and their success depends on how well they cooperate (Groves and Loeb, 1979).

Examples for subsidiaries are found in many industries including the airline market. For example, many network carriers have outsourced low-volume feeder operations to carriers that operate at lower costs. Examples are Lufthansa Regional, Qantas Link and Delta Connection. On behalf of the major carriers, they connect regional markets to the hubs using smaller aircrafts (Shumsky, 2006). Additionally, several airlines have founded low-cost subsidiaries as Germanwings (Lufthansa) or JetStar (Quantas). They take over point-to-point services outside the hubs, which could not be efficiently served by the network carriers themselves.

The distribution of decision-making among internal divisions becomes necessary because of the size and complexity of many companies. Instead of a single decision-maker who manages hundreds of flights, control is distributed, for example, by hubs in multi-hub networks, traffic areas (i.e. east- and west-bound traffic), traffic-types (i.e. long- and short-haul flights) or markets (i.e. India-USA). Such decentralization allows for gaming among the decision-makers, as they might act selfishly to improve their individual performance. To encourage collaborative behavior, it is essential to internally allocate incentives such that all divisions and subsidiaries comply with the company's overall strategy. This problem corresponds to the revenue sharing aspect in alliances and mergers. The difference is that divisions are fully integrated.

2.1.3. Network Topology

Airlines differentiate two kinds of intraline itineraries (compare Jain, 2011):

- direct or local itineraries, using exactly one flight, versus
- transfer or connecting itineraries, traversing two or more flights.

While intraline itineraries span flights of the same airline, interline itineraries use flights from different airlines. A special variant of interline itineraries are code-share itineraries, as for example the typical two-leg code-share itinerary depicted in Figure 2.1.

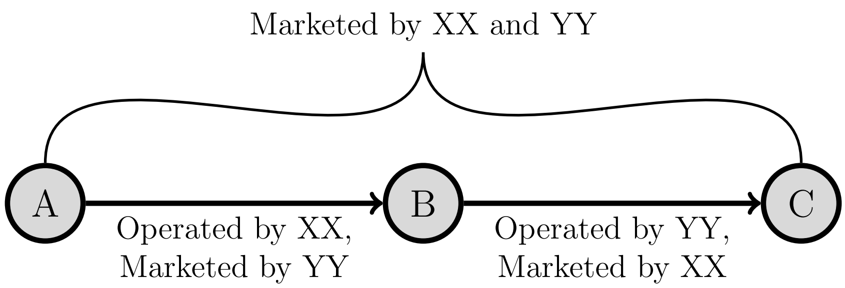

Figure 2.1.: Two-Flight Code-Share Itinerary.

To further understand the impact of code-sharing, it is useful to introduce the concept of strategic *substitutes* and *complements*. Generally speaking, if products are substitutes, they serve the same purpose and customers are indifferent between them. If products are complements, they serve different purposes, but combined they create a new, superior product.

In the context of alliances, Gagnepain and Marín (2010) state that alliance members are substitutes if their networks overlap on a majority of routes. Such an alliance is called parallel alliance. If the airlines mainly offer new markets that none of the unallied carriers serves individually, they are complements. Similarly, a *parallel code-share* exists when each carrier operates the route individually and a *complementary code-share* evolves when the flights of different airlines serve a new city pair (Oum et al., 1996). If the marketing carrier does not operate any segment of the itinerary, this is called *virtual code-share* (for example see Ito and Lee, 2007, for more details). Virtual code-shares can be regarded a marketing tool to increase sales, as Gayle (2007) argues, and will not be further considered in this thesis.

Comparing flight schedules from the last ten years, we found that the networks of international alliances are mostly complementary, representing the main objective of their members to extent market reach and to get access to foreign countries. Between 2001 and 2011 the percentage of city pairs directly served by more than one carrier from the same alliance decreased from around 20 percent to less than 15 percent. Moreover, the use of code-sharing on complementary flights tripled in the same time period. A typical example for complementary alliance partners are the Star Alliance members Lufthansa and United Airlines. They operate distinct networks except for a few hub-to-hub routes and only Lufthansa serves smaller European markets, while only United serves smaller American markets. Code-sharing enables Lufthansa to sell tickets from any of its European destinations to any of United's American destinations and vice versa. The typical structure of an international alliance network is depicted in Figure 2.2.

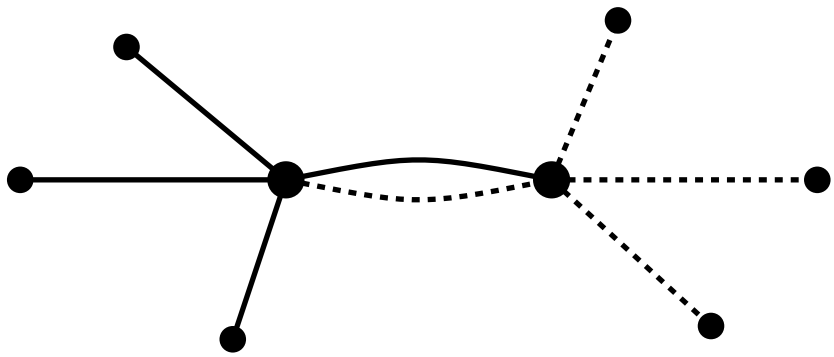

Figure 2.2.: Structure of a Typical Alliance Network with two Complementary Partners

As the previous paragraph suggests, real-world networks are neither fully complementary nor fully parallel. We refer to this type of strategic relationship as hybrid. We also note that there are multiple examples in which flights from a single airline act as substitutes or complements. For instance, in hub-and-spoke networks the flights constituting a transfer itinerary are complements (Economides and Salop, 1992), while flights between the same city pair at similar times or transfer itineraries via different hubs are substitutes.

2.1.4. Welfare Effects

Several theoretical and empirical studies discuss the ambivalent effect of airline alliances on social welfare, prices and output. Adler and Smilowitz (2007) suggest a model to study airline alliances and mergers in competitive markets. Their approach uses revenue as well as cost objectives to determine which partners fit to each other and how their collaboration impacts social welfare. Brueckner (2001) and Park (1997) find that the impact of alliances depends on the network structure: A complementary alliance increases welfare. A parallel alliance decreases welfare. For his study, Park defines a network with three airports – two spokes (A and B) and one hub (H). Three airlines compete on the markets AH, BH and AB, where AB is a connecting service via H. Airline 1 serves all three markets, while Airlines 2 and 3 serve markets AH and BH, respectively. Starting with the pre-alliance case, Park establishes a complementary model in which

Airlines 2 and 3 cooperate as well as a parallel model in which Airlines 1 and 2 cooperate. He finds that in the complementary case Airline 1 reduces its output and earns less profit, whereas the cooperating airlines increase output and profits. In the parallel alliance, partners decrease output on their monopoly market AH and increase it on the other markets. Airline 3 looses on its local market with profits behaving analogously.

From an empirical side these findings are supported by Gagnepain and Marín (2010), Brueckner (2003), Morrish and Hamilton (2002) and Brueckner and Whalen (2000). All of them report substantial price decreases on complementary code-share itineraries accompanied by higher traffic volumes and better productivity. For instance, Brueckner and Whalen state that fares of alliance partners are up to 25 percent lower than those of unallied carriers. Focusing on domestic alliances, Bamberger et al. (2004) find that prices fell by five to seven percent, while traffic rose by about six percent.

On substituting routes and in particular between hubs, the same studies report negative welfare effects. Similar to the findings in Kim and Singal (1993), who analyze mergers, it turns out that increased market power dominates higher efficiency and that prices increase. Nevertheless, most authors argue that the positive impacts prevail in hybrid networks and that alliances are overall beneficial for consumers.

While the above papers are restricted to a single alliances, Zhang and Zhang (2006) examine rivalry between alliances. An example for this are the alliances between Lufthansa and United Airlines as well as British Airways and American Airlines. On the one hand, Lufthansa and British Airways are substitutes on European routes, while United Airlines and American Airlines are substitutes on American routes. On the other hand, the networks of Lufthansa (British Airways) and United Airlines (American Airlines) are mostly complementary, leading to two complementary alliances competing against each other. Zhang and Zhang show that complementary alliances as well as rivalry between such alliances increases social welfare.

2.2. Revenue Management

This section briefly outlines the history of revenue management, points out milestones in its evolution and provides an overview of the approaches currently applied in the airline industry. For a more comprehensive overview we refer the reader to the book by Talluri and van Ryzin (2004) and the survey papers by McGill and van Ryzin (1999) as well as Pak and Piersma (2002).

2.2.1. Revenue Management Concept

Revenue management describes the art of controlling different fare products in order to maximize expected revenue. It is widely used in many industries as for example airlines, hotels, car rentals and cruise ship lines, but may be applied in various other industries as well. Sophisticated revenue management techniques have become a crucial factor for success and may improve revenues between two to eight percent (Cross, 1997; Michaels, 2007; Boyd, 1998).

In order to apply revenue management successfully, a market should satisfy a few conditions: The products should be perishable, the capacity fixed, the sales period finite and the variable costs low. Furthermore, revenue management requires that customers can be segmented by certain characteristics. A typical example is business and leisure demand for air travel. Business passengers are less price-sensitive, book their tickets late and like flexibility with respect to re-bookings and cancellations. On the contrary, leisure passengers are very price-sensitive, book their tickets early and do not require last minute changes in their booking. By offering multiple products with distinct restrictions (e.g. minimum stay, advance purchase, non-refundability, etc.), different customer segments can be addressed at the same time. Revenue management has the task to determine which products are offered at every point in time and how much capacity is reserved for each of them.

The first revenue management systems were introduced by airlines after the deregulation of the market in the late 1970s. To compete against emerging low-cost competitors, incumbents introduced restricted discount-fare products, for example the 'Super Saver Fares' by American Airlines in 1977 (Smith et al., 1992). With multiple fare products the airlines were able to match competitors' offers in low-yield markets and simultaneously protect their original high-yield markets.

The new challenge was to control the different fares: Protecting too many seats for the full fare product leaves some seats empty that could have been filled by low value demand. Protecting too few spills high value demand. In 1972, Littlewood suggested to sell the discounted fare as long as its revenue exceeds the expected revenue of selling the same seat for the full fare in the future (Littlewood, 1972). His result is known as *Littlewood's Rule* and lay the foundation to the revenue management principles as we know them today.

The classical revenue management process consists of three consecutive steps − forecasting, optimization and inventory control. The forecasting estimates the expected bookings, cancellations and no-shows based on historic observations. These forecasts and the current bookings are used in the optimization to calculate capacity allocations and overbooking limits. The inventory controls the sales process based on the optimized steering parameters.

Airlines are the most prominent users of revenue management systems and also the drivers of innovations. In the following paragraphs, we briefly summarize the large body of airline revenue management literature and introduce the reader to several important notions used in later parts of this thesis. Other well-known applications such as hotels, air cargo and liner shipping are discussed in Goldman et al. (2001), Kasilingam (1996) and Agarwal and Ergun (2010), respectively.

Related topics that we do not cover explicitly, are overbooking and dynamic pricing. For early work on overbooking we refer to Rothstein (1971) and Vickrey (1972). The benefits of overbooking are analyzed in Alstrup et al. (1989) and Suzuki (2006). Gallego and van Ryzin (1994) develop optimal dynamic pricing controls for stochastic demand over a finite booking horizon. Dynamic pricing of multiple products is studied in Maglaras and Meissner (2006) as well as Gallego and van Ryzin (1997). Its application to airline markets under competition is presented in Currie et al. (2008) and Luo and Peng (2007), while Zhang and Cooper (2009) focus on dynamic pricing for substituting flights. For a survey we refer the reader to the book by McAfee and te Velde (2006). Pricing in general is discussed in Vinod (2010).

2.2.2. Forecasting

Demand forecasting is crucial for the success of revenue management. Various studies including Cleophas (2009), Weatherford and Belobaba (2002), Pölt (1998) and Lee (1990), show how forecast quality impacts optimization results and revenues. For example, Lee finds that every 10 percent improvement in forecast accuracy increases the expected revenues by three percent.

Air travel demand is forecasted on *origin-destination* level to reflect the passenger's wish to get from some origin to some destination. On the supply side, airlines meet this demand by offering either a direct flight or a set of connecting flights. Especially in hub-and-spoke networks many passengers transfer from one flight to another. Forecasting just the aggregate demand on each flight does not accurately reflect the unique traffic flows and results in revenue distortions.

The classical forecasting process assumes that demand between fare products is independent and that the probability to receive a request for a particular booking class does not depend on the availability of other booking classes. In the early days of revenue management, this was a valid assumption because airlines used several fare restrictions to differentiate their products. However, the emergence of *low-cost carriers* and the growing use of the Internet have changed the market. Low-cost carriers generally offer unrestricted booking classes with products that only differ by the price. The Internet increases transparency because customers can easily compare offers from various carriers. Both undermines the business model of incumbent airlines and pushes them towards less- or un-restricted fares as well. Consequently, they loose the ability to practice price discrimination and demand is no longer independent (Boyd and Kallesen, 2004; Tretheway, 2004). Using traditional forecasting methods in such cases causes the so-called *spiral-down effect* and leads to revenue losses (Cooper et al., 2006).

To account for demand dependences, new forecasting approaches differentiate between *price-oriented* and *product-oriented* customers. Product-oriented customers prefer a specific product (zero percent buy-down) – as assumed in the traditional revenue management literature. Price-oriented customers buy the cheapest available class (100 percent buy-down). Forecasting both groups simultaneously is called *hybrid forecasting*. The best known hybrid approach is commonly referred to as Q-Forecasting and was introduced by Cléaz-Savoyen (2005) and Reyes (2006). Both studies describe several variants of Q-forecasting and provide comprehensive simulation results. As an extension to these methods, Fiig et al. (2010) introduce the idea of fare adjustments. They allow to use traditional revenue management algorithms even when demand is dependent. Fare adjustments transform the demand and prices of a general discrete choice model to an independent demand model by accounting for the buy-down risk stemming from passengers willing to buy several booking classes.

A recent enhancement of hybrid forecasting is market-oriented forecasting. It estimates the buy-down behavior between any two products that an airline offers. This increases the accuracy of the forecasts and better reflects customer behavior, but is also more complex as the number of forecast variables grows.

The process of recording observed bookings and updating the forecasts is called *history building* or *unconstraining*. Since some products might be unavailable over the booking horizon, the observed bookings are censored and the actual demand has to be estimated from such censored bookings in order to avoid gaps in the data. Techniques for unconstraining are discussed in Zeni (2001), Weatherford and Pölt (2002) and Queenan et al. (2007). For more details on forecasting and in particular the forecasting of air travel demand, we refer to the work by van Ryzin (2005), Zickus (1998) and Skwarek (1997).

2.2.3. Optimization

The largest body of revenue management literature is devoted to the optimization step. It requires demand forecasts and fares as input and calculates control parameters in form of booking limits or bid prices (see Section 2.2.4). Since 1972 several researchers have proposed a variety of different methods to optimally or heuristically solve the seat allocation problem. These approaches can be clustered in two main categories, namely *flight* (leg) and *network* (origin-destination) approaches.

Flight Optimization

Given forecasts on flight-booking class level, leg optimization determines seat allocations for every flight separately. This problem is well studied and several researchers have proposed various approaches to it. Among others, Curry (1990), Wollmer (1992) and Brumelle and McGill (1993) develop optimal nested controls, while Lee and Hersh (1993) suggest a dynamic programming formulation. Yet, the best-known concept is *expected marginal seat revenue* (EMSR), a heuristic policy developed by Belobaba (1987, 1989). EMSR calculates nested booking limits by successively solving Littlewood's Rule for multiple classes. A refined version of EMSR, called *EMSRb*, is widely used in the industry. It is easy to implement and provides good results on real-world networks.

The greatest disadvantage of leg optimizers is that the leg availabilities are used for all passenger requests independent of their actual value and their routing. A major enhancement is therefore their extension to multiple (interdependent) flights.

Network Optimization

Compared to leg-based methods, network approaches optimize and control all *origin-destination-itinerary-booking class* combinations (or simply itineraries) separately to account for their network value. It has been recognized that network optimization leads to revenue gains in the order of one to three percent for moderate and high load factors depending on the underlying demand and network characteristics (McGill and van Ryzin, 1999; Talluri and van Ryzin, 2004). It is particularly useful in hub-and-spoke networks with a high share of transfer passengers.

Nevertheless, the finer granularity of decision variables makes the network revenue management problem increasingly complex. Talluri and van Ryzin (1998) present an optimal, though computationally prohibitive, dynamic programming approach. It becomes intractable even on small networks because the state space increases exponentially with the number of flights. Knowing this, various researchers have suggested approximation schemes, which either simplify the network model or decompose it into independent single-resource problems. In both categories there exist multiple approaches.

Simplifications of the network model include the *Deterministic Linear Program* (DLP) first mentioned by Glover et al. (1982) and the *Randomized Linear Program* investigated in Talluri and van Ryzin (1999). DLP produces partitioned capacity allocations and assumes deterministic demand. The result of DLP is an upper bound on the total revenue that can be realized. However, partitioned booking limits have several limitations as discussed in the next section, and the assumption of deterministic demand is problematic in practice. To remedy the latter problem, Talluri and van Ryzin introduce a randomized version of the DLP model that explicitly accounts for demand variability. They replace the expected demand by a random vector of demands and use it to approximate the optimal value function. With some numerical examples they show that their approach slightly improves the DLP solution by up to 0.32 percent.

Decomposition approaches are among others *Prorated-EMSR* (Williamson, 1992), *Displacement Adjusted Virtual Nesting* (DAVN) (Smith and Penn, 1988; Williamson, 1992) and *Dynamic Programming Decomposition* (Zhang, 2011; Talluri and van Ryzin, 2004, Section 3.4.4). The underlying idea is to distribute the revenue of multi-leg itineraries among the individual flights. Then the network problem can be approximated by optimizing each flight separately using standard leg optimization techniques and the adjusted revenues. Following the same thought, Topaloglu (2009) uses Lagrangian multipliers to decompose the problem by flights and shows that this approach outperforms the above mentioned methods.

Finally, van Ryzin and Vulcano (2008) and Bertsimas and de Boer (2005) suggest simulation-based optimization to determine network control parameters. The integration of customer choice behavior in network optimization is presented in Kunnumkal and Topaloglu (2009) as well as Meissner and Strauss (2011).

2.2.4. Inventory Control

There are two widely used methods to control the booking process – *booking limits* and *bid prices*. Both can be used for flight as well as network control.

Booking Limits

Booking limits are upper bounds on the number of salable seats per fare product. If such booking limits are non-overlapping, they are called *partitioned booking limits*, otherwise they are called *nested booking limits*. Booking requests under both schemes are accepted if and only if the requested booking class is available on all flights of the itinerary.

- **Partitioned Booking Limits:** A fixed number of seats is allocated to every flight-booking class combination in the flight case or every itinerary-booking class combination in the network case. Partitioned booking limits have two disadvantages. First, because they are non-overlapping, they are not robust towards stochastic demand. If demand for a product exceeds the assigned capacity, it is spilled independent of its value. Second, the seats on a single flight must be distributed among many different booking classes and itineraries, each of them having relative low expected demand. This may cause inefficiencies and makes partitioned booking limits impractical to use.

- **Nested Booking Limits:** Nested booking limits allow higher booking classes to access the protected capacity of lower classes as well. Compared to partitioned booking limits, seats are assigned to groups of booking classes and high-value demand gets less likely spilled. However, establishing a nesting under network control is not trivial due to the unknown ordering of booking classes from different itineraries. It may be approximated by grouping classes with similar value and then calculating booking limits for the resulting groups instead of all individual classes. This approach is called *virtual nesting* (van Ryzin and Vulcano, 2008; Bertsimas and de Boer, 2005).

Bid Prices

Bid prices represent the marginal value of the resources, i.e. the next seat on a flight. They are calculated on flight level and a customer request gets accepted if its value exceeds the sum of bid prices on all flights it uses. This concept is known as *additive bid prices* and was introduced by Simpson (1989) and Williamson (1992). In a later study, Talluri and van Ryzin (1998) show that additive bid prices are asymptotically optimal by analyzing their performance under a general demand model.

Compared to booking limits there are less control parameters necessary, which makes bid prices more efficient and simpler to use. Moreover, the previously mentioned studies indicate the effectiveness of bid price control in real-world world networks and therefore, they are the most popular control mechanism in the industry. Disadvantages are (1) that there is no upper bound on the number of salable seats at each price and (2) that the actual value of a request is not considered − requests are accepted independent of how much they exceed the bid price (Pak and Piersma, 2002; Darot, 2001). To partially overcome these problems, bid prices should be frequently updated over the booking horizon and when bookings occur.

Having motivated airline alliances as well as introduced the central concepts in the revenue management literature, the next chapter continues with the application of revenue management to code-share alliances and the resulting problems.

3. Revenue Management in Airline Alliances

As noted in Chapter 1, airline alliances play an important role in the industry. The majority of network carriers is engaged in an alliance and together they generated revenues of about 346 bn.$ in 2010. A large share of these revenues came from code-share itineraries, making them a significant source of revenue for many airlines[6].

Following the trend in the aviation market, alliance revenue management has become an active area of research. Several researchers have proposed a variety of theoretical and practical approaches to different aspects of the problem. In this chapter, we review the relevant literature and describe the current implementation of code-sharing in the industry. Our information was gathered in a series of interviews with industry experts and practitioners from Lufthansa and affiliated airlines.

3.1. Alliance Game

From a theoretical perspective the code-share process can be divided into two levels – the contractual and the operational level (Hu et al., 2013; Darot, 2001; de la Torre, 1999). On the contractual level, the alliance partners negotiate the terms of their collaboration, for example which flights are used for code-sharing, how they control the booking process and how the revenue from code-share sales is distributed. On the operational

[6]OneWorld members report shares of 10 or more percent (Hu et al., 2013)

level, airlines operate distinct revenue management systems to maximize their individual revenues. Based on the alliance agreement defined on the contractual level, the carriers implement the sales process in their systems and manage code-share transactions. The resulting booking information is used as feedback to monitor the performance of the alliance and to renegotiate the agreement, if necessary. The complete process is schematically depicted in Figure 3.1.

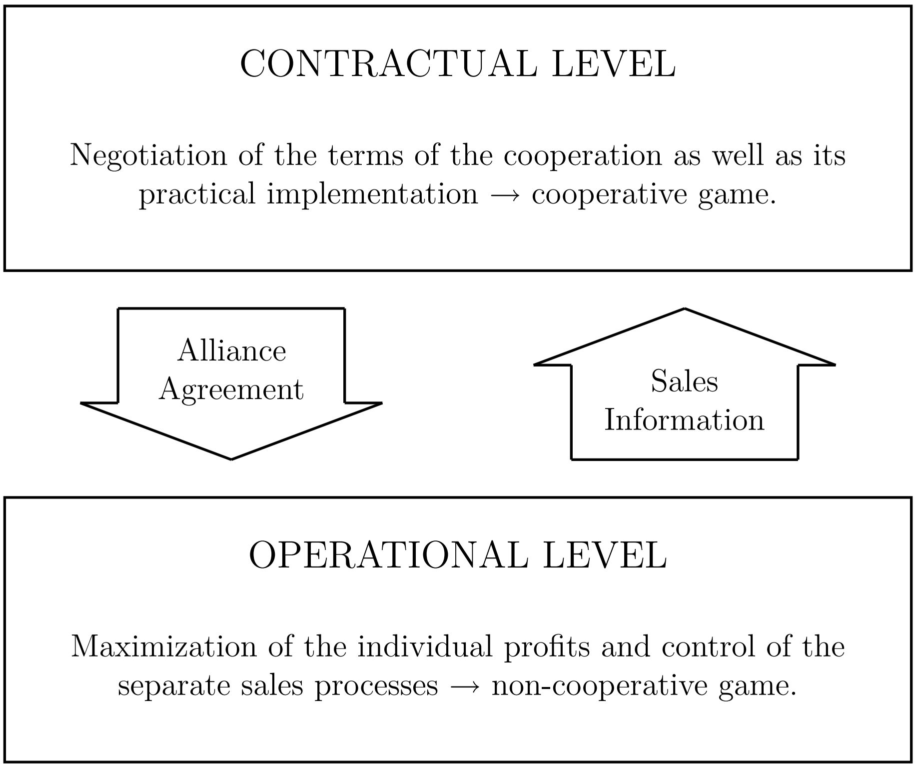

Figure 3.1.: Contractual and Operational Level in Alliance Management

Both levels may also be described by game-theoretic concepts. *Game theory* analyzes the behavior of several players in some interaction model called the game. On the contractual level the airlines face a cooperative game, on the operational level a non-cooperative game.

3.1.1. Contractual Level: Cooperative Game

On the contractual level, airlines negotiate the terms of their cooperation and try to find a setup that makes the alliance beneficial as a whole as well as for every participant. Ideas on how to find such a structure and particularly a fair compensation for the usage of shared resources, are provided in the field of *cooperative game theory*.

A cooperative game consists of (1) a set of players, and (2) some *characteristic function* that determines the value of a *coalition* (subset of cooperating players) (Brandenburger, 2007). The set of all players is called *grand coalition*. The goal of cooperative game theory is to distribute the value of a coalition among its members. A solution is said to be in the *core* if the sum of the payoffs over all players is equal to the value of the grand coalition and the payoff to each player is higher than what this player could achieve in any sub-coalition. The first property is referred to as budget balance property. The second is known as the stability property and prevents the players from colluding.

In the airline alliance game, the value of an alliance equals the revenue that its members achieve in their combined network by pooling their resources. In view of the revenue management process, the most important aspect from the contractual level is the revenue sharing scheme. The concept of the core suggests to distribute the benefits of the alliance such that it is beneficial to all members, implying that each carrier should earn more revenue than in any sub-alliance. An unbalanced distribution may motivate individual airlines to leave the alliance or to not behave in its best interest. Çetiner and Kimms (2012) show theoretically that the alliance game has a non-empty core. So, there always exists an allocation of code-share revenues such that the members form the grand coalition and each airline is better off. Moreover, we note that the alliance game is super-additive: With proper code-share control the carriers may only gain from satisfying code-share demand. Rejecting all code-share passengers and solely serving intraline routes provides a lower bound on their performance.

3.1.2. Operational Level: Non-Cooperative Game

On the operational level, the airlines separately control their inventory. Since they may take independent and selfish decisions, coordination is necessary to ensure that the full potential of the alliance is realized. The appropriate framework to study this behavior is provided in the field of *non-cooperative game theory*.

Non-cooperative game theory describes games in which the players decide independently on their strategies. The players are rational and attempt to maximize their individual utility. However, utilities do not only depend on the own strategy, but also on the strategies of the other players. The strategy that maximizes the own utility given the strategies of the partners is called *best response*. Mutual best responses of all partners lead to a *Nash Equilibrium*. Nash equilibrium is the central equilibrium concept in non-cooperative game theory and describes a situation in which no player has an incentive to deviate by unilaterally changing its strategy (Boulogne et al., 2002).

In airline alliances the acceptance of code-share passengers depends on the individual acceptance decisions of the operating carriers. Only when all airlines provide the necessary capacity, the marketing carrier can issue the code-share ticket. The acceptance decisions constitute a Nash equilibrium if no carrier may gain by changing its decision. Hu et al. (2013) show that this is the case when all airlines reserve the same capacity for a common itinerary. Without recapturing the formal proof, we note that the carriers can sell at most the minimal capacity provided by any of the operating carriers. A carrier reserving more capacity for code-share passengers may want to change its strategy and reallocate the capacity to another itinerary. Therefore, this airline has an incentive to deviate and in equilibrium all carriers must provide the same capacity.

3.2. Code-Share Revenue Management in Practice

Besides revenue sharing, the focus of this thesis is on the operational level and in particular the following issues: code-share forecasting and optimization (including history building and code-share valuations) as well as availability exchange. The role of these aspects in the alliance revenue management process is depicted in Figure 3.2.

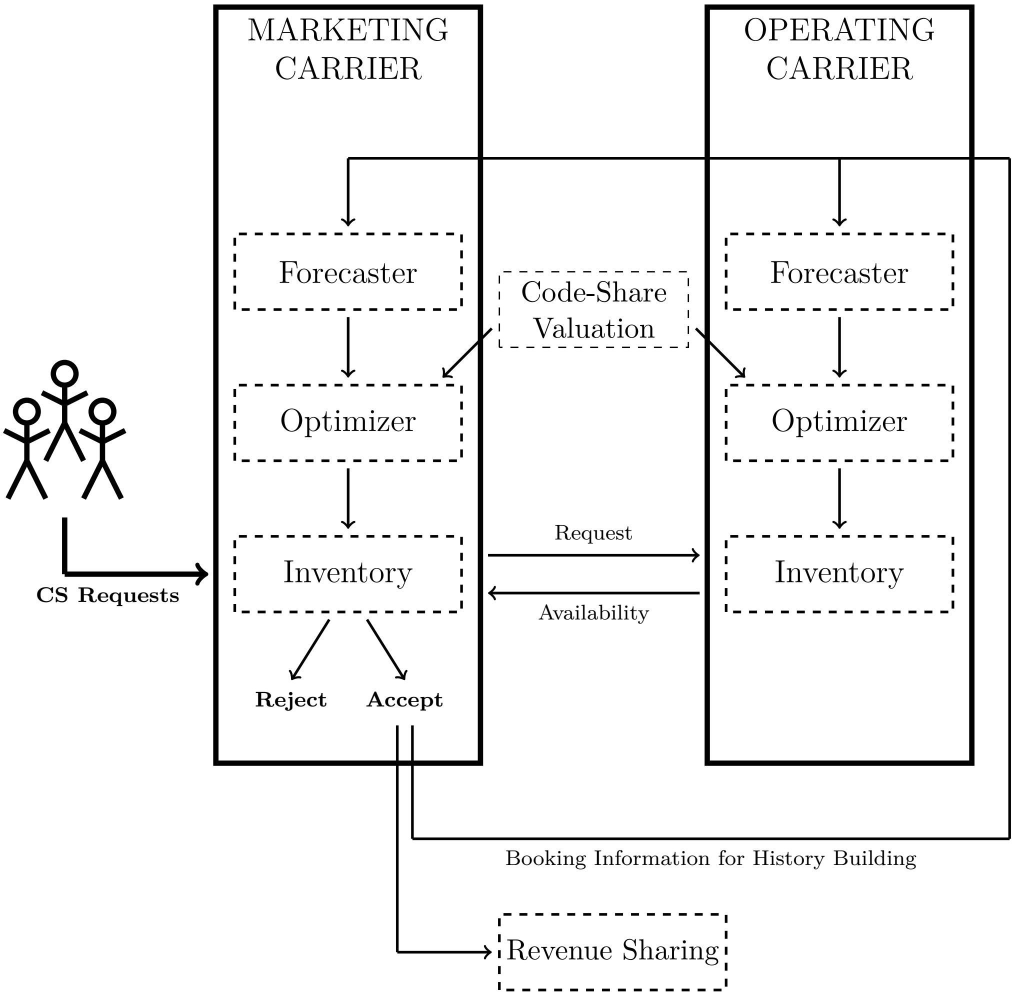

Figure 3.2.: Alliance Revenue Management Process

The process begins with the marketing carrier forwarding a code-share request to the operating carrier. In return it receives the availability on the flights that are outside its own network. Since the carriers might have different booking classes, a *booking class* *mapping* translates the booking class availabilities of one carrier into the booking class structure of the partner. It maps each of the operating carrier's booking classes into one of the marketing carrier's classes. Based on the aggregated availability the marketing carrier decides whether a customer is accepted or rejected. If the ticket is sold, the fare is divided and the booking information transfered to each carrier for history building. The updated forecasts are used to reoptimize the steering parameters.

3.2.1. Code-Share Forecasting

The forecasting of code-share bookings depends on the revenue management capabilities of the operating carrier as well as the information exchange with the marketing carrier. Without the complete routing information or without a network revenue management system, code-share bookings are treated as local bookings on the respective intraline sub-itineraries. The operating carrier simply observes a local booking in a local booking class determined by the booking class mapping. The carrier might not know whether the booking is part of a code-share itinerary, in which booking class of the partner it occurred (hence, which restrictions the customer accepted) and how much the customer paid. In other words, the operating carrier cannot distinguish intraline and code-share bookings and thus, they are forecasted in a single variable. This process is not ideal, but the transfer of itinerary-specific booking information is often prevented due to organizational or technical barriers.

If, on the contrary, the alliance partners are able to exchange the complete routing information and a carrier has the necessary system requirements, it can store and forecast code-share bookings separately from intraline itineraries. The advantage of this is similar to switching from flight to network revenue management in the classical case: It allows to control the itineraries separately and to account for their unique value to the network. The problem of how to determine the network value of code-share passengers to an individual carrier is further addressed in Section 3.2.2.

Code-Share History Building

The history building (or unconstraining) has to deal with similar complications as the forecasting. First of all, the marketing carrier monitors the availability situation as offered to the customer, while the operating carrier only observes its local availability. Without the full availability information, forecast and buy-down values may not be updated correctly and get distorted in the long run.

Second of all, carriers often use different forecasting methodologies, for example one carrier calculates buy-down, while the other assumes independent demand. In such cases, the observed booking behavior as well as the unconstraining process may not be consistent with the data structures and model assumptions of the various partners.

Last, inaccurate mappings can distort the history building process. For example, if the mapping is not unique, bookings from several classes are counted in a single booking class and cannot be distinguished. Similarly, the value of the mapped booking classes might not be equivalent when the carriers use different price or product structures.

3.2.2. Code-Share Optimization

In the same way as forecasting, the optimization of code-share bookings depends on the revenue management capabilities of a carrier as well as the information exchange among the partners. With the appropriate capabilities and given separate forecasts, code-share itineraries can be optimized analogously to intraline itineraries. However, while each carrier knows the exact value of its intraline itineraries – the price charged to the customer –, the value of code-share itineraries needs to be estimated. The marketing carrier knows the final price, but has to pay a part of it to the operating carrier. The operating carrier in return observes the transfer payment without knowing what the customer actually paid. The choice of the valuations is crucial for the outcome of the optimization and we highlight several possible schemes in the next section.

Code-Share Valuations

Valuations are fed together with demand forecasts into the optimizer to calculate seat availabilities. They directly affect the sales process of the respective itinerary, but also impact control decisions on other itineraries that share the same capacity. Generally speaking, the higher the valuation, the more likely a product becomes available.

Counting code-share bookings together with local bookings implies that they are valued at the local fares. If each airline does that, the total value of a code-share booking becomes the sum of its local valuations. This does not necessarily correspond to its real value and may result in sub-optimal control decisions. As an alternative, separate code-share optimization allows to assign each code-share itinerary a unique value. Several schemes have been proposed for this purpose. They can be divided into static and dynamic ones. Static prorates do not change over time or when capacity depletes. Examples for static schemes are proration based on distance and proration based on local fares. Dynamic schemes are usually based on the bid prices, which depend on the time and the current capacity levels.

- **Local Fare:** Each carrier uses the local fare of the booking class in which the customer is counted. Local fare valuation overestimates the actual value of a code-share passenger since the sum of local fares is typically higher than the fare of the transfer itinerary.

- **Full Fare:** Each carrier uses the total fare of the code-share itinerary to value its intraline sub-itinerary. This policy systematically overestimates the actual code-share value.

- **Fixed Amount or Fixed Percentage:** Every segment along the code-share itinerary is valued by a fixed amount or a fixed percentage of the total fare. The sum of these valuations should amount to the complete itinerary fare.

- **Local Fare Proration:** Because the full economy fare is usually in Y-class, this proration scheme is sometimes referred to as *Y-proration*. It weighs the relative value of the individual flights by the ratio of the highest fares on the respective intraline sub-itineraries. Alternatively, the fares of the local booking classes in which the code-share booking is counted, may be used. This provides booking class specific prorates.

- **Mileage Proration:** The code-share fare is prorated based on the distance flown by each carrier. The distance can be interpreted as a measure for the operating costs of each carrier.

- **Absolute Bid Price:** The code-share fare is adjusted by the current bid prices of the partner to reflect the possible displacement of other passengers. In aggregation, absolute bid price adjustments tend to over- or underestimate the value of code-share itineraries since the surplus revenue remains with every carrier.

- **Bid Price Proration:** The code-share fare is prorated by the ratio of the current bid prices. This proration scheme reflects the relative value of a seat given the other demand. Contrary to the absolute bid price adjustment, the surplus revenue is distributed among the partners.

In a simulation study, Darot (2001) compares how local fare, full fare and local fare proration impact alliance revenues. He finds that revenue performance varies only slightly, but especially the booking class mix changes significantly. The higher code-share itineraries are valuated, the more bookings are observed on these itineraries and the more is the traffic mix affected in the local network. Other studies as Wright (2010) and Hu et al. (2013) study valuation schemes implicitly when analyzing the effects of revenue sharing. We discuss revenue sharing in Section 3.2.4 and also point out the differences to the valuations used in the optimizer.

3.2.3. Code-Share Booking Control

To control the booking process on code-share itineraries, the marketing and the operating carrier exchange current availability information. Following Boyd (1998) and Vinod (2005) there are two variants: *blocked space* and *free sale* agreements.

- **Blocked space agreement:** The marketing carrier receives some share of its partner's capacity, which can be interpreted as a virtual flight, and is responsible for managing and selling the allocated seats. The operating carrier considers these seats as sold and reduces its inventory accordingly. Because sales might realize unbalanced among the carriers, soft blocks allow for regular reallocation of seats over the booking horizon. Hard blocks are fixed once they are established.

- **Free sale agreement:** Free sale is more dynamic than blocked space agreements. Every operating carrier controls its own inventory and transfers local availabilities in form of *Availability Status* (AVS) messages to the marketing carrier. AVS contains flight-booking class availabilities and does not transmit any itinerary information. So, all code-share itineraries receive the same availability information – the one from the local itinerary. Additionally, a booking class mapping needs to map the booking classes of the operating carrier into the ones of the marketing carrier. Based on the mapped availabilities the marketing carrier determines the final code-share availability.

  An alternative to AVS is bid price exchange or *bid price sharing* (BPS). Instead of booking class availabilities, the operating carrier transfers the flight-compartment bid price to its partner. As most network carriers control their inventory with bid prices, this method becomes more and more popular. It reduces the information transfer and diminishes the risk of bad mappings. However, BPS requires antitrust immunity as bid prices are private information. To determine the code-share availability, the marketing carrier sums up the bid prices of all flights along the itinerary.

Analyzing industry data from Lufthansa, we find that about 95 percent of the partnerships use a free sale environment to govern code-share sales and that AVS is used in 97 percent of the cases to determine code-share availabilities. Moreover, talking to practitioners has revealed that only a few airlines world-wide have implemented BPS, although many carriers use bid price inventory and enjoy partial immunity with their partners. Darot (2001) estimates the benefit of BPS over AVS to be around one percent in alliance revenue.

Finally, we note that in practice the marketing carrier usually accepts or rejects a request. This kind of control is referred to as *single availability control*. In contrast, *dual availability control* requires that the operating carrier has to accept the request as well. Otherwise the code-share itinerary is not sold. The latter is more accurate, but also more restrictive, and information must be exchanged in both directions each time a customer request arrives.

3.2.4. Revenue Sharing

In bilateral agreements the alliance partners specify how code-share revenues are distributed. This is referred to as revenue sharing and the individual contracts are called *special prorate agreements*. Several researchers, such as Wright (2010) and Hu et al. (2013), point out the importance of revenue sharing on alliance performance. On the one hand, the alliance wants to maximize combined revenues. On the other hand, the carriers maximize their individual shares. Inadequate compensation can make collaboration unattractive to individual partners and may result in significant losses for the alliance. Instead, sophisticated mechanisms should encourage cooperative behavior and guarantee that every carrier receives a fair share of the code-share revenue (Boyd, 1998; Vinod, 2005; Shumsky, 2006).

Revenue sharing contracts are negotiated on the contractual level during the formation of the alliance and they are fixed before the sales period starts (de la Torre, 1999). On the contrary, code-share valuations are used on the operational level during the optimization process. Although both steps distribute code-share revenues, their impact is fundamentally different: Revenue sharing allocates the revenues that were actually realized during the sales process, while the optimizer valuations influence which itineraries and booking classes can be sold at all. Moreover, we notice that optimizer valuations are only used in the internal systems and the carriers may choose any values that lead to consistent optimization results. The carriers are consequently more flexible with the choice of these values, for example they do not have to add up to the actual code-share fare. In contrast, revenue sharing can only distribute the realized revenue. This implies that some schemes such as local and full fare valuation are not feasible.

In practice, revenue sharing and code-share valuations are usually two separate schemes. Because most carriers do not distinguish intraline and code-share itineraries in the revenue management process, they value code-share bookings with the local fares. At the same time, most prorate agreements use production costs, typically represented by flown mileage, as the main criterium to allocate revenues. This is also the default policy suggested by the International Air Transportation Association (IATA). It should be applied whenever no other rule has been defined.

3.3. Literature Review

Next to practical challenges, the presence of code-sharing has also led to a new and active area of research. Because centralized inventory control is not feasible, several authors have proposed novel approaches to manage alliances and to coordinate separate revenue management systems. In the following paragraphs, we summarize this literature and discuss relevant results from related subjects and other industries.

3.3.1. State-of-the-Art Alliance Revenue Management

Boyd (1998) and de la Torre (1999) are among the first authors who raise the alliance revenue management problem. Boyd discusses ideas of centralized and decentralized inventory control in alliance networks and points out practical challenges that need to be addressed. The early work by de la Torre provides a comprehensive overview of aspects related to the formation of alliances and the effects of code-sharing on revenue management. His discussion includes advantages and shortcomings of different availability exchange and proration methods. Moreover, he highlights their potential impact on the individual carriers.

Boyd (1998) as well as Vinod (2005) propose equilibrium conditions to the decentralized inventory control problem faced by alliances under soft-blocking. The idea is to trade seats on code-share flights until the marginal seat revenues match for all partners (Boyd, 1998), or until their bid prices are equal (Vinod, 2005). In equilibrium, i.e. when these conditions hold, the decentralized systems achieve the optimal alliance revenue. While intuitive appealing, most airlines operate under free sale agreements and do not include their partners' flights in the optimization. Moreover, the reallocation of seats must happen in real time and iteratively − both is difficult to realize in practice. The existence of equilibrium conditions in more general resource exchange alliances, similar to airline alliances under soft- or hard-blocking, are studied in Chun et al. (2011). They evaluate them against the cases with no alliance and perfect coordination.

Darot (2001) and Jain (2011) present two extensive simulation studies on alliance revenue management using the PODS simulation environment[7]. They analyze how current code-share practices affect alliance performance as well as the members. Compared to most other contributions, their analyses are not limited to a few code-share itineraries. Instead they use a large-scale network with multiple competing airlines, hub-and-spoke

[7]We further discuss the use of revenue management simulations as well as the development of PODS in Chapter 9.

traffic and realistic customer behavior. The focus of Darot's work is the customer's perception of code-share itineraries. He finds that high perception, i.e. when customers perceive the offers from different alliance members as distinct alternatives, substantially increases the total revenue. Moreover, he studies alliance performance under symmetric and asymmetric revenue management methodologies and introduces the concept of bid price sharing (see Section 3.2.3). Jain focuses on the revenue impacts of different availability exchange and history building methods. He shows that sharing the complete routing information leads to revenue increases in the range of 0.1 percent due to more accurate forecasts. Both studies verify their findings using various valuation schemes and revenue management methodologies.

A comprehensive theoretical analysis is presented in the dissertation by Wright (2010). In the first part, which is also published by Wright, Groenevelt and Shumsky (2010), he formulates a two-partner alliance as a markovian game. His model uses the network dynamic programming formulation by Talluri and van Ryzin (1998) and assumes complete information. The author derives equilibrium conditions for several static and dynamic proration schemes[8] and tests their performance in terms of total alliance revenue. He finds that dynamic proration provides better results and is more robust than static proration. The author further demonstrates with a counterexample that no markovian transfer price guarantees optimal alliance revenue. The problem are sub-optimal intraline decisions, which cannot be prevented in a markovian system. Finally, their analysis shows that the performance of dynamic proration quickly degrades when carriers choose transfer prices maximizing their individual revenues. The reason is that the operating carriers report higher prices to capture a larger share of the revenue. Wright refers to this scheme as *partner-price scheme* and it builds the foundation for Kadatz (2011), who demonstrates in a game-theoretic setting how carriers may gain additional revenues by manipulating their bid prices. Kadatz concludes that such schemes are problematic as long as the partners are not able to verify each others honesty.

[8]Note that he uses the same scheme to valuate the code-share itinerary in the optimizer and to distribute the realized revenues.

The initial model by Wright, Groenevelt and Shumsky is based on the assumption that each airline has full information about the state of the partner. This includes demand forecasts, future arrival probabilities and current inventory levels. Even in a highly integrated alliance, this is hardly possible. Moreover, their approach is computationally expensive because it relies on the network dynamic program and every carrier must solve the revenue management problem of its partner as well. This makes the model intractable even for small alliances. Wright acknowledges this fact in the second and third part of his dissertation. He suggests several approximations that require less information and reduce the computational load. First, he decouples the alliance problem into separate single-airline problems and links them through the exchange of current bid prices. In order to determine the correct bid prices in each time frame, it is necessary to iteratively solve the revenue management systems of the participating carriers. The performance of the decoupled problem is close to the complete information case. Second, he applies the same bid price exchange logic to other frequently used revenue management algorithms such as DAVN, and finds that they perform similarly well.

Following Wright's idea, Jain (2011) introduces an approximation using the bid price from the previous time period. His approach is easier to implement in practice and the iterative execution of the revenue management systems, as proposed by Wright, becomes redundant. Jain refers to his policy as *dynamic valuation* and shows with an extensive simulation study that it performs better than other valuation schemes. In his large-scale sample network, total alliance revenue increases between 0.3 and 0.5 percent.

Topaloglu (2012) derives central optimal code-share valuations from the deterministic linear programming formulation mentioned in Simpson (1989) and Williamson (1992). He decomposes the central problem by the airlines and relaxes the constraints linking the resulting sub-problems – the constraints ensuring that each carrier allocates the same capacity to a particular code-share itinerary. The dual multipliers associated with the relaxed constraints are used as code-share valuations in the optimizers of the individual airlines. Topaloglu shows that these valuations implement the central solution as long as the carriers do not re-optimize their inventory. He provides a numerical analysis indicating a performance gap of two to four percent between the central solution and the local problems with the dual multipliers.

Netessine and Shumsky (2005) develop a static revenue management game model with complete information and study the competition on complementary flights. They define conditions under which a pure strategy Nash equilibrium exists, show how code-sharing leads to revenue losses and discuss revenue sharing schemes that coordinate alliance behavior. Graf and Kimms (2001) present an option-based approach to the alliance problem. For a single flight and two airlines they iteratively adapt booking limits and use simulation-based optimization to evaluate the options. In this context, options are the right but not the obligation for the marketing carrier to sell a particular seat. If it exercises the option, it has to pay a predetermined strike price to the operating carrier.

The previously described models solely consider the operational level of the alliance revenue management problem. The impact of revenue sharing has been neglected so far. Hu et al. (2013) are the first ones who examine both levels simultaneously. They propose a two-stage hierarchical approach that incorporates the revenue sharing scheme from the contractual phase in the revenue management decisions on the operational level. Using the deterministic linear programming formulation, Hu et al. show that some static revenue sharing rule exists that implements the central solution in the decentralized systems and that provides a revenue allocation in the core of the alliance game. However, the proposed scheme uses the central optimal bid prices and therefore, cannot be implemented without solving the central problem. To circumvent this practical limitation, the authors suggest a heuristic policy based on publicly available fares. In their numerical experiments, this heuristic leads to an optimality gap in the expected revenue of 0.35 percent.

Çetiner and Kimms (2012) model the revenue allocation problem as a cooperative game. They calculate the contribution of each airline by evaluating the deterministic linear program over all possible coalitions. The nucleolus values of this game are used as benchmarks for a fair revenue distribution among the alliance members. Further, they show that the alliance game has a non-empty core by proving that it is balanced and applying the Bondareva-Shapley theorem (Bondareva, 1963; Shapley, 1967).

3.3.2. Related Research

The approaches discussed in the following paragraphs have fundamental differences to the revenue management problem faced by airline alliances. Nevertheless, they provide valuable insights into the management and coordination of such partnerships. Our overview includes various theoretical results as well as practical examples from other industries, in which similar questions have been studied more intensively than in the context of revenue management and airline alliances.

Cargo Industry

The coordination of liner shipping and cargo alliances is discussed in Agarwal and Ergun (2010) as well as Houghtalen et al. (2011). Their analyses focus on the network design and capacity allocation problem, and they use cooperative game theory to model the interaction of alliance members. Both contributions develop mechanisms that provide incentives in form of side payments to encourage collaborative behavior. In contrast to the passenger airline revenue management problem, cargo demand realizes a priori and capacity allocations are done on the contractual level. Moreover, we note that the network design problem corrresponds to the selection of code-share flights in airline alliances. Ideally, the airlines find the set of flights that maximizes their additional revenue and minimizes the cannibalization effect on other flights in the network. This problem is not part of the revenue management process[9] and has been addressed separately in several papers such as LaRoche et al. (2010), Abdelghany et al. (2009), O'Neal et al. (2007), Srinivasan (2004) and Sivakumar (2003). The authors of these papers propose various optimization approaches, but in contrast to Agarwal and Ergun as well as Houghtalen et al., they do not incorporate game-theoretic concepts. For a general discussion on the differences between the cargo and the passenger revenue management problem, we refer the reader to Kasilingam (1996).

Competition

Game theory is used in many competitive situations to model the behavior of different players. Among others, Isler and Imhof (2008) examine the existence of Nash equilibria in competitive airline markets. They show that revenues spiral down when capacities increase and carriers use dynamic pricing. The authors conclude that airlines should focus on long-term strategies to avoid this behavior. Netessine and Shumsky (2005) study competition between two parallel flights operated by different airlines. They model the demand flows between both flights and provide conditions for a pure-strategy Nash equilibrium under the assumption of complete information. Gallego and Hu (2009) present a game-theoretic dynamic pricing formulation that includes competitor's offers in the customer choice set and show that also in this situation a Nash equilibrium exists. Zhang and Kallesen (2008) develop a revenue management approach that explicitly considers competitor's price information and demonstrate in a simulation study that their method outperforms other approaches. Gorin and Belobaba (2004) use simulations to analyze the effects of market entry on revenue management.

Although the research on competition utilizes similar ideas, the underlying assumptions are different to the ones in collaborative partnerships. First of all, competition occurs on parallel flights, while alliances focus on complementary flights (Section 2.1.3).

[9] In the classical revenue management problem, the network, resources and capacities are assumed to be given and fixed.

Second, competitors act in a non-cooperative way. In an alliance, although the carriers remain independent entities, they are assumed to act cooperatively and to be willing to collaborate within the framework of the alliance. Third, in cooperative situations partners are likely to exchange information in order to jointly manage their businesses. On the contrary, competing carriers do not share any information.

Supply Chain

The stream of supply chain research discusses how independent partners manage their inventory under various supply chain contracts. There is a substantial amount of literature devoted to this topic and in particular the coordination of inventory decisions across multiple manufacturers and retailers. Arshinder et al. (2008) survey the supply chain literature, while Cachon and Netessine (2004) provide an overview of the application of cooperative and non-cooperative game theory in static and dynamic supply chains. The use of revenue sharing contracts in supply chain coordination is presented in Chen and Wu (2008), Cachon and Lariviere (2005) and Giannoccaro and Pontrandolfo (2004). These studies compare the performance of revenue sharing schemes to other prevalent supply chain contracts and discuss their strengths and weaknesses. A general comparison between centralized and decentralized production planning in multi-agent supply chains is presented in Saharidis et al. (2006).

The main difference between supply chains and airline alliances is the relationship between supplier and assembler. In the supply chain literature, they are assumed to be distinct entities. In airline alliances, however, every airline operates flights and sells code-share itineraries. Thereby, it acts as supplier (operating carrier) and assembler (marketing carrier) at the same time. Furthermore, airline seats are perishable and may be part of various itineraries. Supply chains typically represent the downward flow of products from the producer to a specific group of customers and also provide the option to store products whenever there is no demand.

Telecommunications

The growth of the telecommunications industry as well as the Internet has led to a large body of research in this field. The central questions concern the routing and pricing of data streams in constrained networks owned by several providers. Orda et al. (1993) investigate the existence of a unique Nash equilibrium in telecommunications networks with selfish users who make independent routing decisions. A survey on networking games in telecommunication networks is provided in Altman et al. (2006). They summarize various equilibrium concepts from other areas such as transportation planning, and analyze their applicability to the telecommunications market. In a different paper, Altman et al. (2001) use dynamic flow adjustments to coordinate traffic in a data network and show that their policy converges to a Nash equilibrium. Boulogne et al. (2002) apply a similar convergence idea to distributed computing, while Zhang and Douligeris (1992) study the convergence of greedy algorithms in telecommunications networks from a more general perspective.

Telecommunication companies and airline alliances face similar challenges regarding the utilization of limited resources and the partitioned network ownership. However, routing and pricing decisions, which are a major aspect in the above mentioned literature, are not part of the classical revenue management process and distinguish the stream of telecommunications research from the problem studied in this thesis.

Bargaining and Incentive Design

In contrast to predefined revenue sharing schemes, bargaining enables the players to mutually agree on the revenue distribution. Chatterjee and Lilien (1984) survey several two player bargaining methods and evaluate them in terms of efficiency, while Chatterjee and Samuelson (1983) focus on the incomplete information case where buyers and sellers make sealed offers. Nagarajan and Bassok (2008) provide a bargaining framework in which a single assembler purchases complementary items from several suppliers. The

Nash bargaining concept (see Nash, 1950) is used in Waslander et al. (2003) to coordinate multi-agent systems using decentralized optimization. The application of bargaining to revenue management is presented in Bhandari and Secomandi (2011).

Incentives (side payments and penalties) are provided to players in order to make them comply with the central objectives. In decentralized systems, the goal of an individual player may jeopardize the performance of the collective and additional incentives are necessary to avoid sub-optimal decisions. Kim (1991) develops a mechanism that explicitly values the actions of all players in a multi-agent system. They associate a specific reward or penalty with each possible action and the players are expected to take rational decisions – decisions that maximize their individual payoff. A general view on incentive design and decentralized control in divisionalized firms is provided in Groves and Loeb (1979) as well as Shubik (1962). Balachandran and Ronen (1989) suggest incentive contracts for subcontracted production. More related research is found in Ho et al. (1982) and Ronen and Balachandran (1988), who study incentive design from a control-theoretic perspective and under uncertainty.

Bargaining as well as incentive design provide two feasible instruments to coordinate alliances. Approaches from both fields have not yet been applied in the context of revenue management and airline alliances. Only Hu et al. (2013) mention that their optimal static proration scheme corresponds to the Nash bargaining solution. Agarwal and Ergun (2010) and Houghtalen et al. (2011) use incentives to coordinate cargo alliances.

The previous paragraphs, together with the findings reported in Section 3.3.1 on the specific code-share revenue management problem, provide a comprehensive overview of the relevant literature and other major results from related subjects and industries. With this knowledge, we continue with the description of potential research gaps.

4. Research Gap

Chapters 2 and 3 introduced the reader to airline alliances, revenue management as well as the state-of-the-art literature on alliance code-share management. Although a substantial amount of research aims at these problems, specific work on alliance revenue management is rather limited. In the following, we point out gaps in the published literature and discuss potential research opportunities.

Two main conditions characterize the alliance revenue management problem faced by airlines in practice: (1) Alliance partners operate separate revenue management systems, and (2) information exchange is limited. Both conditions differentiate alliance revenue management from the classical case − the well-studied single-airline problem. While only a few studies explicitly model this situation, most theoretical work solves the central problem or assumes that the airlines possess private information about their partners. Either approach might promise superior results or more sophisticated coordination schemes, but is unrealistic to implement in practice. In fact, by violating any of the two conditions above, the airlines would overcome their biggest practical burdens, providing them the opportunity to completely merge their revenue management processes with no need for decentralization and coordination schemes.

A few authors acknowledge these limitations and propose heuristic schemes that approximate the optimal policies. Although being closer to real-world applications, there are several other aspects that need to be considered. Among others, they are (1) the stochastic realization of customer demand, (2) the closed loop in the revenue management process consisting of forecasting, optimization and history building, and (3) the timing of the airlines' actions. Most contributions do not account for such constraints: They assume a deterministic environment or solely model the optimization process without observing the feedback effect from history building and forecast updating. Timing problems arise when airlines share current system information or iteratively execute their revenue management systems.

The level of information sharing also impacts the trust and commitment among the alliance partners as well as the effort related to the implementation of alliance interfaces. From an industry point of view, low information sharing is faster to install and easier to monitor; higher information sharing promises superior results. While most research tends to use more information than typically available, optimal results are ideally realized with minimal interaction among the partners. Such an approach would be particularly interesting on the operational level, where frequent information exchange is necessary to determine optimal availabilities and to keep the systems in equilibrium.

As discussed in the previous chapter, code-share fares are prorated on the contractual as well as the operational level. In the published literature, most authors use a single scheme to solve both levels, i.e. the revenue sharing scheme is also used for the optimizer valuations. Although both levels are correlated, studying them independently may provide an additional advantage: The schemes can be customized to better meet the primary objectives of the respective phases – the optimal control of resources and the fair distribution of realized revenues. First, coordinate the decisions on the operational level in order to maximize the joint performance. Then, distribute the realized revenues among the airlines such that the alliance is stable and all members willing to cooperate.

We also note that most contributions examine one particular revenue sharing or valuation policy. The respective scheme is usually motivated from practice or has been derived theoretically. Additional insights about the structure of valuation schemes may be gained by establishing general conditions. This would provide a better understanding on how to choose proration rates. Moreover, all valuations satisfying these conditions implement the desired solution and thereby, they offer more flexibility than a single, fixed scheme.

Finally, large-scale stochastic simulations provide a good framework to verify the effectiveness of different approaches under real-world circumstances. They are the best way to capture the full dynamics of the airline market and to investigate the performance of alliance revenue management methods in practice. However, only a few contributions in the context of alliance revenue management make use of simulations. Instead, most analyses are limited to smaller numerical examples based on simplified network and product structures.

From the research gaps described in the previous paragraphs, we derive the following cornerstones for the approach studied in this thesis.

Decentralization: In a decentralized setting, each partner takes its own decisions and only controls its own inventory. Such an environment represents current industry practice and is suitable to study the interaction of revenue management decisions as well as the impact of decentralized code-share control on the local sub-networks.

Optimality Conditions: Optimality conditions state requirements that must be satisfied in order to implement an optimal solution. Finding such conditions for the central alliance revenue management problem allow to evaluate the local outcomes of the independent airlines without actually knowing the central solution. At the same time, optimality conditions serve as guidelines according to which proration schemes can be designed.

Adaptive Updating: Whenever the local solutions do not satisfy such conditions, updating the optimization parameters associated with the code-share itineraries may improve the overall result. As opposed to static schemes, adaptive updates change over time and depending on the currently available inventory. In addition to many other dynamic schemes, adaptation rules should be structured such that the local solutions converge to the central optimum.

Stochastic Simulations: Compared to deterministic experiments in a laboratory setting, stochastic simulations allow to construct more dynamic and closer-to-the-real-world settings. Sophisticated simulation environments model the interaction between customers and airlines along the complete revenue management process. They provide a controllable and isolated environment in which new methodologies can be tested without being affected by external influences, such as changing market conditions, marketing activities or competitor actions. Furthermore, they allow to model demand fluctuations and other shocks to measure sensitivity as well as to study long-term behavior. Yet, simulations are transparent in the sense that all data is a priori known and any effects that are observed can be related back to the input.

Part II.

Theoretical Approach to Code-Share Optimization

5. Code-Share Revenue Management Model

This chapter derives the code-share revenue management model. To formally depict the problem, we begin with the notations used throughout this thesis. Afterwards, we define the central model, which optimizes the entire alliance as if it was a single airline. Finally, we derive the decentralized model from the centralized one. We decompose it by the airlines and explicitly consider the itineraries connecting the sub-problems – the code-share itineraries. In the final section of this chapter, we identify assumptions and limitations underlying the model.

5.1. Centralized Model

The central model contains m flights (resources) and n itineraries[10] (products) with M and N being the sets of all flights and itineraries, respectively. We label the flights by index $i = 1, \ldots, m$ and itineraries by index $j = 1, \ldots, n$. Further, let $\mathbf{A}$ be an incidence matrix with m rows and n columns. The column vector $\mathbf{A}_j$ denotes the flights used by itinerary j (the routing of j). The row vector $\mathbf{A}_i$ denotes the itineraries traversing leg i. For every matrix element a_{ij} it holds that

$$a_{ij} = \begin{cases} 1 & \text{if itinerary } j \text{ uses flight } i \\ 0 & \text{otherwise} \end{cases}.$$

[10] An itinerary is described by the origin, the destination, the routing as well as the booking class.

The current state of the inventory is represented by the number of free seats on the m flights. It is denoted by vector $\mathbf{x} = (x_i : i \in M)$. To sell an itinerary there must be at least one empty seat on each flight that is used. If a booking is done, the inventory is reduced by $\mathbf{A}_j$, so the new available capacity becomes $\mathbf{x} - \mathbf{A}_j$. Note that in this version of the network control problem as well as throughout this thesis, we only consider individual bookings; batch bookings are not possible.

The booking horizon is divided into T discrete time frames. The current time frame is denoted by t and decreases to $t = 0$, the time of departure. The time intervals are chosen sufficiently small such that the probability to receive more than one request per time interval is negligible. We denote the probability that a request for itinerary j arrives in time period t by $q_j^t \geq 0$. The probability that no request arrives is q_0^t. The n-dimensional vector $\mathbf{r} = (r_j : j \in N)$ contains the fares associated with the n itineraries.

We describe the structure of the optimal solution in Section 5.1.1 and introduce a decomposition approach in Section 5.1.2.

5.1.1. Network Dynamic Program

Talluri and van Ryzin (1998) propose a dynamic programming formulation to calculate controls $\mathbf{u}(t, \mathbf{x}) = (u_j\,(t, \mathbf{x}) : j \in N)$ for the network revenue management problem under stochastic demand. It is based on the value function $V^t(\mathbf{x})$, which determines the maximum expected revenue at any inventory level $\mathbf{x}$ and any point of time t along the booking horizon. $V^t(\mathbf{x})$ can be computed using the Bellman equation

$$\begin{aligned} V^t(\mathbf{x}) =& q_0^t V^{t-1}(\mathbf{x}) + \sum_{j \in N} q_j^t E\left[r_j u_j(t, \mathbf{x}) + V^{t-1}\left(\mathbf{x} - \mathbf{A}_j u_j(t, \mathbf{x})\right)\right] \\ & \text{where } u_j(t, \mathbf{x}) = \arg\max\nolimits_{u \in \{0,1\}} \left\{ r_j u + V^{t-1}(\mathbf{x} - \mathbf{A}_j u) \right\} \end{aligned}$$

with boundary conditions

$$V^0(\mathbf{x}) = 0 \quad \forall\, \mathbf{x} \qquad \text{and} \qquad V^t(\mathbf{0}) = 0 \quad \forall\, t \ .$$

Talluri and van Ryzin prove that the optimal controls to this problem are of the form

$$u_j^*(t,\mathbf{x}) = \begin{cases} 1 & \text{iff } r_j \geq V^{t-1}(\mathbf{x}) - V^{t-1}(\mathbf{x} - \mathbf{A}_j) \text{ and } \mathbf{A}_j \leq \mathbf{x} \\ 0 & \text{otherwise} \end{cases} \ .$$

A request is accepted if and only if there is enough capacity available and the value of the request exceeds the opportunity cost of the required resources. The opportunity cost or bid price is the difference between the value function at the next point in time with and without the additional seat.

5.1.2. Dynamic Programming Decomposition

Because the number of state variables of the network dynamic program increases exponentially in the number of flights, this problem is difficult to solve in its exact form and becomes intractable even for small networks. In this section, we describe a common approximation called Dynamic Programming Decomposition – see Section 2.2.3 as well as Zhang (2011) and Talluri and van Ryzin (2004). It consists of two steps: First, the deterministic linear program described in Williamson (1992) calculates the optimal passenger flow across all flights. This step captures the network effect stemming from the interrelation of various demand flows traversing the same set of flights. Second, m independent single-flight dynamic programs, similar to the one described above, determine the leg bid prices used for inventory control. This step captures the stochastic realization of customer demand over time. The decomposition approach provides good results in real-life networks and is frequently implemented in the industry. It will be used as base method throughout the rest of this thesis.

In order to formally describe the linear program, let y_j denote the capacity assigned to itinerary j and let D_j be the aggregated mean demand-to-come for j. Then the network linear program can be formulated as in (5.1).

$$\hat{Z} = \max \sum_{j \in N} r_j y_j \tag{5.1a}$$

$$\text{subject to} \sum_{j \in N} a_{ij} y_j \leq x_i \qquad \forall\, i \in M \tag{5.1b}$$

$$0 \leq y_j \leq D_j \qquad \forall\, j \in N \tag{5.1c}$$

The objective of (5.1) is to find capacity allocations y_j such that the total revenue is maximized. The first set of constraints ensures that the capacity of each flight (the number of currently available seats) is not exceeded. The second set of constraints restricts the capacity allocations to be non-negative and to be at most the mean demand-to-come for the respective itinerary. The (primal) solution to this problem are optimal capacity allocations y_j^*. The dual solution to the first set of constraints is a m-dimensional vector of shadow prices, say $\boldsymbol{\lambda} = (\lambda_i : i \in M)$. λ_i is also referred to as the deterministic bid price of flight i and represents the marginal value of an additional seat.

The shadow prices are used to prorate the revenue of multi-leg itineraries. The revenue contribution of itinerary j to flight i is calculated as the total itinerary fare r_j minus the sum of the shadow prices on the other legs:

$$\bar{r}_{ij} = r_j - \sum_{\substack{l \in \mathbf{A}_j \\ l \neq i}} \lambda_l \ . \tag{5.2}$$

The $\bar{r}_{ij}$ is called the displacement adjusted revenue of itinerary j on flight i and replaces the full itinerary fare in the single-flight dynamic programs.

$$
\begin{aligned}
V^t(\mathbf{x}) &\approx \sum_{i\in M} V_i^t(x_i) \\
&= \sum_{i\in M}\left[q_0^t V_i^{t-1}(x_i) + \sum_{j\in N} q_j^t E\left[\bar{r}_{ij}u_{ij}(t,x_i) + V_i^{t-1}\left(x_i - \mathbf{A}_j u_{ij}(t,x_i)\right)\right]\right] \qquad (5.3)\\
&\text{where } u_{ij}(t,x_i) = \arg\max\nolimits_{u\in\{0,1\}}\left\{\bar{r}_{ij}u + V_i^{t-1}(x_i - \mathbf{A}_j u)\right\}\\
&\text{and } V^0(x_i) = 0 \qquad \forall\, x_i \ .
\end{aligned}
$$

The leg bid prices, $\mathbf{b} = (b_i : i \in M)$, to (5.3) are of the form

$$
b_i = V_i^t(x_i) - V_i^t(x_i - 1)
$$

for seat index x_i on flight i at time t. A request is accepted if its value exceeds the sum of the currently posted bid prices on the flights that it traverses, i.e.

$$
r_j \geq \sum_{i\in \mathbf{A}_j} b_i \ . \qquad (5.4)
$$

The centralized model introduced so far represents the alliance as if it was a single airline. It serves as an upper bound on the performance of the decentralized alliance model derived in upcoming section.

5.2. Decentralized Model

Let there be c airlines with C being the set of all carriers. We index the carriers by $k = 1, \ldots, c$ and assume that the entire alliance consists of all airlines in C. Together they operate m flights and sell n itineraries such that the central problem corresponds exactly to the formulations in Section 5.1.2. Let C_j denote the set of carriers operating the flights along itinerary j. For intraline itineraries C_j has cardinality one; for code-share itineraries $|C_j|$ is larger than one.

In the decentralized setting, the set of flights operated by carrier k is denoted M^k. We assume that each flight is operated by exactly one carrier such that the set M^k forms a partition of M and $\bigcup_{k\in C} M^k \equiv M$. The set of itineraries N is divided into itineraries completely operated by a single carrier k – sets N^k – and itineraries partially operated by k denoted N^k_{CS}. The former are intraline itineraries, the latter code-share itineraries. The superscript k denotes the marketing carrier of the respective itineraries. For code-share itineraries these are all carriers operating at least one flight from the itinerary. On intraline itineraries the operating carrier is the only marketing carrier. There are no virtual code-shares and $\bigcup_{k\in C} N^k + N_{CS} \equiv N$ where $N_{CS} = \bigcap_{k\in C} N^k_{CS}$. Following the same pattern, we also partition matrix $\mathbf{A}$ by the carriers.

$$\mathbf{A} = \begin{array}{c} \\ M^1 \\ \vdots \\ M^c \end{array} \begin{array}{c} \begin{array}{cccc} N^1 & \cdots & N^c & N_{CS} \end{array} \\ \left[\begin{array}{c:c:c} \mathbf{A}^1 & \mathbf{0} & \mathbf{A}^1_{CS} \\ \hdashline \mathbf{0} & \mathbf{A}^c & \mathbf{A}^c_{CS} \end{array}\right] \end{array}$$

In the first c columns, the sub-matrices represent the airline's intraline networks. The respective itineraries use flights of a single airline and the other sub-matrices are all zero. The third column covers the code-share itineraries where $\mathbf{A}^1_{CS}$ to $\mathbf{A}^c_{CS}$ mark the code-share flights of the respective airline.

Because every flight is operated by exactly one airline, the m dynamic programs naturally decompose by the airlines and the alliance effect is captured in the linear programs. In the following, we discuss the integration of the code-share itineraries in the linear program. The modification of the flight dynamic programs is straightforward and presented at the end. In order to distinguish the central and the local problems, we use the superscript k on all decision variables associated with the local problems of the individual carriers. Variables without superscript k refer to the central problem.

We begin with the local deterministic linear program of carrier k without code-share itineraries. It covers solely the flights M^k and the intraline itineraries N^k:

$$
\begin{aligned}
Z^k \quad &= \quad \max \sum_{j \in N^k} r_j^k y_j^k && && (5.5a)\\
&\text{subject to} \sum_{j \in N^k} a_{ij}^k y_j^k \leq x_i^k && \forall\, i \in M^k && (5.5b)\\
&\qquad 0 \leq y_j^k \leq D_j^k && \forall\, j \in N^k && (5.5c)
\end{aligned}
$$

In the next step, we add carrier k's code-share itineraries to formulation (5.5). The resulting linear program is (5.6).

$$
\begin{aligned}
Z_{CS}^k \quad &= \quad \max \sum_{j \in N^k} r_j^k y_j^k + \sum_{j \in N_{CS}^k} \tilde{r}_j^k y_j^k && && (5.6a)\\
&\text{subject to} \sum_{j \in N^k} a_{ij}^k y_j^k + \sum_{j \in N_{CS}^k} a_{ij}^k y_j^k \leq x_i^k && \forall\, i \in M^k && (5.6b)\\
&\qquad y_j^k \leq D_j^k && \forall\, j \in N^k \cup N_{CS}^k && (5.6c)\\
&\qquad y_j^k \geq 0 && \forall\, j \in N^k \cup N_{CS}^k && (5.6d)
\end{aligned}
$$

The two summations in the objective function (5.6a) are the aggregate revenue generated from intraline and code-share itineraries respectively, where $\tilde{r}_j^k$ denotes the prorated revenue from itinerary j allocated to carrier k. Since intraline and code-share itineraries traverse the same flights, they are both included in the capacity constraints (5.6b). Finally, the upper and lower bound constraints (5.6c) and (5.6d) limit the capacity allocations.

The fare adjustment is done as before. The revenues of multi-leg itineraries are adjusted by (5.2). The valuations of single-flight itineraries remain unchanged. This holds for code-share itineraries as well, but carriers may additionally exchange their shadow prices to accurately imitate the central process.

Given the adjusted revenues, the dynamic program (5.3) is divided by the carriers as shown in (5.7).

$$V_i^{t,k}(\mathbf{x}) = \sum_{i \in M^k} \Big[q_0^{tk} V_i^{t-1,k}(x_i) + \\ \sum_{j \in N^k \cup N_{CS}^k} q_j^{tk} E \Big[\bar{r}_{ij} u_{ij}^k(t, x_i) + V_i^{t-1,k} \Big(x_i - \mathbf{A}_j^k u_{ij}^k(t, x_i) \Big) \Big] \Big] \quad (5.7)$$

The acceptance decision (5.4) remains the same for intraline itineraries. The availability of code-share itineraries is determined via the availability exchange methods − the carriers either exchange their local booking class availabilities (AVS) or the respective flight bid prices (BPS). Both methods are explained in Section 3.2.3.

5.3. Assumptions and Limitations

The motivation for the above mentioned model stems from current industry practice. Most airlines operate under free sale agreements and exercise full control over their own resources. In the process, code-share itineraries are broken apart and controlled by the separate operating carriers, as reflected in our model. Alternative hard- or soft-blocking agreements would provide control over the entire itinerary and diminish the proration of code-share fares. However, the carriers need to add the flights of their partners to their own optimization and equilibrium is achieved by reallocating seats on shared flights as described in Boyd (1998) and Vinod (2005). Both is hardly done in practice.

Our approach decomposes the central network linear program into independent subproblems. Decomposition ideas date back to the early work by Dantzig and Wolfe (1960) and they are frequently used to solve complex problems, as for example the network revenue management problem (see Section 2.2.3). In the context of airline alliances, Topaloglu (2012) as well as Hu et al. (2013) apply decomposition to divide the central problem by the airlines – the same procedure as used in this thesis. We also note that the incidence matrix **A** has the block-angular structure associated with Dantzig-Wolfe decompositions. The intraline itineraries are independent and belong to a particular carrier. The code-share itineraries link these sub-problems. In contrast to the Dantzig-Wolfe solution method, we do not solve the sub-problems to iteratively improve the restricted master problem containing the dependent constraints. Instead we divide the dependent constraints over the airlines such that the entire network decomposes into independent sub-problems and there is no additional master problem.

Choosing the linear program for our analysis is justified by two reasons: The main advantage is its mathematical tractability as well as its simple structure. The network revenue management problem grows quickly with the size of the network and is already difficult to solve for an individual airline, not to mention alliances with multiple members. The second reason is the frequent usage of the linear program in practice. It captures the full complexity of the network and the dual solution provides good estimates of the network value of the individual flights. Various decomposition approaches use these values to prorate multi-leg fares and then apply flight optimization methods to determine the actual steering parameters. As a consequence to the frequent use of decomposition approaches, we focus on finding the central optimal shadow prices, but not the central optimal capacity allocations. The primal solution is usually not used due to the inefficiencies of static booking limits.

We derive the decentralized code-share model by dividing the central one over the airlines. This ensures the comparability between the central and the alliance solution. Furthermore, we use the central approach during the derivations of the adaptation algorithm and as a benchmark for the performance of the alliance. A disadvantage of this approach is that it limits the airlines to the same revenue management system and the same product structure. Otherwise the central solution would not be a valid upper bound and could not be used to verify the performance of the alliance.

In the next paragraphs, we discuss three assumptions regarding our model and the code-share process. First, we note that our model restricts control decisions to flights operated by the respective carrier, its intraline itineraries as well as the code-share itineraries traversing its network. The carriers do neither possess information about the current inventory of their partners (except what is exchanged related to the code-share process) nor their local demand flows or their local fares. This information remains private and is not shared.

Assumption 1 (Information Disclosure)**.** *Information about capacities, intraline fares and intraline demand is private to the operating carrier.*

As argued in Chapter 4, we believe that a central limitation in alliance revenue management is the lack of information. Exchanging all private information requires far reaching immunity, but also raises the question of why not implementing a central revenue management system instead of transferring large amounts of data. Moreover, heterogeneous revenue management systems could make this data inconsistent and additional processing steps may be required.

With respect to code-sharing we assume that every carrier has the appropriate revenue management capabilities to separately handle code-share itineraries as well as to collect and store the relevant data. Using these capabilities, all operating carriers can observe code-share demand and determine unique forecasts and availabilities. We assume further that by using the same revenue management methodology and the same product structure, the carriers calculate the same expected code-share demand.

Assumption 2 (Code-Share Demand)**.** *All operating carriers observe and forecast the same code-share demand for a common itinerary.*

From a practical perspective, having the same forecasts is a strong assumption. Besides a common revenue management methodology and the same product structure, airlines must agree on the same history building rules and the coordination of manual forecast adjustments. Alternatively, the airlines could regularly align their forecasts or appoint one carrier to maintain joint forecasts. In either case, the carriers need to ensure data consistency and the information exchange increases.

In our model, code-share control is based on the local optimization results of the operating carriers. As they do not possess information about the capacities and intraline demands of alliance members, we chose the prorated code-share valuations $\tilde{r}_j^k$ to coordinate their decisions. In the subsequent chapter, we discuss how such valuations should be selected in order to implement the central optimal solution in the local systems. To be consistent with the central problem, we assume that the prorated valuations must always add up to the full itinerary fare. We define $\tilde{R}$ to be the set of all code-share valuations satisfying this property. We assume further that all operating carriers have knowledge about the full itinerary fare as well as the prorated valuations $\tilde{r}_j^k$ of all carriers $k \in C_j$. This is summarized in Assumption 3.

Assumption 3 (Code-Share Valuations)**.** *The code-share valuations $\tilde{r}_j^k$ are known to all carriers $k \in C_j$ and must add up to the full itinerary fare r_j:*

$$\tilde{R} = \left\{ \tilde{r}_j^k : \sum_{k \in C_j} \tilde{r}_j^k = r_j \ \textit{for all} \ j \in N_{CS} \right\}$$

Last, we note that we do not make an explicit assumption about the revenue sharing scheme. Our analysis considers the aggregate result of the entire alliance. We do not analyze the performance of the individual airlines and any revenue sharing scheme may be used, if necessary.

6. Adaptation Algorithm for Code-Share Valuations

Ideally, the alliance game is in equilibrium at all stages. However, the airlines are neither in equilibrium right from the beginning, nor can any airline determine the equilibrium individually because it depends on private information of the partners. Moreover, centrally computing the equilibrium is also not possible due to legal, technical and organizational reasons explained in Section 1.2. Nevertheless, we assume that the carriers have a certain willingness to collaborate and that they correct their decisions whenever these are not optimal. In this chapter, we use this idea to develop the concept of adaptive code-share valuations. Compared to static valuation schemes, adaptation changes the valuations over time based on the current state of the system. Compared to other dynamic schemes, the decentralized systems converge to the solution of the central planner.

This chapter introduces the adaptation algorithm in Section 6.1 and derives necessary as well as sufficient conditions for central optimality in Section 6.3. Thereafter, Section 6.4 develops adaptation rules that converge to valuations satisfying these conditions. Their advantages and shortcomings are assessed in Section 6.5. The performance of the algorithm is demonstrated with small numerical experiments in the following chapter.

6.1. Structure of the Algorithm

Before we derive the individual steps of the adaptation process, we briefly describe the general structure of the algorithm. It starts by initializing the prorated code-share valuations $\tilde{r}_j^k \in \tilde{R}$ in the linear programs (5.6) defined in Section 5.1. Given the initial valuations, the linear programs are solved and the iterative process begins. An iteration consists of two steps: updating the valuations and solving the linear programs. The updating adapts the code-share valuations one by one and independently of each other based on the update rules derived in Section 6.4. Afterwards the linear programs are solved to update the solutions given the new valuations.

In the process, we distinguish between exact and approximate updates. Exact updates require to solve the linear programs whenever an itinerary has been updated. For approximate updates the linear programs are only solved once per iteration − after all code-share itineraries have been updated. The latter approximates the former because the linear programming solutions may change with every update and therefore, subsequent updates may not be accurate. We further discuss the potential problems with both variants in the final section of this chapter. Until then and in particular throughout our theoretical analysis, we assume that updates are exact, i.e. based on the currently valid linear programming solution.

When all code-share itineraries and the linear programming solutions have been updated, the current local solutions are checked for central optimality. For this purpose we derive necessary and sufficient conditions in Section 6.3. They are in the form of constraints on the prorated valuations $\tilde{r}_j^k$ as well as the decentralized capacity allocations y_j^k. In order to be central optimal, the local solutions must satisfy these optimality conditions for all code-share itineraries simultaneously. Once this is the case, the structure of the update rules ensures that the local solutions continue to satisfy the optimality conditions in all subsequent iterations.

The adaptation process stops when all updates – for all carriers and all code-share itineraries – are zero. With no changes in the valuations, the linear programs provide the same results and the algorithm has reached a stable solution. The termination criterion is considered separately from the optimality criterion because they may be satisfied at different times. Once the optimality criterion holds, it usually takes an additional iteration until the convergence criterion is satisfied as well. Alternatively, it may happen that the algorithm converges to a stable solution and the process terminates, but the optimality conditions are not satisfied. The structure of the complete algorithm is schematically depicted in Figure 6.1.

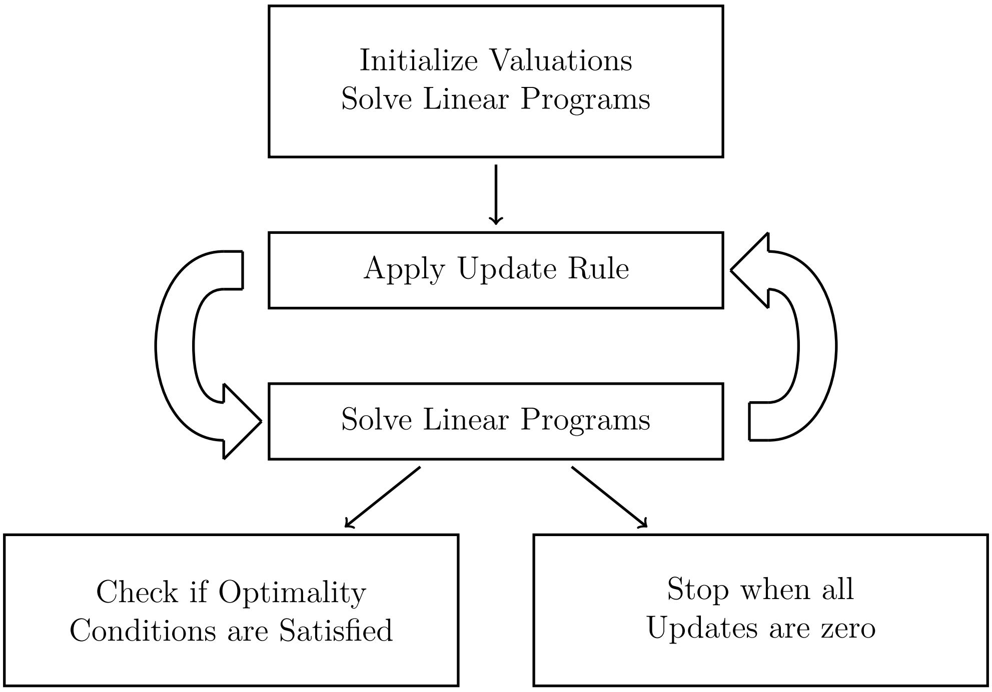

Figure 6.1.: Adaptation Process

In preparation of the theoretical derivations of the adaptation process, we start with introducing some useful results needed throughout our discussion.

6.2. Prerequisites

This section describes four fundamental principles. With regard to linear programming, we present necessary conditions for optimality and define degeneracy in view of decentralized code-share control. In the following, we propose two propositions that connect the prorated code-share valuations to the central problem. Our discussion begins with the optimality conditions.

An useful mathematical concept to describe optimality are the Karush-Kuhn-Tucker (KKT) conditions. They generalize the idea of Lagrange multipliers to linear and nonlinear programs with inequality constraints (Bertsekas, 1999; Bertsimas and Tsitsiklis, 1997). There are four necessary conditions that must hold in order for a solution to be optimal. These are primal feasibility, dual feasibility, complementary slackness and stationarity. In order to formally define these conditions, let $\boldsymbol{\lambda}^k = \left(\lambda_i^k : i \in M^k\right)$, $\boldsymbol{\rho}^k = \left(\rho_j^k : j \in N^k \cup N_{CS}^k\right)$ and $\boldsymbol{\omega}^k = \left(\omega_j^k : j \in N^k \cup N_{CS}^k\right)$ be the dual variables (shadow prices) to the capacity constraints (5.6b) and the flow constraints (5.6c) and (5.6d) of the local linear program (5.6) of airline k. As before, λ_i^k measures the value of an additional seat on flight i. The dual variables ρ_j^k and ω_j^k measure the value of an additional passenger on itinerary j, which corresponds to the change in the objective function when the right hand side of the constraint marginally changes. Having defined the dual variables, the KKT conditions for the linear program (5.6) of carrier k are:

$$\text{Primal Feasibility:} \quad \sum_{j \in N^k \cup N^k_{CS}} a^k_{ij} y^k_j \leq x^k_i \qquad \left(i \in M^k\right)$$

$$0 \leq y^k_j \leq D^k_j \qquad \left(j \in N^k \cup N^k_{CS}\right)$$

$$\text{Dual Feasibility:} \quad \lambda^k_i \geq 0 \qquad \left(i \in M^k\right)$$

$$\rho^k_j,\, \omega^k_j \geq 0 \qquad \left(j \in N^k \cup N^k_{CS}\right)$$

$$\text{Complementary Slackness:} \quad \lambda^k_i \left(\sum_{j \in N^k \cup N^k_{CS}} a_{ij} y^k_j - x^k_i \right) = 0 \qquad \left(i \in M^k\right)$$

$$\rho^k_j \left(y^k_j - D^k_j\right) = 0 \ ; \ \omega^k_j y^k_j = 0 \qquad \left(j \in N^k \cup N^k_{CS}\right)$$

$$\text{Stationarity:} \quad r^k_j - \sum_{i \in A^k_j} \lambda^k_i - \rho^k_j + \omega^k_j = 0 \qquad \left(j \in N^k\right)$$

$$\tilde{r}^k_j - \sum_{i \in A^k_j} \lambda^k_i - \rho^k_j + \omega^k_j = 0 \qquad \left(j \in N^k_{CS}\right)$$

Primal and dual feasibility say that the solution to the linear program must satisfy the primal constraints and the dual variables must be non-negative. Complementary slackness enforces that a constraint is binding or the corresponding dual variable zero: Either the primal (dual) constraint is satisfied with equality or the dual (primal) variable is zero. Stationarity describes the relationship between the dual variables, where ρ^k_j and ω^k_j measure the difference between the revenue associated with some itinerary and the shadow prices on the flights that it uses. If the revenue exceeds the shadow prices, ρ^k_j is positive. Hence, all demand is accepted and ω^k_j zero because of complementary slackness. Similarly, if the revenue is below the shadow prices, ω^k_j must be positive and the accepted demand as well as ρ^k_j are zero.

A special case occurs when complementary slackness is satisfied with both terms being zero. In this case, the linear programming solution is called *degenerate* as stated in the definition by Greenberg (1986)[11]. A degenerate linear program may not be uniquely defined because some variables can leave or enter the basic solution without changing the objective value. Degeneracy is frequently found in real-world problems and may cause the simplex algorithm to cycle. It occurs due to special constellations of the coefficients in the objective function and the right hand side bounds (Dantzig, 1963).

In our subsequent analysis, degenerate capacity allocations associated with the demand constraints (5.6c) and (5.6d) describe a borderline case in which demand for an itinerary is fully accepted (rejected) without providing positive (negative) surplus. In contrast to Greenberg, we are not interested in the linear program as a whole, but whether a particular itinerary is degenerate or not. This is captured in Definition 1, a refined version of the definition by Greenberg.

Definition 1 (Degenerate Capacity Allocation)**.** *A capacity allocation y_j^k is called degenerate if an optimal solution exists such that either $\rho_j^k = y_j^k - D_j^k = 0$ or $\omega_j^k = y_j^k = 0$.*

In the decentralized alliance case, degeneracy of code-share itineraries must be checked for a common capacity allocation that is local optimal to the linear program of every carrier $k \in C_j$. We call a code-share itinerary $j \in N_{CS}$ degenerate if such a capacity allocation exists and it is degenerate in the sense of Definition 1 in all local solutions of the carriers $k \in C_j$. Similarly, we call it non-degenerate, if it is not degenerate in the local solutions. However, as the carriers take independent allocation decisions, it may happen that an itinerary is degenerate in one solution and non-degenerate in another one. We refer to this situation as *partial degeneracy* and it is defined in Definition 2.

[11]"A solution is degenerate if two complementary terms are zero." (Greenberg, 1986, p. 1)

Definition 2 (Partial Degeneracy). *A code-share itinerary $j \in N_{CS}$ is partially degenerate if there exists a common and local optimal capacity allocation y_j^k for all $k \in C_j$, which is degenerate in at least one but not all local solutions.*

The KKT conditions as well as the definitions of degeneracy are used several times throughout the upcoming sections to characterize linear programming solutions and to describe the adaptation process. In addition, we introduce two simple propositions that capture the relationships between the prorated code-share valuations and the central dual problem as well as the overall value of the itinerary. We begin with the connection to the central dual problem. The dual associated with the primal linear program (5.1) is denoted by (6.1).

$$\begin{aligned} Dual(\hat{Z}) \quad &= \quad \min \sum_{i \in M} \lambda_i x_i + \sum_{j \in N} \rho_j D_j \\ &\text{subject to} \sum_{i \in M} a_{ij} \lambda_i + \rho_j \geq r_j \quad && \forall\, j \in N \\ &\lambda_i \geq 0 && \forall\, i \in M \\ &\rho_j \geq 0 && \forall\, j \in N \end{aligned} \tag{6.1}$$

Duality theory states that the primal of a maximization problem approaches the optimal solution from below, while the dual problem approaches the optimum from above. As a consequence, weak duality says that any feasible solution to the primal problem is always smaller or equal than any feasible solution to the dual problem. Furthermore, strong duality implies that the optimal objective value of the primal and the dual are equal. For more details on duality theory and the derivation of the dual of a linear program we refer to Bertsimas and Tsitsiklis (1997, Chapter 4).

With respect to the code-share model we note that all combinations of prorated valuations $\tilde{r}_j^k \in \tilde{R}$ are feasible to the central dual problem. This result is captured in Proposition 1.

Proposition 1. *For all code-share valuations in $\tilde{r}_j^k \in \tilde{R}$, the optimal local solutions are feasible to the central dual problem.*

Proof. To be feasible, the optimal local dual variables must satisfy the constraints of the central dual problem. To see this, note that optimality of the local primal solution implies that the KKT conditions hold. As the code-share valuations $\tilde{r}_j^k \in \tilde{R}$ and $\omega_j^k \geq 0$, stationarity implies

$$\begin{aligned} r_j = \sum_{k \in C_j} \tilde{r}_j^k &= \sum_{k \in C_j} \sum_{i \in A_j^k} \lambda_i^k + \sum_{k \in C_j} \rho_j^k - \sum_{k \in C_j} \omega_j^k \\ &\leq \sum_{k \in C_j} \sum_{i \in A_j^k} \lambda_i^k + \sum_{k \in C_j} \rho_j^k \ . \end{aligned}$$

Defining $\lambda_i = \lambda_i^k$ for all $i \in A_j$ and $\rho_j = \sum_{k \in C_j} \rho_j^k$, the local solutions satisfy the central dual problem, which shows Proposition 1. □

The second proposition defines a relationship between the prorated valuations and the overall value of the code-share itinerary. Therefore, let

$$\Delta s_j = r_j - \sum_{k \in C_j} \sum_{i \in A_j^k} \lambda_i^k \qquad \forall j \in N_{CS}$$

denote the difference between the complete itinerary fare and the sum of all local optimal shadow prices. Δs_j describes the gain ($\Delta s_j > 0$) or loss ($\Delta s_j < 0$) associated with code-share itinerary j.

Proposition 2. *It holds that:*

$$\text{If } \Delta s_j > 0, \text{ then } \tilde{r}_j^k > \sum_{i \in A_j^k} \lambda_i^k \text{ for at least one } k \in C_j \ .$$

$$\text{If } \Delta s_j < 0, \text{ then } \tilde{r}_j^k < \sum_{i \in A_j^k} \lambda_i^k \text{ for at least one } k \in C_j \ .$$

Proof. We can show Proposition 2 by contradiction and noticing that $r_j = \sum_{k \in C_j} \tilde{r}_j^k$. First, assume that $\tilde{r}_j^k \leq \sum_{i \in A_j^k} \lambda_i^k$ for all $k \in C_j$. Summing up over $k \in C_j$ gives

$$\sum_{k \in C_j} \tilde{r}_j^k = r_j \leq \sum_{k \in C_j} \sum_{i \in A_j^k} \lambda_j^k \quad \Leftrightarrow \quad \Delta s_j \leq 0 \ ,$$

which shows the first part.

Similarly, we can show the second part by assuming that $\tilde{r}_j^k \geq \sum_{i \in A_j^k} \lambda_i^k$ for all $k \in C_j$. This leads to

$$\sum_{k \in C_j} \tilde{r}_j^k = r_j \geq \sum_{k \in C_j} \sum_{i \in A_j^k} \lambda_j^k \quad \Leftrightarrow \quad \Delta s_j \geq 0 \ ,$$

and completes the proof to Proposition 2. □

Finally, we notice that Δs_j can be written using solely the dual variables ρ_j^k and ω_j^k. According to the stationarity criterion of the KKT conditions, $\tilde{r}_j^k - \sum_{i \in A_j^k} \lambda_i^k$ can be replaced by $\rho_j^k - \omega_j^k$ such that

$$\begin{aligned} \Delta s_j = r_j - \sum_{k \in C_j} \sum_{i \in A_j^k} \lambda_i^k &= \sum_{k \in C_j} \tilde{r}_j^k - \sum_{k \in C_j} \sum_{i \in A_j^k} \lambda_i^k \\ &= \sum_{k \in C_j} \left(\tilde{r}_j^k - \sum_{i \in A_j^k} \lambda_i^k \right) \\ &= \sum_{k \in C_j} \left(\rho_j^k - \omega_j^k \right) \ . \end{aligned} \tag{6.2}$$

With the description of the Karush-Kuhn-Tucker conditions, degeneracy as well as the two propositions, we laid the foundation for our subsequent analysis and are ready to continue with the derivation of the optimality conditions.

6.3. Conditions for Central Optimality

The network linear programs allocate the available capacity to the itineraries. For multi-leg itineraries the same capacity must be reserved on each flight along the itinerary. In the decentralized alliance case, however, different carriers control the flights belonging to code-share itineraries and they make independent allocation decisions. As a result, the allocation decisions may differ for the same itinerary. To maximize their joint revenue, carriers should imitate the central solution. Consequently, taking contrary acceptance decisions on a single itinerary is not optimal.

Subsection 6.3.1 captures this idea and describes necessary conditions for implementing the central optimal solution in decentralized systems. In the following Subsection 6.3.2, we first examine why the previous result is not sufficient and provide a refined formulation.

6.3.1. Necessary Conditions

Let $\mathbf{y}^* = \left(y_j^* : j \in N\right)$ be the central optimal capacity allocations to (5.1). They maximize the revenue across the entire alliance network and satisfy constraints (5.1b) and (5.1c). In order to implement the central optimal solution, Theorem 1 states necessary conditions on the choice of the prorated valuations $\tilde{r}_j^k$. If these conditions are not satisfied, the local solutions cannot be central optimal.

Theorem 1 (Necessary Condition)**.** *To implement the central optimal capacity allocation y_j^*, the prorated valuations $\tilde{r}_j^k \in \tilde{R}$ must satisfy one of these conditions:*

$$\sum_{i \in A_j^k} \lambda_i^k \leq \tilde{r}_j^k \leq r_j - \sum_{k' \in C_j^{\setminus \{k\}}} \sum_{i \in A_j^{k'}} \lambda_i^{k'} \qquad \forall\, k \in C_j \tag{6.3a}$$

$$r_j - \sum_{k' \in C_j^{\setminus \{k\}}} \sum_{i \in A_j^{k'}} \lambda_i^{k'} \leq \tilde{r}_j^k \leq \sum_{i \in A_j^k} \lambda_i^k \qquad \forall\, k \in C_j \tag{6.3b}$$

Proof. Because allocations $\mathbf{y}^*$ are primal optimal to the central problem, they are also primal optimal to the linear programs (5.6) and can be implemented by taking $y_j^k = y_j^*$ for all $j \in N_{CS}$ and all $k \in C_j$. Local optimality implies further that the KKT conditions to the local problems must hold and valuations $\tilde{r}_j^k$ need to be chosen accordingly.

To show Condition (6.3a), assume $y_j^k = y_j^* > 0$ for some code-share itinerary $j \in N_{CS}$. By complementary slackness it holds that $\omega_j^k = 0$ whenever $y_j^k > 0$. Moreover, dual feasibility requires that $\rho_j^k \geq 0$ and stationarity implies

$$0 \leq \rho_j^k = \tilde{r}_j^k - \sum_{i \in A_j^k} \lambda_i^k \ .$$

Rearranging the terms gives

$$\tilde{r}_j^k \geq \sum_{i \in A_j^k} \lambda_i^k \ ,$$

which shows the left hand side of Condition (6.3a).

The right hand side of Condition (6.3a) follows directly from the left hand side. As $\tilde{r}_j^k \geq \sum_{i \in A_j^k} \lambda_i^k$ should hold for all carriers $k \in C_j$ simultaneously and $r_j = \sum_{k \in C_j} \tilde{r}_j^k$, we can write

$$r_j = \sum_{k \in C_j} \tilde{r}_j^k \geq \tilde{r}_j^k + \sum_{k' \in C_j^{\setminus \{k\}}} \sum_{i \in A_j^{k'}} \lambda_i^{k'}$$

and hence,

$$\tilde{r}_j^k \leq r_j - \sum_{k' \in C_j^{\setminus \{k\}}} \sum_{i \in A_j^{k'}} \lambda_i^{k'} \ .$$

To show Condition (6.3b), assume $y_j^k = y_j^* < D_j$ for some code-share itinerary $j \in N_{CS}$. Analogously to the proof of Condition (6.3a), complementary slackness requires that ρ_j^k

is zero and by stationarity as well as dual feasibility we can write

$$0 \leq \omega_j^k = \sum_{i \in A_j^k} \lambda_i^k - \tilde{r}_j^k \ ,$$

which implies that

$$\tilde{r}_j^k \leq \sum_{i \in A_j^k} \lambda_i^k \ .$$

Last, the left hand side of Condition (6.3b) follows from the same argument as the right hand side of Condition (6.3a) and completes the proof to Theorem 1. □

The conditions in Theorem 1 are relatively weak because they do not distinct the borderline case $\tilde{r}_j^k = \sum_{i \in A_j^k} \lambda_i^k$. It belongs to both conditions and thereby makes them ambiguous. With the exception of partial degeneracy, the borderline case can be separated, providing stronger necessary conditions on the optimality of the local solutions. Theorem 2 captures this result.

Theorem 2 (Strong Necessary Condition)**.** *To implement the central optimal capacity allocation y_j^* such that the local solutions are not partially degenerate, the prorated valuations $\tilde{r}_j^k \in \tilde{R}$ must satisfy one of these conditions:*

$$\sum_{i \in A_j^k} \lambda_i^k < \tilde{r}_j^k < r_j - \sum_{k' \in C_j^{\setminus \{k\}}} \sum_{i \in A_j^{k'}} \lambda_i^{k'} \qquad \forall\, k \in C_j \tag{6.4a}$$

$$\tilde{r}_j^k = \sum_{i \in A_j^k} \lambda_i^k \qquad \forall\, k \in C_j \tag{6.4b}$$

$$r_j - \sum_{k' \in C_j^{\setminus \{k\}}} \sum_{i \in A_j^{k'}} \lambda_i^{k'} < \tilde{r}_j^k < \sum_{i \in A_j^k} \lambda_i^k \qquad \forall\, k \in C_j \tag{6.4c}$$

Proof. We first note that all $\tilde{r}_j^k$ satisfying Conditions (6.4a)–(6.4c) also satisfy the weaker necessary conditions defined in Theorem 1. As can be easily verified, if valuations $\tilde{r}_j^k$ satisfy Condition (6.4a) or (6.4c), they also satisfy Condition (6.3a) or (6.3b), respectively. For valuations satisfying the border line condition (6.4b), Conditions (6.3a) and (6.3b) hold simultaneously. On the contrary, not every choice of prorated valuations in Theorem 1 satisfies Theorem 2 and hence, they describe a superset of the latter.

In the second step, we show that every optimal allocation can be expressed by valuations satisfying Theorem 2. As in the proof of Theorem 1, we implement the central optimal solution in the local systems by defining $y_j^k = y_j^*$. We first assume that for some code-share itinerary $j \in N_{CS}$ the associated revenue exceeds the shadow prices of the required resources, i.e. $r_j > \sum_{i \in A_j} \lambda_i$. Stationarity together with complementary slackness implies that $y_j^* = D_j$. To implement this solution in the local systems, Theorem 1 states that $\tilde{r}_j^k \geq \sum_{i \in A_j^k} \lambda_i^k$ for all $k \in C_j$. If this condition holds with equality for carrier k, stationarity says that $\rho_j^k = \omega_j^k = 0$ and y_j^k is degenerate according to Definition 1. Furthermore, Proposition 2 implies that $\tilde{r}_j^{k'} > \sum_{i \in A_j^{k'}} \lambda_i^{k'}$ for some other carrier $k' \in C_j^{\setminus\{k\}}$ such that itinerary j is partially degenerate by Definition 2. So, to be non-degenerate, Condition (6.3a) must be satisfied with strict inequality for all carriers $k \in C_j$.

In a similar fashion we can show the case $r_j < \sum_{i \in A_j} \lambda_i$. It implies that $y_j^* = 0$ and Theorem 1 states that $\tilde{r}_j^k \leq \sum_{i \in A_j^k} \lambda_i^k$ for all $k \in C_j$. If this condition is satisfied with equality for some carrier k, the dual variables ρ_j^k and ω_j^k are both zero and itinerary j is partially degenerate by Definition 1 and Proposition 2. To avoid partial degeneracy, the condition must be satisfied with strict inequality.

Last, we consider the borderline case $r_j = \sum_{i \in A_j} \lambda_i$, which does not impose a unique value on y_j^*. If $0 < y_j^* = y_j^k < D_j$, complementary slackness implies that $\rho_j^k = \omega_j^k = 0$ and hence, $\tilde{r}_j^k = \sum_{i \in A_j^k} \lambda_j^k$. If $y_j^k = D_j^k$, then $\tilde{r}_j^k$ may also be larger than the sum of the local shadow prices. Similarly, if $y_j^k = 0$, then $\tilde{r}_j^k$ may be smaller than the shadow prices. However, due to the optimality of the dual variables λ_j^k, the valuation of another carrier $k' \neq k$ must be smaller or larger than the local shadow prices, respectively, which contracts the conditions in Theorem 1 and hence, $\tilde{r}_j^k = \sum_{i \in A_j^k} \lambda_j^k$ for all $k \in C_j$. □

Theorem 2 is stronger than Theorem 1 because the borderline case is separated. However, this is done at the expense of an additional requirement − partial degeneracy. On the one hand, the stronger conditions exclude many choices of $\tilde{r}_j^k$, which are feasible in Theorem 1, but do not implement the central solution. On the other hand, a small subset of optimal valuations is excluded: When the allocations are partially degenerate, all carriers may implement the central optimal solution, but the prorated valuations do not satisfy Theorem 2. The relationship between the sets of necessary, strong necessary and optimal valuations is graphically depicted in Figure 6.2.

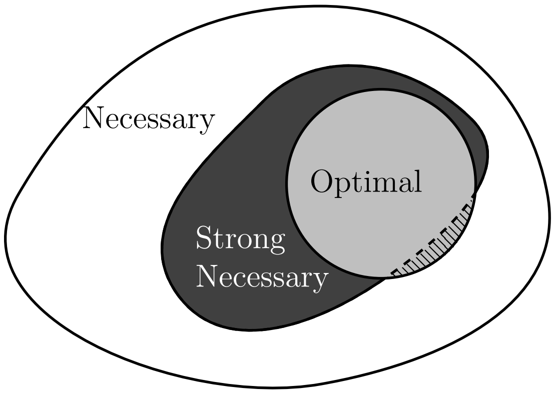

Figure 6.2.: Sets of Feasible and Optimal Code-Share Valuations

To resolve partial degeneracy, one additional update step is required. It needs to reallocate some of the surplus (loss) from one of the carriers for which the itinerary is not degenerate, to the carriers for which the itinerary is degenerate. This step does neither change the overall result nor the dual variables λ_j^k because the capacity allocations remain the same. Furthermore, it is always possible to redistribute the surplus (loss) among the carriers and therefore, there always exist alternative valuations that resolve partial degeneracy without violating the conditions in Theorem 2.

6.3.2. Sufficient Conditions

Theorem 1 and 2 state necessary, but not sufficient conditions: So, the valuations $\tilde{r}_j^k$ do not guarantee to determine the central optimum. The reason is that the borderline case $\tilde{r}_j^k = \sum_{i \in A_j^k} \lambda_i^k$ does not lead to unique capacity allocations and the local allocations do not necessarily match. This section presents a small example demonstrating such a case and then provides an alternative formulation, which is necessary and sufficient.

Example: Assume there are two airlines A and B and two flights 1 and 2. Airline A operates flight 1; Airline B operates flight 2. There are three itineraries, two intraline one-leg itineraries and one code-share itinerary spanning both flights. The capacity per flight is two seats and the expected demand is depicted in Table 6.1:

Table 6.1.: Expected Itinerary Demand

Itinerary	Airline	Demand	Value
1	A	1	100
2	B	1	150
1, 2	A, B	3	300

The central optimum is to accept two code-share passengers and no intraline ones, i.e. $y_1 = y_2 = 0$ and $y_{1,2} = 2$. To solve the decentralized problems, take for example the valuations $\tilde{r}_{1,2}^A = 200$ and $\tilde{r}_{1,2}^B = 100$. Optimizing the local problems of airlines A and B gives:

$$\text{A: } y_1 = 0;\ y_{1,2}^A = 2;\ \lambda_1^A = 200$$
$$\text{B: } y_2 = 1;\ y_{1,2}^B = 1;\ \lambda_2^B = 100$$

Both valuations satisfy Condition (6.4b), but do not implement the central solution. The reason is that they are not central feasible, i.e. they are not equal: $y_{1,2}^A = 2 \neq 1 = y_{1,2}^B$.

There are two possibilities that build up on Theorem 2 and also ensure sufficiency. On the one hand, we can apply the ideas of Hu et al. (2013) and use the central optimal shadow prices in Conditions (6.4a)–(6.4c). On the other hand, we can force the capacity allocations of all carriers to be equal on code-share itineraries. As we are interested in finding the central optimal shadow prices in the first place, using them as input would make our goal redundant. Further, we explicitly forbid to use the central solution and replacing the local shadow prices by the central ones is therefore no valid alternative. Consequently, we are left with the second alternative, focussing on the central feasibility of the independent allocations. The resulting sufficient condition is described in Theorem 3.

Theorem 3 (Sufficient Condition)**.** *If the local optimal solutions are central feasible, they implement the central optimal solution.*

Proof. Assume that the local primal solutions to linear programs (5.6) are central feasible. By Proposition 1 they are also feasible to the central dual problem. Furthermore, strong duality implies that the optimal objective function values to the primal linear programs (5.6) and the respective dual problems are equal. Because the local solutions are central feasible, the local objective values sum up to the alliance solution. Consequently, the local solutions are primal and dual feasible to the central problem and also have the same aggregate objective function value. Hence, weak duality implicates that the local solutions implement the central optimum. □

Central feasible means that the local capacity allocations of all carriers are equal for a common code-share itinerary. As there may be multiple optimal solutions due to degeneracy and non-uniqueness, it suffices to have one combination that satisfies central feasibility. To illustrate this problem as well as to explain the intuition behind Theorem 3, we continue the example above.

Example Continued: The previous result for this example was unique and satisfied Theorem 2, but the capacity allocation by Airline A was larger than the one by Airline B: $y^A_{1,2} = 2 \neq 1 = y^B_{1,2}$. To adjust this, we need to allocate more revenue to Airline B. For example, we choose new valuations $\tilde{r}^A_{1,2} = 150$ and $\tilde{r}^B_{1,2} = 150$. The updated linear programming solutions become:

$$\text{A: } \tilde{r}^A_{1,2} = 150;\ y_1 = 0;\ y^A_{1,2} = 2;\ \lambda^A_1 = 150$$
$$\text{B: } \tilde{r}^B_{1,2} = 150;\ y_2 = 0;\ y^B_{1,2} = 2;\ \lambda^B_2 = 150$$

The updated solutions still satisfy Theorem 2 and the local capacity allocations $y^A_{1,2} = 2 = y^B_{1,2}$ are central feasible. Hence, the local solutions are central optimal according to Theorem 3. However, the allocations reported by Airline B are not unique and $y^B_{1,2}$ may vary between one and two. To resolve non-uniqueness, the valuations could be changed further, for example to $\tilde{r}^A_{1,2} = 125$ and $\tilde{r}^B_{1,2} = 175$. Then all local solutions are also unique.

Finally, we observe that the aggregate objective value of the local solutions corresponds to the central objective value.

$$Z^A_{CS} + Z^B_{CS} = 2 \times 150 + 2 \times 150 = 600 = \hat{Z}$$

In contrast, the initial valuations are dual feasible but not optimal, and in line with weak duality, the aggregate objective value is larger than the optimal one:

$$Z^A_{CS} + Z^B_{CS} = 2 \times 200 + 150 + 100 = 650 > 600 = \hat{Z}$$

Changing the initial valuations too much (e.g. $\tilde{r}^A_{1,2} = 50$ and $\tilde{r}^B_{1,2} = 250$) or in the opposite direction (e.g. $\tilde{r}^A_{1,2} = 250$ and $\tilde{r}^B_{1,2} = 50$) increases the aggregate result as well.

The example demonstrates the implications of Theorem 3 and dual feasibility. We continue this discussion during the derivations of the sufficient update rule in the next section and also investigate how central feasibility can be checked in light of degeneracy and non-uniqueness.

6.4. Updating Procedure

If the conditions in Theorem 1, 2 or 3 are not satisfied, the code-share valuations should be updated. So, let $\breve{r}_j^k$ to be the updated valuation for code-share itinerary j of carrier k and υ_j^k to be the change in the valuation or simply the update. Then we define $\breve{r}_j^k = \tilde{r}_j^k + \upsilon_j^k$ and assume that $\sum_{k \in C_j} \upsilon_j^k = 0$ to ensure that the total value of the itinerary does not change.

As with the description of the optimality conditions, we divide our discussion in two parts. Subsection 6.4.1 describes the updating procedure converging to the necessary conditions. Subsection 6.4.2 extends these results to the sufficient case.

6.4.1. Necessary Update Rule

The derivations in this section are based on the necessary conditions stated in Theorem 1, but almost analogously apply to the stronger necessary conditions in Theorem 2. The slight modifications needed to adjust the update rule are discussed in-line with the respective results.

To begin with, Proposition 2 implies that the valuations need to satisfy Condition (6.3a) whenever $\Delta s_j > 0$ and Condition (6.3b) whenever $\Delta s_j < 0$. The corresponding bounds on the updates υ_j^k are stated in Lemma 1.

Lemma 1 (Bounds on the Updates)**.** *If the updates υ_j^k satisfy*

$$\omega_j^k - \rho_j^k + \min\{0, \Delta s_j\} \;\leq\; \upsilon_j^k \;\leq\; \omega_j^k - \rho_j^k + \max\{0, \Delta s_j\}\,, \tag{6.5}$$

then the updated valuations $\breve{r}_j^k$ satisfy Theorem 1.

Proof. By stationarity we add $\tilde{r}_j^k$ to υ_j^k and $\sum_{i\in A_j^k} \lambda_i^k + \rho_j^k - \omega_j^k$ to the left and right hand side. As ρ_j^k and ω_j^k cancel each other out, we get

$$\sum_{i\in A_j^k} \lambda_i^k + \min\{0, \Delta s_j\} \leq \tilde{r}_j^k + \upsilon_j^k \leq \sum_{i\in A_j^k} \lambda_i^k + \max\{0, \Delta s_j\}\ . \tag{6.6}$$

Replacing $\tilde{r}_j^k + \upsilon_j^k$ by $\breve{r}_j^k$ and assuming either $\Delta s_j \geq 0$ or $\Delta s_j \leq 0$, the above formulation reduces to

$$\sum_{i\in A_j^k} \lambda_i^k \leq \breve{r}_j^k \leq \sum_{i\in A_j^k} \lambda_i^k + \Delta s_j \quad \text{and} \quad \sum_{i\in A_j^k} \lambda_i^k + \Delta s_j \leq \breve{r}_j^k \leq \sum_{i\in A_j^k} \lambda_i^k\ .$$

In each equation we further replace Δs_j by $r_j - \sum_{k'\in C_j} \sum_{i\in A_j^{k'}} \lambda_i^{k'}$. The shadow prices of carrier k cancel out and we get

$$\begin{aligned} \sum_{i\in A_j^k} \lambda_i^k \leq \breve{r}_j^k \leq r_j - \sum_{k'\in C_j^{\setminus\{k\}}} \sum_{i\in A_j^{k'}} \lambda_i^{k'} \\ r_j - \sum_{k'\in C_j^{\setminus\{k\}}} \sum_{i\in A_j^{k'}} \lambda_i^{k'} \leq \breve{r}_j^k \leq \sum_{i\in A_j^k} \lambda_i^k\ , \end{aligned}$$

which are the conditions in Theorem 1. Furthermore, this shows that (6.6) provides an alternative formulation for the conditions in Theorem 1. $\square$

The boundaries in Lemma 1 separate the cases $\Delta s_j \geq 0$ and $\Delta s_j \leq 0$ and use regular inequalities. This suffices to satisfy Theorem 1. To further satisfy the stronger necessary conditions, the inequalities must be replaced with strict inequalities. Moreover, the case

$\Delta s_j = 0$ needs to be separated as the right and left hand side become equal and would not be feasible. The respective boundaries are depicted in Table 6.2 for the different constellations of Δs_j, ρ_j^k and ω_j^k and help to clarify the intuition behind Lemma 1.

Table 6.2.: Bounds on the Updates

Δs_j	ρ_j^k, ω_j^k	Bounds
$\Delta s_j > 0$	$\rho_j^k > 0$, $\omega_j^k = 0$	$-\rho_j^k < \upsilon_j^k < \Delta s_j - \rho_j^k$
	$\rho_j^k = 0$, $\omega_j^k = 0$	$0 < \upsilon_j^k < \Delta s_j$
	$\rho_j^k = 0$, $\omega_j^k > 0$	$\omega_j^k < \upsilon_j^k < \Delta s_j + \omega_j^k$
$\Delta s_j < 0$	$\rho_j^k > 0$, $\omega_j^k = 0$	$\Delta s_j - \rho_j^k < \upsilon_j^k < -\rho_j^k$
	$\rho_j^k = 0$, $\omega_j^k = 0$	$\Delta s_j < \upsilon_j^k < 0$
	$\rho_j^k = 0$, $\omega_j^k > 0$	$\Delta s_j + \omega_j^k < \upsilon_j^k < \omega_j^k$
$\Delta s_j = 0$	$\rho_j^k > 0$, $\omega_j^k = 0$	$-\rho_j^k \le \upsilon_j^k \le -\rho_j^k$
	$\rho_j^k = 0$, $\omega_j^k = 0$	$0 \le \upsilon_j^k \le 0$
	$\rho_j^k = 0$, $\omega_j^k > 0$	$\omega_j^k \le \upsilon_j^k \le \omega_j^k$

Every constellation of the three variables leads to unique lower and upper bounds on the update υ_j^k. For example consider the first row: Overall the code-share itinerary is beneficial as its value exceeds the sum of the local shadow prices ($\Delta s_j > 0$). Proposition 2 states that at least one valuation $\tilde{r}_j^k$ must be larger than the local shadow prices and hence, the updated valuations of all carriers $k \in C_j$ must satisfy Condition (6.4a) of Theorem 2. For a particular carrier that can lead to three situations: First, its current valuation already exceeds the shadow prices ($\rho_j^k > 0$, $\omega_j^k = 0$). Then the valuation should not be reduced by more than the current excess, i.e. $-\rho_j^k < \upsilon_j^k$. Second, the valuation is equal to the shadow prices ($\rho_j^k = 0$, $\omega_j^k = 0$). Then it should not be decreased ($0 < \upsilon_j^k$). Third, the valuation is below the shadow prices ($\rho_j^k = 0$, $\omega_j^k > 0$). To satisfy Condition (6.4a), it must be increased above the shadow prices, i.e. $\omega_j^k < \upsilon_j^k$. The right hand side in all three cases states the maximum update such that all carriers satisfy Condition (6.4a). Following the same logic, every of the nine constellations of Δs_j, ρ_j^k and ω_j^k can be explained.

Applying Lemma 1 provides code-share valuations satisfying Theorem 1 and possibly Theorem 2. However, the boundaries are based on the old shadow prices given the previous valuations $\tilde{r}_j^k$. Resolving the linear programs may change the boundaries and the valuations need to be updated. For this purpose, let

$$\mathcal{A} = \left\{ \alpha_j^k \geq 0 : \sum_{k \in C_j} \alpha_j^k = 1 \text{ for all } j \in N_{CS} \right\}$$

denote the set of proration rates, i.e. percentages governing the allocations to the carriers. We assume that every α_j^k is positive and their sum over all carriers equal to one. Theorem 4 states the necessary update rule.

Theorem 4 (Update Rule). *Updates of the form* $\upsilon_j^k = \omega_j^k - \rho_j^k + \alpha_j^k \Delta s_j$ *with* $\alpha_j^k \in \mathcal{A}$ *converge to valuations satisfying Theorem 1.*

Proof. For the proof of Theorem 4 we first show that the prorated valuations always sum up to the total fare of the itinerary. So, assuming $\sum_{k \in C_j} \tilde{r}_j^k = r_j$, we need to show that $\sum_{k \in C_j} \check{r}_j^k = \sum_{k \in C_j} \tilde{r}_j^k + \sum_{k \in C_j} \upsilon_j^k = r_j$. To see this, we replace υ_j^k by the update rule from Theorem 4 and notice that $\sum_{k \in C_j} \alpha_j^k = 1$. We get

$$\begin{aligned} r_j &= \sum_{k \in C_j} \tilde{r}_j^k + \sum_{k \in C_j} \upsilon_j^k \\ &= \sum_{k \in C_j} \tilde{r}_j^k + \sum_{k \in C_j} \omega_j^k - \sum_{k \in C_j} \rho_j^k + \Delta s_j \ . \end{aligned}$$

In view of (6.2), Δs_j is replaced by $\sum_{k \in C_j} \left(\rho_j^k - \omega_j^k \right)$ and the expression reduces to

$$\begin{aligned} r_j &= \sum_{k \in C_j} \tilde{r}_j^k + \sum_{k \in C_j} \omega_j^k - \sum_{k \in C_j} \rho_j^k + \sum_{k \in C_j} \left(\rho_j^k - \omega_j^k \right) \\ &= \sum_{k \in C_j} \tilde{r}_j^k \ . \end{aligned}$$

Hence, the total value of the itinerary remains constant and $\sum_{k \in C_j} \upsilon_j^k = 0$.

Next, we show that the update rule converges. Convergence means that the prorated valuations become stable when consecutively applying the update rule. Hence, the updates υ_j^k must become zero and the shadow prices constant. We prove this by induction: First, we show that Theorem 1 is satisfied after a single iteration. Then, we assume that it is satisfied after an arbitrary number of iterations and prove that it is also satisfied in the subsequent iteration with the bounds getting tighter.

We start with arbitrary valuations $\tilde{r}_j^k \in \tilde{R}$. As the proration rates $\alpha_j^k \in \mathcal{A}$ are bounded by zero and one, the updates are within the bounds defined by Lemma 1 such that after one iteration the updated code-share valuations satisfy the conditions in Theorem 1.

Having shown that Theorem 1 is satisfied after a single iteration, we need to show that it holds after an arbitrary number of iterations and that the boundaries on $\tilde{r}_j^k$ get tighter. To prove this, let $\tilde{\boldsymbol{\lambda}}^k = \left(\tilde{\lambda}_i^k : i \in M^k\right)$ and $\breve{\boldsymbol{\lambda}}^k = \left(\breve{\lambda}_i^k : i \in M^k\right)$ be the local shadow prices determined with $\tilde{r}_j^k$ and $\breve{r}_j^k$, respectively, and assume that the updated valuations $\breve{r}_j^k = \tilde{r}_j^k + \omega_j^k - \rho_j^k + \alpha_j^k \Delta s_j$ satisfy Theorem 1. Because the updates are exact and only a single itinerary is changed at a time, the updated shadow prices $\breve{\lambda}_i^k$ are bounded by the previous shadow prices $\tilde{\lambda}_i^k$ and the updated valuations $\breve{r}_j^k$, where $\Delta\tilde{s}_j = r_j - \sum_{i \in A_j^k} \tilde{\lambda}_i^k$:

$$\sum_{i \in A_j^k} \tilde{\lambda}_i^k + \min\left\{0, \alpha_j^k \Delta\tilde{s}_j\right\} \leq \sum_{i \in A_j^k} \breve{\lambda}_i^k \leq \sum_{i \in A_j^k} \tilde{\lambda}_i^k + \max\left\{0, \alpha_j^k \Delta\tilde{s}_j\right\}$$

$$\min\left\{\sum_{i \in A_j^k} \tilde{\lambda}_i^k, \breve{r}_j^k\right\} \leq \sum_{i \in A_j^k} \breve{\lambda}_i^k \leq \max\left\{\sum_{i \in A_j^k} \tilde{\lambda}_i^k, \breve{r}_j^k\right\}$$

To see this, assume that $\Delta\tilde{s}_j \geq 0$. The update rule ensures that $\breve{r}_j^k \geq \tilde{\lambda}_i^k$ and because all else remains equal, allocation y_j^k does not decrease such that $\breve{\lambda}_i^k \geq \tilde{\lambda}_i^k$. Similarly, $\breve{\lambda}_i^k \leq \breve{r}_j^k$ because otherwise y_j^k would be zero and optimality of the two solutions would imply $\breve{\lambda}_i^k = \tilde{\lambda}_i^k$ such that $\tilde{\lambda}_i^k \leq \breve{r}_j^k < \breve{\lambda}_i^k = \tilde{\lambda}_i^k$, which is a contradiction. By the same arguments, we can show the second case with $\Delta\tilde{s}_j \leq 0$.

Given the bounds on the updated shadow prices, we distinct two cases. On the one hand, assume $\sum_{i\in A_j^k} \breve{\lambda}_i^k \neq \sum_{i\in A_j^k} \tilde{\lambda}_i^k$. Then the updated shadow prices are either smaller or larger than the previous ones depending on $\Delta\tilde{s}_j$. Consequently, the absolute surplus or loss of the itinerary decreases, i.e. $|r_j - \sum_{i\in A_j^k} \breve{\lambda}_i^k| = |\Delta\breve{s}_j| < |\Delta\tilde{s}_j| = |r_j - \sum_{i\in A_j^k} \tilde{\lambda}_i^k|$, and the resulting bounds defined by Lemma 1 get tighter:

$$\sum_{i\in A_j^k} \tilde{\lambda}_i^k + \min\left\{0, \alpha_j^k \Delta\tilde{s}_j\right\} < \sum_{i\in A_j^k} \breve{\lambda}_i^k + \min\left\{0, \alpha_j^k \Delta\breve{s}_j\right\} \leq$$
$$\sum_{i\in A_j^k} \breve{\lambda}_i^k + \max\left\{0, \alpha_j^k \Delta\breve{s}_j\right\} < \sum_{i\in A_j^k} \tilde{\lambda}_i^k + \max\left\{0, \alpha_j^k \Delta\tilde{s}_j\right\} \ .$$

If, on the other hand, the shadow prices are the same, i.e. $\sum_{i\in A_j^k} \breve{\lambda}_i^k = \sum_{i\in A_j^k} \tilde{\lambda}_i^k$, then $\Delta\tilde{s}_j = \Delta\breve{s}_j$ and hence, $\tilde{r}_j^k = \breve{r}_j^k$. As the valuations no longer change, the solutions are stable and this completes the proof to Theorem 4. □

So far we faced the updating process from the perspective of the current valuations, i.e. given the current valuations we examined how they should be updated. Alternatively, we could neglect the current valuations and calculate new ones independently. The equivalent formulation for $\breve{r}_j^k$ without $\tilde{r}_j^k$ is:

$$\begin{aligned} \breve{r}_j^k &= \tilde{r}_j^k + \upsilon_j^k \\ &= \tilde{r}_j^k + \omega_j^k - \rho_j^k + \alpha_j^k\,\Delta s_j \\ &= \sum_{i\in A_j^k} \lambda_i^k + \alpha_j^k\,\Delta s_j \end{aligned} \tag{6.7}$$

To derive (6.7) substitute the expression for υ_j^k from Theorem 4 in the formula for $\breve{r}_j^k$. In view of the KKT conditions, the first three terms are replaced by $\sum_{i\in A_j^k} \lambda_i^k$ such that $\breve{r}_j^k$ no longer depends on $\tilde{r}_j^k$. Expression (6.7) is more intuitive and provides an alternative view on the structure of the update rule: Allocate to every carrier the sum of its current shadow prices plus some share of the surplus or loss of the itinerary.

In order to satisfy the stronger necessary conditions described in Theorem 2, the proration rates α_j^k must be strictly positive and smaller than one, i.e. $\alpha_j^k \in (0,1)$ for all carriers $k \in C_j$. Then $\alpha_j^k \Delta s_j \neq 0$ whenever $\Delta s_j \neq 0$ and the updates satisfy the boundaries with strict inequalities as outlined in Table 6.2.

Convergence of Theorem 4 is guaranteed for fixed (static) proration rates as well as some dynamic ones as for example proration by shadow prices λ_i^k. The latter is probably the most popular rule in this context and may also be based on bid prices in practice. The next subsection shows that this scheme implements the update rule in Theorem 4, but does not satisfy the stronger necessary conditions stated in Theorem 2.

Shadow Price Proration

The shadow price proration scheme is mentioned in Wright (2010) and Hu et al. (2013). It allocates the code-share fares by the ratio of the current shadow prices on the flights of each carrier, i.e. the shadow prices of one carrier over the sum of shadow prices of all carriers as defined by (6.8). For the special case $\sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \lambda_i^{k'} = 0$, the shadow prices must be set to arbitrary, positive values to avoid division by zero.

$$\tilde{r}_j^k = \frac{\sum_{i \in A_j^k} \lambda_i^k}{\sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \lambda_i^{k'}} \; r_j \qquad \forall\, k \in C_j; j \in N_{CS} \tag{6.8}$$

Proration by shadow prices weights the individual valuations by the marginal value of each carrier's resources and reflects the value of the flights to the entire alliance network. Furthermore, we can show that consecutively applying (6.8) gives a solution satisfying Theorem 1. This is captured in Lemma 2.

Lemma 2 (Shadow Price Proration Scheme)**.** *Proration by shadow prices using expression* (6.8) *implements the update rule in Theorem 4.*

Proof. (6.8) can be implemented with the update rule from Theorem 4 by choosing $\alpha_j^k = \sum_{i \in A_j^k} \lambda_i^k / \sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \lambda_i^{k'}$. To see this, we rewrite (6.8).

$$\begin{aligned}
\tilde{r}_j^k &= \frac{\sum_{i \in A_j^k} \lambda_i^k}{\sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \lambda_i^{k'}} \; r_j \\
&= \frac{\sum_{i \in A_j^k} \lambda_i^k}{\sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \lambda_i^{k'}} \left(\sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \lambda_i^{k'} + \sum_{k' \in C_j} \rho_j^{k'} - \sum_{k' \in C_j} \omega_j^{k'} \right) \\
&= \frac{\sum_{i \in A_j^k} \lambda_i^k}{\sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \lambda_i^{k'}} \left(\sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \lambda_i^{k'} + \Delta s_j \right) \\
&= \sum_{i \in A_j^k} \lambda_i^k + \frac{\sum_{i \in A_j^k} \lambda_i^k}{\sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \lambda_i^{k'}} \; \Delta s_j
\end{aligned}$$

By Assumption 3 and stationarity we replace r_j by the equivalent expression using the dual variables. Then the first summation cancels out, while the sum over the dual variables ρ_j^k and ω_j^k corresponds to Δs_j by (6.2). The resulting equation has the same form as (6.7) and implements Theorem 4 with $\alpha_j^k = \sum_{i \in A_j^k} \lambda_i^k / \sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \lambda_i^{k'}$. ☐

Because expression $\sum_{i \in A_j^k} \lambda_i^k$ can be zero, the proration rate α_j^k may also become zero. Similarly, if it is nonzero, but all other shadow prices are zero, the proration rate becomes one. Hence, $\alpha_j^k \in [0, 1]$ and the valuations determined by (6.8) do not satisfy the stronger necessary conditions stated in Theorem 2.

6.4.2. Sufficient Update Rule

In view of Theorem 3, the local solutions must be central feasible to achieve sufficiency. Besides the borderline case $\tilde{r}_j^k = \sum_{i \in A_j^k} \lambda_j^k$, this is already the case due to complementary slackness. As preventing the borderline case would limit the practicability of the linear program, this sections derives an alternative procedure. The only requirement is that alliance partners exchange their code-share allocations in addition to the shadow prices.

Associated with the improved updating scheme there come two challenges: (1) We need to calculate alternative update values. So far we used the difference between the prorated valuations and the shadow prices, which becomes zero in the borderline case. (2) We need to find some representation of the optimality criterion (central feasibility), which is easy to check and terminates the updating procedure.

Particularly the second challenge is difficult to realize due to degeneracy and non-uniqueness of the local linear programming solutions. There may be multiple optimal capacity allocations to each of the carriers' optimization problems. Furthermore, they may depend on each other such that the allocation to one code-share itinerary impacts the allocation to another. Because of this, there is no straightforward or simple way to implement the optimality criterion without violating the assumptions of decentralization and incomplete information.

As a consequence, both procedures developed in this section are heuristic. They neither ensure convergence nor to find the central optimum. Nevertheless, the exchange of the capacity allocations improves the update decisions and the methods promise superior results. In the next subsections, we outline the approaches and Section 6.5 discusses why they may fail when there exist multiple local solutions. The performance of the heuristics is analyzed in Chapter 7.

Heuristic 1 – Comparison of Capacity Allocations

The idea for the first heuristic is to compare the code-share allocations determined by each airline and to update the valuations as long as they do not match. For this purpose, the airlines need to solve their local optimization problems and determine bounds on the code-share allocations. These are shared with the partner and the valuations are updated when the allocations do not overlap. Such a comparison is easy to implement and more accurate than the necessary procedure because it also considers borderline allocations. The difficulty is to determine the update values. For this we introduce two auxiliary linear programs. One determines the incremental update and one the decremental update, where incremental and decremental update refer to the minimum changes in the code-share valuations $\tilde{r}_j^k$ that are necessary to increase or decrease the respective capacity allocations y_j^k.

The auxiliary linear programs are given by (6.9) and (6.10). They solely differ by the optimization goal and must be solved for every code-share itinerary j and every carrier k separately. The objective of the first problem is to maximize the capacity allocation y_j^k. The constraints are the same ones as in formulation (5.6) − the local network linear program with code-shares − plus an equality constraint corresponding to the objective function of (5.6). It must be equal to the optimal objective value denoted by Z_{CS}^k. The first auxiliary program (6.9) determines the incremental update.

$$\begin{aligned}
\overline{y}_j^k \quad = \quad & \max \quad y_j^k \\
& \text{subject to} \sum_{j \in N^k} r_j^k y_j^k + \sum_{j \in N_{CS}^k} \tilde{r}_j^k y_j^k = Z_{CS}^k \\
& \sum_{j \in N^k} a_{ij}^k y_j^k + \sum_{j \in N_{CS}^k} a_{ij}^k y_j^k \leq x_i^k \qquad \forall\, i \in M^k \\
& 0 \leq y_j^k \leq D_j^k \qquad\qquad\qquad\qquad \forall\, j \in N^k \cup N_{CS}^k
\end{aligned} \tag{6.9}$$

The second auxiliary linear program (6.10) includes the same constraints, but minimizes allocation y_j^k and determines the decremental update.

$$\begin{aligned}
\underline{y}_j^k \quad = \quad & \min \quad y_j^k \\
& \text{subject to} \sum_{j \in N^k} r_j^k y_j^k + \sum_{j \in N_{CS}^k} \tilde{r}_j^k y_j^k = Z_{CS}^k \\
& \sum_{j \in N^k} a_{ij}^k y_j^k + \sum_{j \in N_{CS}^k} a_{ij}^k y_j^k \leq x_i^k \qquad \forall\, i \in M^k \\
& 0 \leq y_j^k \leq D_j^k \qquad\qquad\qquad\qquad \forall\, j \in N^k \cup N_{CS}^k
\end{aligned} \tag{6.10}$$

The update values are derived from the shadow prices of the first constraint. So, let $\overline{\lambda}_j^k$ and $\underline{\lambda}_j^k$ denote the shadow prices associated with the equality constraints in (6.9) and (6.10), respectively. For itinerary j and carrier k we define the incremental update to be $\phi_j^k = 1/\overline{\lambda}_j^k$ and the decremental update to be $\psi_j^k = 1/\underline{\lambda}_j^k$.

To see that these values describe the minimum updates, note that $\overline{\lambda}_j^k$ and $\underline{\lambda}_j^k$ correspond to the change in y_j^k when Z_{CS}^k is reduced by one unit. So, to satisfy the first constraint with $Z_{CS}^k - 1$, some positive optimal allocation needs to be reduced and the free capacity can be reallocated such that y_j^k is either maximized or minimized. This means to select some itinerary j' traversing the same flights as j and to reallocate capacity between j' and j. This changes the value of the equality constraint by $\Delta y_j^k(r_{j'}^k - \tilde{r}_j^k)$, i.e. the amount of reallocated capacity times the difference between the valuations of j' and j. As the change in the constraint should be one, $\Delta y_j^k(r_{j'}^k - \tilde{r}_j^k) = 1$ and by the definition of the shadow prices, we can replace Δy_j^k with $\overline{\lambda}_j^k$ or $\underline{\lambda}_j^k$. Evaluating the resulting expressions for all feasible itineraries j', the shadow prices $\overline{\lambda}_j^k$ and $\underline{\lambda}_j^k$ determined in (6.9) and (6.10) are the maximal change in y_j^k, corresponding to itinerary j' minimizing the difference between $\tilde{r}_j^k$ and $r_{j'}^k$ – the value we are looking for. Rearranging the terms shows that the minimum change in the valuations is the inverse of the shadow prices.

$$\begin{aligned}
\phi_j^k &= \min_{r_{j'}^k > \tilde{r}_j^k:\ A_j^k = A_{j'}^k \text{ and } j' \neq j} \left(r_{j'}^k - \tilde{r}_j^k \right) = \frac{1}{\overline{\lambda}_j^k} \quad \text{and} \\
\psi_j^k &= \min_{r_{j'}^k < \tilde{r}_j^k:\ A_j^k = A_{j'}^k \text{ and } j' \neq j} \left(\tilde{r}_j^k - r_{j'}^k \right) = \frac{1}{\underline{\lambda}_j^k} \quad .
\end{aligned}$$

In addition to the update values, the objective values $\overline{y}_j^k$ and $\underline{y}_j^k$ provide bounds on the capacity allocations y_j^k. With the same code-share valuations and keeping the optimal solution value Z_{CS}^k constant, the respective allocation may be changed to any value $y_j^k \in \left[\underline{y}_j^k, \overline{y}_j^k\right]$ by reallocating capacity. In order to determine the central optimum, it suffices to find central allocations y_j that belong to the range of feasible allocations of every operating carrier $k \in C_j$. Therefore, the valuations are updated if the ranges do not overlap and otherwise not.

Last, we note that ϕ_j^k is positive whenever $\overline{y}_j^k < D_j^k$ and ψ_j^k is positive whenever $\underline{y}_j^k > 0$. Otherwise they are zero. Moreover, ϕ_j^k and ψ_j^k correspond to the dual variables ω_j^k and ρ_j^k if $\tilde{r}_j^k \neq \sum_{i\in A_j^k} \lambda_i^k$. Then $\phi_j^k = \omega_j^k$ and $\psi_j^k = \rho_j^k$.

Having derived the update values as well as the ranges on the code-share allocations, we define the actual update rule. It distinguishes different cases depending on the values of $\overline{y}_j^k$ and $\underline{y}_j^k$, and uses ϕ_j^k and ψ_j^k to update the code-share valuations accordingly.

- If the allocations $y_j^k \in \left[\underline{y}_j^k, \overline{y}_j^k\right]$ of all carriers $k \in C_j$ overlap, there exists a feasible allocation and we apply the necessary update rule described by Theorem 4:

$$\upsilon_j^k = \omega_j^k - \rho_j^k + \alpha_j^k \, \Delta s_j \quad .$$

- If some allocations do not overlap, their valuations are updated with the minimum amount necessary to reduce the gap. This means to either increase the valuations of all allocations that are too small or to decrease the valuations of the ones that are too big. For example, assume there are two airlines, say k and k', with $\overline{y}_j^k < \underline{y}_j^{k'}$. The updates are

$$\upsilon_j^k = \min\left\{\phi_j^k, \psi_j^{k'}\right\} \qquad \text{and} \qquad \upsilon_j^{k'} = -\min\left\{\phi_j^k, \psi_j^{k'}\right\} \quad .$$

  With more than two carriers all allocations need to be compared and the direction in which the valuations are updated depends on the minimum of the aggregated updates necessary to adjust all deviating allocations.

Heuristic 2 – Evaluation of Primal Feasible Solution

The idea behind the second heuristic is that the optimal solution must be bounded by primal and dual feasible solutions. In view of Proposition 1, all code-share valuations $\tilde{r}_j^k \in \tilde{R}$ are dual central feasible and provide an upper bound on the central solution.

At the same time, optimizing the local networks with some predefined combination of central feasible code-share allocations gives a lower bound on the optimum. Our heuristic uses both bounds to derive improved update values that reduce the gap between the primal and dual feasible solutions.

The process starts with the airlines independently solving their linear programs and exchanging the capacity allocations from code-share itineraries. In the second step, they fix the code-share allocations to some combination of central feasible allocations

$$y_j^k = y_j \quad \forall\, k \in C_j, j \in N_{CS}$$

and resolve their local linear programs.

$$\begin{aligned}
Z_{CS}^{k\,(\mathrm{PF})} \quad &= \quad \max \sum_{j \in N^k} r_j^k y_j^k + \sum_{j \in N_{CS}^k} \tilde{r}_j^k y_j^k \\
&\text{subject to} \sum_{j \in N^k} a_{ij}^k y_j^k + \sum_{j \in N_{CS}^k} a_{ij}^k y_j^k \leq x_i^k && \forall\, i \in M^k \\
&0 \leq y_j^k \leq D_j^k && \forall\, j \in N^k \\
&y_j^k = y_j && \forall\, j \in N_{CS}^k
\end{aligned}$$

In the two-carrier case, each airline may fix its code-share allocations to the values determined by the partner. Alternatively and particularly with more carriers, they may adopt a convex and feasible combination of the individual allocations. Despite the choice, the solutions of the resolved linear programs are primal feasible to the central problem, bound the optimal solution from below and provide a Nash equilibrium (Hu et al., 2013). As the airlines still use the original solution to control their inventory, the primal feasible solutions only help to adjust the code-share valuations, but do not limit the sovereignty of the airlines nor impact their control decisions. The calculation of the steering parameters remains independent.

The linear program with fixed code-share allocations provides a second set of shadow prices, which we denote by $\boldsymbol{\lambda}^{k(\mathrm{PF})} = \left(\lambda_i^{k(\mathrm{PF})} : i \in M^k\right)$, and a second set of surplus values $\boldsymbol{\Delta s}^{(\mathrm{PF})} = \left(\Delta s_j^{(\mathrm{PF})} : j \in N_{CS}\right)$. The superscript (PF) stands for primal feasible to indicate the difference to the original, dual feasible solution. With the second set of shadow prices we calculate a second set of boundaries on the prorated valuations using expression (6.6). Together with the dual feasible solution, we get two ranges:

$$\mathrm{LB}_j^k = \sum_{i \in A_j^k} \lambda_i^k + \min\{0, \Delta s_j\} \leq \tilde{r}_j^k \leq \sum_{i \in A_j^k} \lambda_i^k + \max\{0, \Delta s_j\} = \mathrm{UB}_j^k$$

$$\mathrm{LB}_j^{k(\mathrm{PF})} = \sum_{i \in A_j^k} \lambda_i^{k(\mathrm{PF})} + \min\left\{0, \Delta s_j^{(\mathrm{PF})}\right\} \leq \tilde{r}_j^k \leq \sum_{i \in A_j^k} \lambda_i^{k(\mathrm{PF})} + \max\left\{0, \Delta s_j^{(\mathrm{PF})}\right\} = \mathrm{UB}_j^{k(\mathrm{PF})}$$

In the optimum, the prorated valuations $\tilde{r}_j^k$ should satisfy the primal and dual feasible boundaries. Therefore, the update rule selects valuations belonging to both ranges or, if they do not overlap, from the gap between them.

- If the ranges overlap,

$$\max\left\{\mathrm{LB}_j^k, \mathrm{LB}_j^{k(\mathrm{PF})}\right\} \leq \tilde{r}_j^k \leq \min\left\{\mathrm{UB}_j^k, \mathrm{UB}_j^{k(\mathrm{PF})}\right\} .$$

- If the ranges do not overlap,

$$\min\left\{\mathrm{UB}_j^k, \mathrm{UB}_j^{k(\mathrm{PF})}\right\} \leq \tilde{r}_j^k \leq \max\left\{\mathrm{LB}_j^k, \mathrm{LB}_j^{k(\mathrm{PF})}\right\} .$$

The boundaries can be combined to a single update rule calculating the code-share valuations for every carrier $k \in C_j$ by

$$\begin{aligned}\tilde{r}_j^k = \alpha_j^k \ \max&\left\{\min\left\{\mathrm{UB}_j^k, \mathrm{UB}_j^{k(\mathrm{PF})}\right\}, \max\left\{\mathrm{LB}_j^k, \mathrm{LB}_j^{k(\mathrm{PF})}\right\}\right\} \\ &+ (1-\alpha_j^k) \ \min\left\{\min\left\{\mathrm{UB}_j^k, \mathrm{UB}_j^{k(\mathrm{PF})}\right\}, \max\left\{\mathrm{LB}_j^k, \mathrm{LB}_j^{k(\mathrm{PF})}\right\}\right. ,\end{aligned}$$

where $\alpha_j^k \in \mathcal{A}$ weights the two bounds.

6.5. Discussion of the Algorithm

Our discussion in the previous sections shows that sufficiency is difficult to achieve due to non-uniqueness and degeneracy. Because of this, the comparison of the capacity allocations in the first heuristic as well as the derivation of the central feasible solution in the second heuristic may not provide a unique result and cause the update procedure to fail. To illustrate this problem, we consider once more the example from Section 6.3.2.

Example Continued: With valuations $\tilde{r}^A_{1,2} = 150$ and $\tilde{r}^B_{1,2} = 150$, capacity allocation $y^B_{1,2}$ is between one and two and Airline A's allocation is exactly two. A simple comparison might lead to different conclusions: Either $y^A_{1,2} = 2 = y^B_{1,2}$ and the valuations are not updated, or $y^A_{1,2} = 2 > y^B_{1,2} \geq 1$ and the update rule increases valuation $\tilde{r}^B_{1,2}$ in order to reduce the gap. If, alternatively, we compare the ranges on the code-share allocations, i.e. $y^A_{1,2} \in [2, 2]$ and $y^B_{1,2} \in [1, 2]$, the result is unique and the update process terminates as the ranges overlap.

To calculate the central feasible solution in the second heuristic, the local allocations are fixed to a central feasible value. For example, we may choose one of the extreme points $y^A_{1,2} = y^B_{1,2} = y_j = 1$ or $y^A_{1,2} = y^B_{1,2} = y_j = 2$. The respective linear programming solutions are

$$\begin{array}{llll} \text{A:} & y^A_{1,2} \equiv 2;\ y_1 = 0;\ \lambda^A_1 = 100 & \text{or} & y^A_{1,2} \equiv 1;\ y_1 = 1;\ \lambda^A_1 = 0 \\ \text{B:} & y^B_{1,2} \equiv 2;\ y_2 = 0;\ \lambda^B_2 = 150 & \text{or} & y^B_{1,2} \equiv 1;\ y_2 = 1;\ \lambda^B_2 = 0 \end{array}$$

If the code-share allocation is two, all capacity is occupied and the shadow prices equal the value of the local demand. If the code-share allocation is one, both airlines also accept the local passengers and the shadow prices become zero as there is no additional demand. Hence, the central feasible solutions may lead to different local solutions and consequently impact the updates.

The example demonstrates how the heuristics are impacted by non-uniqueness and degeneracy. In addition, the decisions on code-share itineraries may affect each other and makes both approaches heuristic. As these special situations only occur in the borderline case, i.e. when the value of a code-share itinerary matches the shadow prices, we do not expect a severe impact on the overall performance of the algorithms. With a large number of itineraries the results should converge close to the central solution and the effects are likely to diminish. This holds for the two heuristics as well as the necessary procedure. We further investigate this aspect in the next chapter during our numerical analysis.

Regarding the two heuristics, the first one is computationally more expensive because of the calculation of the minimal update values. The linear program must be computed twice for every code-share itinerary on the borderline and potentially all itineraries in the worst case. In contrast, the second heuristic solves the linear program only once more and all update values are derived from this solution − a crucial advantage in practice.

The theoretical derivations described above only hold when the linear programs of all airlines are solved after every update (exact updates). The alternative are approximate updates, where the linear programs are not solved in between. They may be interpreted as a greedy heuristic to the exact procedure: Every update corresponds to the locally best decision given the current information − the last linear programming solution not including the updates from other itineraries. Together these local optimal decisions do not necessarily produce the global optimum, but approximate the exact solution in reasonable time. We examine the performance of both variants in the next chapter. For a description of greedy algorithms we refer to the book by Cormen et al. (1990, Chapter 16). While this chapter focuses in the theoretical aspects of code-share control, Chapter 8 provides a discussion on the practical implications of our findings.

6.5.1. Implications for the Contractual Level

The algorithm is intended for the optimal inventory control on the operational level. However, we can also derive some logical consequences for the contractual level. The basic idea behind Theorem 2 is that all operating carriers must make the same acceptance decision. This translates to the fact that the code-share itinerary must be beneficial for each of them. The same requirement may also be applied during the revenue sharing. It implies that the airlines should receive at least the marginal value of their resources. This corresponds to the stability property in cooperative game theory (see Section 3.1.1) and is essential to find an revenue allocation in the core of the alliance game.

While it is important that no airline is worse off by accepting a code-share passenger, i.e. its compensation is less than what it could earn in its intraline network, the exact distribution of the surplus revenue Δs_j can be handled flexibly. Given the base revenue corresponding to the opportunity cost of the consumed inventory – represented by the shadow or bid prices –, the remaining revenue may be allocated by a different principle. For example, proration by mileage reflects the relative operating costs or the carriers may negotiate route-specific schemes based on their market positions[12].

Finally, additional aspects such as robustness and cheating may play a role for the revenue sharing scheme. While our approach focuses on the optimality of the alliance solution, the compensation scheme should also hinder airlines from taking selfish decisions. For example, assume that the entire revenue is prorated by bid prices and due to low demand all bid prices are one. By manually changing it to two, an airline immediately increases its revenue share from 50 to 67 percent without significantly impacting the acceptance decisions. Such simple possibilities to manipulate revenue allocations may motivate carriers to cheat on their partners and the benefits from improved optimization are eroded in the long run.

[12] Gerlach et al. (2013) note that the market position impacts the potential benefits and risks from code-sharing and should be reflected in the revenue sharing.

7. Numerical Experiments

This chapter presents numerical experiments testing the performance of the algorithms developed in the previous chapter. We describe the experimental setup and formally define the valuation methods in Section 7.1. The results are presented in Section 7.2.

7.1. Experimental Setup

For our analysis we conduct Monte Carlo simulations over three different scenarios. The scenarios use the same network with the same flights, but differ by the number of booking classes and the price structure. Per scenario we execute 10,000 instances with randomly generated demand and randomly generated prices.

Our model solely covers the revenue optimization characterized by the linear program used in the previous chapter. We assume that demand forecasts are deterministic and do not consider the other steps of the revenue management process. There is no customer interaction and our experimental environment is static.

The sample network for our analysis contains two airlines, who operate two flights each. The flights have a fixed capacity of 100 seats and there is only a single compartment. Every airline offers three intraline routes: the flight into the hub, the flight out of the hub and the transfer. Two complementary code-share connections arise by linking the flights of the two carriers. Parallel and virtual code-shares are not considered. The network is depicted in Figure 9.2.

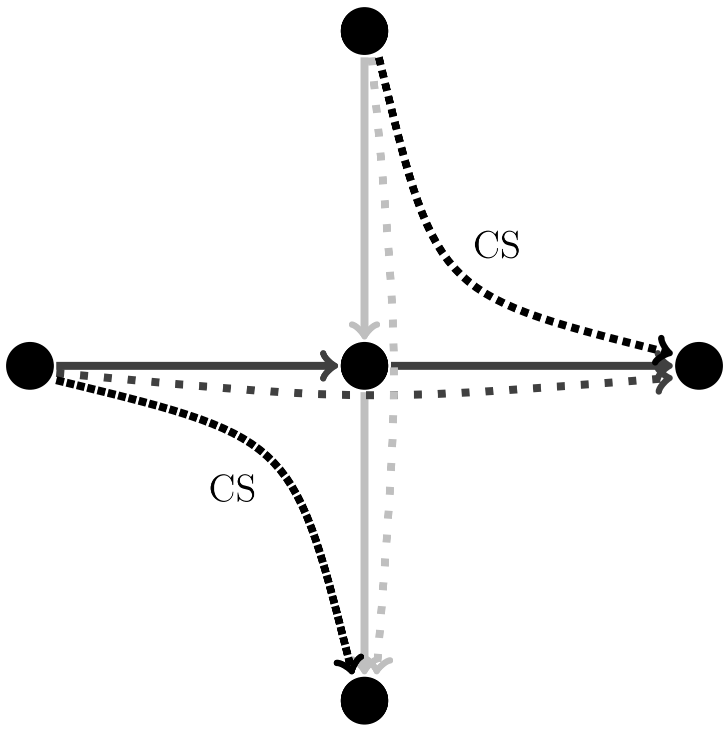

Figure 7.1.: Sample Network with two Airlines and four Flights

For this network we generate 10,000 random instances. An instance is described by the distances between the airports, the expected demand per flight and the relative value of each route. Per instance these parameters are randomly drawn from different distributions: The distance $\mathfrak{d}_i$ of flight i (corresponding to the length of flight i) is uniformly distributed between 1000 and 10000. The expected demand per flight is given by a factor of the total capacity. It is taken from a uniform distribution with boundaries 0.8 and 2.2. When it is larger than one, there is excess demand for the respective flight and capacity is tight. When it is below one, there is less demand than capacity. As the result of the optimization is trivial in the latter case, there is only a small number of flights with low demand. Last, the value of a route is determined by a factor taken from a normal distribution with mean one and standard deviation 0.1. It is multiplied with all itinerary fares on the respective route to change their relative value compared to other routes.

Given an instance, we calculate the expected demand as well as the base fare for each route. The former is determined by equally distributing the demand for a flight over local, transfer and code-share itineraries such that on average they occupy one third of the capacity each. The latter is set to the square root of the distance times the route value factor. The distance is the length of the underlying flight or the sum of the respective flight distances for transfer itineraries. We chose distance-based prices because interviews with practitioners have revealed that many network carriers base their pricing decisions in large parts on the distance of the itinerary. Hence, such fares reflect real-world pricing structures relatively well.

The setup described so far including the expected demands and the base fares remains the same in all scenarios. The scenarios solely differ by the number booking classes and the price structure that the airlines offer. The price structure is characterized by a fare multiplier γ. The number of booking classes is denoted by κ. The price of a booking class is determined by multiplying the price of the next lower booking class by γ. The price for the lowest booking class corresponds to the base fare such that the fare for the highest booking class becomes

$$base\_fare \ \times \ \gamma^{\kappa-1} \ .$$

The distribution of demand among the booking classes is determined by random weights. There are different weights for every scenario, every instance and every itinerary. They are taken from a normal distribution with mean one and standard deviation 0.25. The demand for a particular booking class on a specific route is the total expected demand for that route times the weight of the booking class over the sum of the weights of all booking classes on the respective route. If all weights were one, this would correspond to equally distributing the demand with ratio $1/\kappa$.

Taking the pair (κ, γ) with values $\{(2, 1.5), (6, 1.25), (12, 1.1)\}$, we get three scenarios on which we run the 10,000 instances each. The more booking classes there are, the smaller we choose the fare multiplier − reducing the price steps between consecutive fares on the same route. This leads to more and denser price points on each flight without changing the underlying network.

Each scenario is solved by three variants of the adaptation procedure − satisfying the different conditions in Theorem 1, 2 and 3, respectively − as well as several other static and dynamic valuation schemes for comparison. We formally describe the benchmark methods in the next paragraphs and refer to Section 3.2.2 for additional information. All dynamic methods including the adaptation schemes are executed with exact and approximate updates and the valuations must fully converge (all updates must be zero) or cycle (determining the same valuations again) in order to terminate the updating process. If in some rare instances neither of the two termination criteria is satisfied, the updating stops after 1,000 iterations.

7.1.1. Valuation Schemes

For the formal definition of the valuation schemes we apply the same notation as in the previous chapters. The abbreviations mentioned in brackets are used in the subsequent section to refer to the respective method.

- **Sufficient Updating Process (SUP):** Code-share fares are updated by the two heuristics (H1 and H2) outlined in Section 6.4.2. For the second heuristic we define $\alpha_j^k = 1/2$ for all carriers k and all code-share itineraries $j \in N_{CS}$.
- **Necessary Updating Process (NUP):** Code-share fares are updated by the rule described in Theorem 4. The proration rates are fixed to $\alpha_j^k = 1/2$ for all carriers k and all code-share itineraries $j \in N_{CS}$.

- **Shadow Price Proration (SPP):** The code-share fares are prorated by the ratio of the current shadow prices on the flights of each carrier as defined by (6.8). If the sum over all shadow prices is zero, they are set to one. SPP also implements Theorem 4, but determines α_j^k dynamically with $\alpha_j^k \in [0,1]$.

- **Absolute Shadow Price (ASP):** The current shadow prices of the partners' flights are subtracted from the full itinerary fare.

$$\tilde{r}_j^k = r_j - \sum_{\substack{k' \in C_j \\ k' \neq k}} \sum_{i \in A_j^{k'}} \lambda_i^{k'} \qquad \forall\, k \in C_j;\ j \in N_{CS}$$

- **Distance Proration (DP):** The code-share fare is prorated based on the relative distance flown by each carrier.

$$\tilde{r}_j^k = \frac{\sum_{i \in A_j^k} \mathfrak{d}_i^k}{\sum_{k' \in C_j} \sum_{i \in A_j^{k'}} \mathfrak{d}_i^{k'}}\ r_j \qquad \forall\, k \in C_j;\ j \in N_{CS}$$

- **Local Fare Proration (LFP):** The code-share fare is prorated by the local fares of the booking classes in which the code-share bookings are counted.

$$\tilde{r}_j^k = \frac{r_{j'}^k}{\sum_{k' \in C_j} r_{j'}^{k'}}\ r_j \qquad \forall\, k \in C_j;\ j \in N_{CS};\ j' \in N^k \text{ with } A_j^k = A_{j'}^k$$

- **Local Fare (LF):** Code-share bookings are valued by the local fares of the booking classes in which they are counted.

$$\tilde{r}_j^k = r_{j'}^k \qquad \forall\, k \in C_j;\ j \in N_{CS};\ j' \in N^k \text{ with } A_j^k = A_{j'}^k$$

- **Full Fare (FF):** Each carrier uses the total fare of the code-share itinerary to value its intraline sub-itinerary.

$$\tilde{r}_j^k = r_j \qquad \forall\, k \in C_j;\ j \in N_{CS}$$

These schemes are the most common ones discussed in the literature and used in practice. Five of them – LFP, DP and the adaptive schemes – prorate the full itinerary fare. The other three calculate adjustments, which may change the aggregated value of the itinerary. Furthermore, the two schemes LF and LFP are based on the local fares of the same booking class on the sub-itinerary. The results of the benchmarking experiments are presented next.

7.2. Computational Results

For better readability we use abbreviations to refer to the different methods and denote in brackets behind them whether the updates are exact or approximate, where E stands for exact and A for approximate. For each of the three scenarios we report the following results:

- The percentage of instances that converge to a stable solution.
- The percentage of instances that converge to the central shadow prices.
- The mean average percentage error (MAPE) between the central and the local shadow prices.
- The aggregated expected revenue of the local solutions in percentage of the central expected revenue.
- The size of the updates.

The results are structured in two categories. Section 7.2.1 discusses the quality of the local solutions in relation to the central solution. Afterwards, Section 7.2.2 examines the convergence behavior. In both sections we analyze the results of all three scenarios and compare them.

7.2.1. Quality of the Local Solutions

The primary goal of our approach is to determine the central shadow prices as most revenue management algorithms only use this information. The quality of the local shadow prices is depicted in Tables 7.1, 7.2 and 7.3. They show the percentage of central optimal instances as well as the MAPE between the local and the central shadow prices. All results reported in this section are taken from the last iteration, i.e. when all instances have converged.

Table 7.1.: Percentage of Central Optimal Shadow Prices (in Percent of Instances)

Scenario	(2, 1.5)	(6, 1.25)	(12, 1.1)
SUP-H1 (E)	96.90	98.86	98.72
SUP-H1 (A)	96.29	97.35	97.03
SUP-H2 (E)	95.23	96.71	96.04
SUP-H2 (A)	95.36	96.67	96.29
NUP (E)	59.21	70.15	71.07
NUP (A)	59.66	70.04	71.26
SPP (E)	28.76	35.16	32.86
SPP (A)	32.62	42.15	40.62
ASP (E)	49.84	58.66	59.37
ASP (A)	50.40	58.48	58.75
DP	2.89	1.10	0.47
LFP	4.93	1.73	0.77
LF	11.36	2.33	0.53
FF	11.65	11.45	11.13

Table 7.1 shows that the two heuristics outperform the other methods. They reach the central shadow prices in 95 percent and more of the instances. The second best procedure is NUP, which reaches the optimal values in 60–70 percent of the instances, followed by ASP with around 10 percent less. The worst result among the dynamic methods comes from SPP. The percentages vary between 30 to 40 percent. The performance of the static methods is well below the others.

When comparing the solutions with exact and approximate updates, we see only small differences in the final solutions. Except for SPP, the gap is below one percent and not resolving the linear programs seems to have no significant impact. Also, there is no clear difference across the three scenarios. In some of the dynamic methods we observe a small increase, while for the static methods there is a small, but consistent decrease.

As not all instances actually reach the central solution, a second quality measure is the error between the local and the central shadow prices. We calculate it in percent of the local shadow price. Table 7.2 denotes the error per flight averaged over all instances and all flights.

Table 7.2.: MAPE between Local and Central Shadow Prices

Scenario	(2, 1.5)	(6, 1.25)	(12, 1.1)
SUP-H1 (E)	0.76	0.15	0.08
SUP-H1 (A)	0.95	0.32	0.19
SUP-H2 (E)	0.35	0.19	0.08
SUP-H2 (A)	0.29	0.15	0.07
NUP (E)	8.99	3.59	1.96
NUP (A)	8.91	3.49	1.91
SPP (E)	27.42	24.24	25.25
SPP (A)	25.17	21.17	22.17
ASP (E)	13.36	5.12	2.33
ASP (A)	13.68	5.23	2.31
DP	26.60	26.46	25.80
LFP	23.82	23.50	23.13
LF	24.77	23.71	23.68
FF	31.81	27.59	27.50

Again the adaptation schemes outperform the other methods. The two heuristics return an average error below one percent, while the second lowest errors are determined by NUP. The static methods together with SPP give the highest errors. They vary between 20 to 30 percent.

We further see that the MAPE decreases across the three scenarios. This effect occurs for all methods and indicates that the denser the prices, the more precise are the local solutions. The reason is that with more price points the average gap between the central shadow prices and the next higher or lower valuations decreases. Consequently, the potential error is lower and the accuracy increases. The observed effect is particularly strong for the dynamic schemes as they attempt to approach the central solution and the more price points there are, the closer they get. For the static methods the effect diminishes because the calculation of the valuations is based on constant values and does not change.

Although the MAPE provides a good measure for the quality of the shadow prices, it highly correlates with the percentage of central optimal solutions. The higher the number of optimal instances, the lower becomes the error. Therefore, Table 7.3 provides an adjusted MAPE only including those instances that are not central optimal.

Table 7.3.: Adjusted MAPE between Local and Central Shadow Prices

Scenario	(2, 1.5)	(6, 1.25)	(12, 1.1)
SUP-H1 (E)	24.47	13.16	6.67
SUP-H1 (A)	25.54	12.10	6.31
SUP-H2 (E)	7.43	5.64	2.00
SUP-H2 (A)	6.16	4.53	1.91
NUP (E)	22.03	12.03	6.77
NUP (A)	22.10	11.64	6.74
SPP (E)	38.49	37.38	37.61
SPP (A)	37.35	36.60	37.34
ASP (E)	26.63	12.38	5.74
ASP (A)	27.58	12.59	5.60
DP	27.39	26.76	25.92
LFP	25.05	23.92	23.31
LF	27.95	24.27	23.81
FF	36.01	31.16	30.94

In terms of the adjusted MAPE, SPP performs worst and returns the largest errors, even larger than the ones from the static schemes. This causes a big gap between the instances that reach the central solution and the others. Plus, the adjusted MAPE does not decrease with the increase in price points as it happens for all other methods. It remains at 37 percent, while for the other dynamic schemes it falls below seven percent.

The lowest adjusted MAPE is determined by SUP-H2. It reaches the central shadow prices in 96 percent of the cases and minimizes the error in the other instances. In contrast, SUP-H1 achieves an even higher number of optimal instances, but the adjusted MAPE is also much higher – at about the same level as NUP and ASP. So, the scheme incurs a bigger risk when it fails to find the optimum. For the static methods the MAPE just changes a little as there are only a few optimal instances.

Besides the shadow prices, we use the difference between the alliance and the central revenue to indicate the quality of the local solutions. It is presented in Table 7.4.

Table 7.4.: Alliance Revenue (in Percent of Central Revenue)

Scenario	(2, 1.5)	(6, 1.25)	(12, 1.1)
SUP-H1 (E)	100.0035	100.0003	100.0001
SUP-H1 (A)	100.0047	100.0006	100.0002
SUP-H2 (E)	100.0044	100.0005	100.0001
SUP-H2 (A)	100.0035	100.0004	100.0001
NUP (E)	100.1225	100.0147	100.0041
NUP (A)	100.1170	100.0148	100.0042
SPP (E)	101.7490	101.1115	101.2989
SPP (A)	101.6706	100.9203	101.0505
ASP (E)	108.4003	112.1103	111.0378
ASP (A)	107.7188	111.7370	110.8621
DP	102.9527	102.2158	102.3621
LFP	102.4998	101.9493	102.0561
LF	113.5125	113.4051	113.4803
FF	136.6185	134.4494	134.9483

The revenue results across the different methods and the three scenarios support out findings about the quality of the shadow prices. As in the previous tables, the two heuristics as well as NUP outperform the other methods and reduce the revenue gap to 0.0001 percent of the central revenue, making it negligible. Moreover, the revenue gaps of these schemes decrease with the number of price points. The relative revenue of the static methods, on the contrary, is unaffected. All four return a similar value for each of the three scenarios. The reason for this observation is that only the distribution of the code-share demand over the booking classes changes, but its share of the total demand as well as the proration rates remain the same.

The valuation schemes that prorate exactly the itinerary fare show the smallest gaps, while the other methods lead to generally larger gaps. The greatest difference stems from full fare valuation. It overestimates the code-share value by factor two. As code-share demand takes up about one third of the total demand, the overall revenue is about one third higher. The second and third highest gaps are attributed to LF and ASP. Both also increase the overall itinerary value with local fare valuations being a little bit higher on average than the valuations from the shadow price adjustment.

It is also worthwhile to notice that the alliance revenue is always larger than the optimal revenue. This might seem unexpected as the central revenue is supposed to be the optimal (highest) revenue. The reason for the alliance revenue to be higher, is a direct consequence of Proposition 1. As has been noted in Section 6.3.2, the local solutions are dual feasible, but not primal feasible. From dual theory it holds that the dual solution to a maximization problem approaches the optimal result from above, i.e. is always larger than the optimal value. In view of Proposition 1, the local solutions must therefore be larger or equal than the central optimal solution.

7.2.2. Convergence Analysis

As the first step of the convergence analysis, we verify whether the dynamic methods determine a stable solution. The percentage of instances for which this is the case are depicted in Table 7.5. We see that for the two schemes based on Theorem 4 − NUP and SPP − 100 percent of the instances return a stable solution. This holds for exact as well as approximate updates. Of the two heuristics only the first one, the comparison of the local allocations, achieves 100 percent convergence when using exact updates. With approximate updates and both variants of the second heuristic find stable solutions in about 98 to 99 percent of the instances. For ASP the percentage drops to 53 to 63 percent. It does not implement any convergence idea and the results diverge more often.

Table 7.5.: Convergence to Stable Solution (in Percent of Instances)

Scenario	(2, 1.5)	(6, 1.25)	(12, 1.1)
SUP-H1 (E)	100.00	100.00	100.00
SUP-H1 (A)	98.32	98.53	98.37
SUP-H2 (E)	98.57	98.74	98.82
SUP-H2 (A)	98.57	98.65	98.78
NUP (E)	100.00	100.00	100.00
NUP (A)	100.00	100.00	100.00
SPP (E)	100.00	100.00	100.00
SPP (A)	100.00	100.00	100.00
ASP (E)	52.74	61.06	63.12
ASP (A)	53.03	60.19	61.63

Overall, we note three things: First, the two heuristics SUP-H1 and SUP-H2 have good convergence properties and reach a stable solution in most instances. Second, convergence is not much affected by the type of updates. Exact and approximate updates achieve about the same convergence rate. Third, although the solution quality of ASP is relatively good, a large share of instances does not converge.

In the following, we investigate the convergence behavior of the average update values, the shadow prices, the MAPE and the expected revenue, and analyze their development over the iterations. The results are depicted in Figures 7.2 to 7.10. If appropriate, we also show the results of the static schemes. They are plotted as horizontal gray lines.

We begin with the average update size per code-share itinerary. The results are divided into exact and approximate updates and depicted in Figures 7.2 and 7.3 for every of the three scenarios. First, we find that all cases show similar convergence patterns. In the first iteration, the updates are highest, but within five iterations they rapidly decrease close to zero. The only exception is ASP. In line with the structure of the scheme, the updates do not converge to zero and remain relatively high. However, the difference to the other updates reduces, the more price points there are. This effect is visible with exact as well as approximate updates.

Second, we see that the approximate updates are generally larger than the exact ones. By resolving the linear programs, the solution is improved and therefore, the subsequent updates become smaller. The only exception is SUP-H1. The updates correspond to the minimum change in the valuation and by improving the solution the minimum necessary to adjust the next allocation may increase. Moreover, the updates of SUP-H1 are consistently one of the smallest and also have the lowest convergence rate. Both can also be explained with the minimal adjustments: They are smaller than the ones from other schemes and if the minimal update does not suffice, another update is necessary in the next iteration.

Last, we notice that although the updates quickly decrease, only some converge to zero. This is the case for the methods based on Theorem 4 – NUP and SPP – as well as SUP-H1 with exact updates. For the other schemes the updates remain positive, either reaching a stable value as for ASP or oscillating between nearby values. The latter is not visible in the graphs and is caused by the instances not converging to a stable solution.

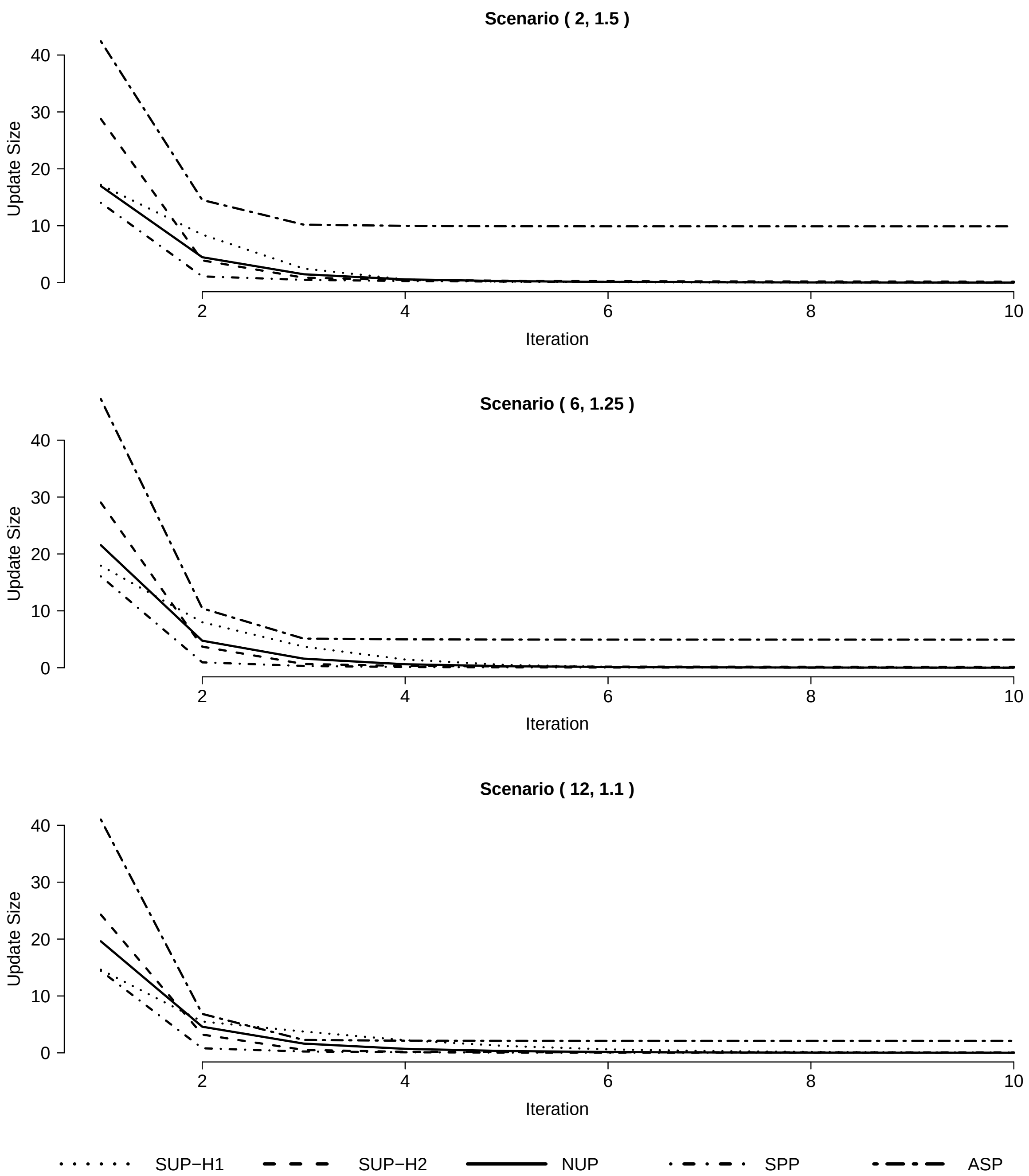

Figure 7.2.: Average Update Size – Exact Updates

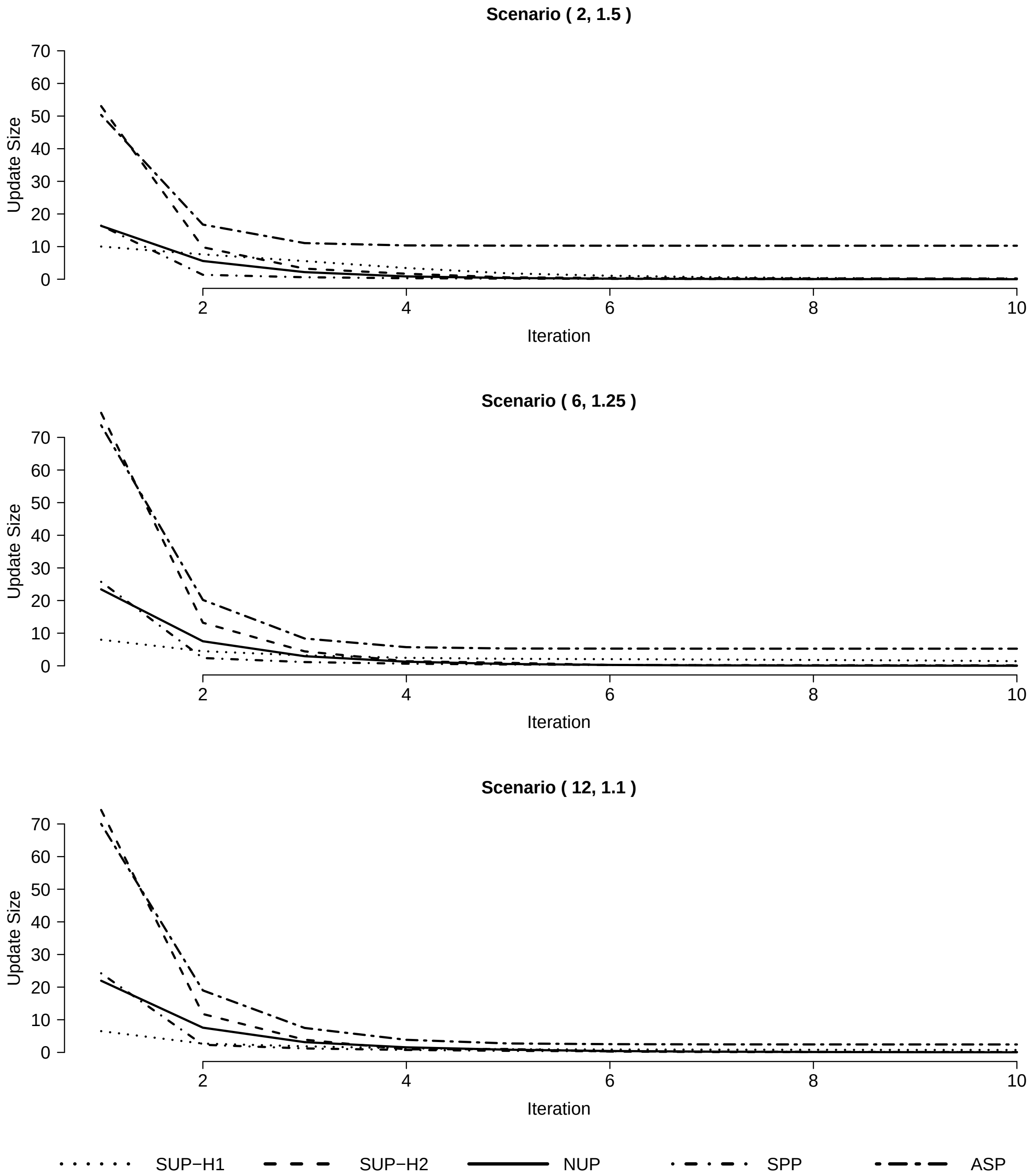

Figure 7.3.: Average Update Size – Approximate Updates

Next, we look at the convergence of the shadow prices. Figures 7.4 and 7.5 depict the number of central optimal instances per iteration. The results for exact and approximate updates are plotted on a logarithmic scale over the first 1000 iterations. We see that after the first iteration the highest percentages are achieved by SPP and ASP. These schemes provide the best initial valuations and reach up to 25 percent optimal instances. They also consistently outperform the best static scheme (FF), while some of the other methods need one or two iterations more to pass FF, in particular in the scenario with 12 booking classes.

Although SPP and ASP start well, both do not increase much over the subsequent iterations such that all other methods get ahead of them. The results of SPP only slightly improve and the convergence rates are very small. ASP gives the second lowest percentages, however it is the first method to reach stable shadow prices in all instances. They converge within five iterations.

After ten iterations, SUP-H2 provides consistently the best results with percentages of around 90 percent. Moreover, SUP-H2 and NUP show a similar pattern: Both develop analogously for about ten iterations and then increase sharply before they reach a stable level. The sudden jump in the optimal shadow prices is a coincidence explained by the way we check central optimality. An instance is counted as central optimal when all local shadow prices are within ± 0.0001 of the central shadow prices. If we increase the accuracy of this check, the effect occurs later and disappears eventually, providing a smoother curve.

In contrast to the other methods, the convergence behavior of SUP-H1 is impacted by the scenarios and the type of updates. With exact updates the convergence rate is higher than for approximate updates and the more price points there are, the longer SUP-H1 needs to reach the final values. In the biggest scenario and using approximate updates it takes up to 100 iterations – the worst convergence throughout our analysis.

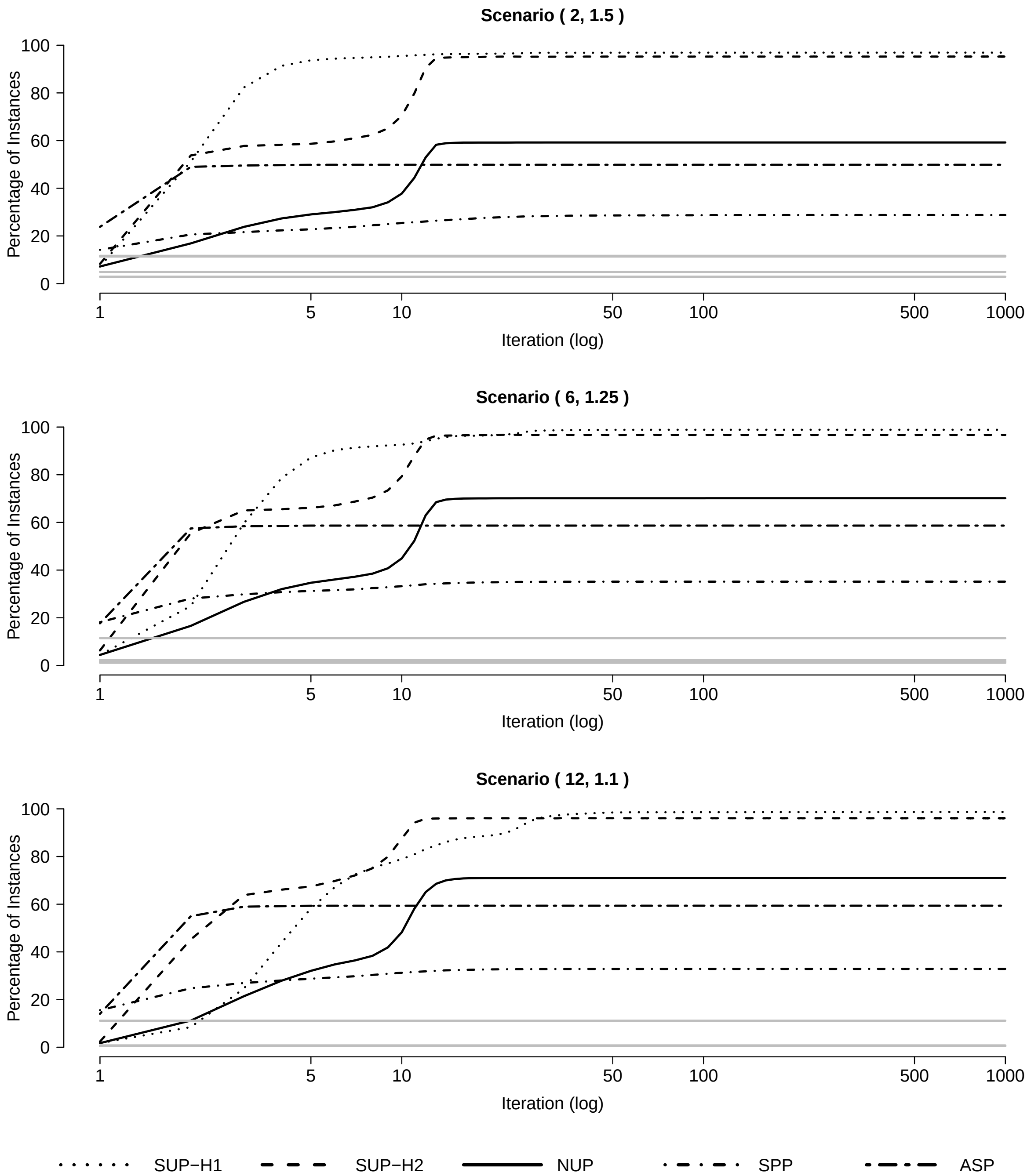

Figure 7.4.: Percentage of Central Optimal Shadow Prices – Exact Updates

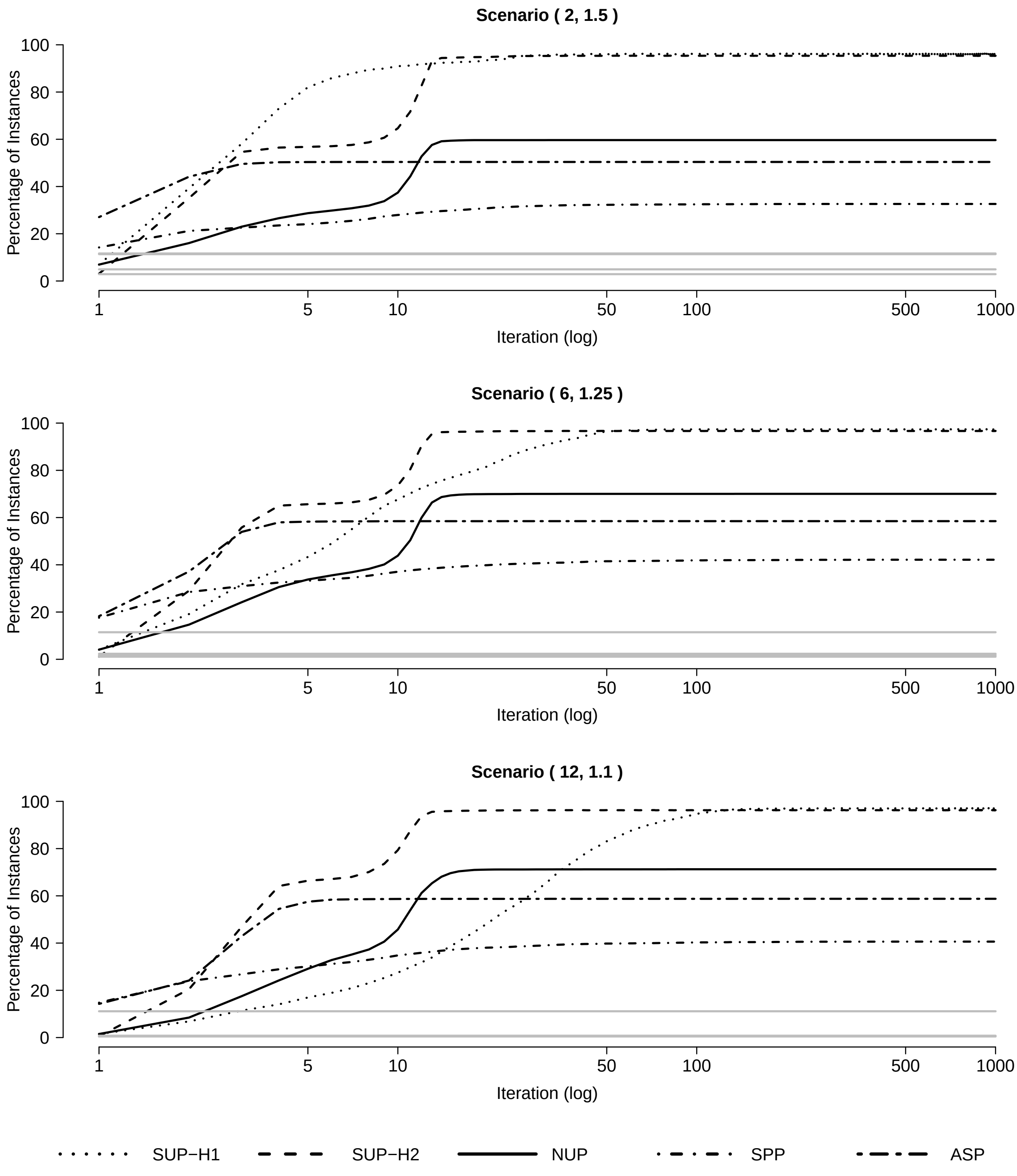

Figure 7.5.: Percentage of Central Optimal Shadow Prices – Approximate Updates

As in Section 7.2.1, we examine next to the optimal shadow prices also the MAPE between the central and the local solution. Its convergence is depicted in Figures 7.6 and 7.7. The gray lines correspond again to the static schemes and besides SPP their performance is strictly worse than those of the dynamic schemes. As in the previous graphs, SPP hardly improves the initial results and it only outperforms the static schemes in the large scenarios with approximate updates.

Regarding the other schemes, we see that the initial valuations lead to an average MAPE of about 15 percent for most methods and most scenarios. So, in terms of the MAPE, all schemes begin at about the same level. In the subsequent iterations, NUP and ASP as well as SUP-H1 and SUP-H2 approach the same levels. NUP converges to a value just below ASP and provides a lower bound on the MAPE of ASP. Similarly, the heuristics reach about the same final value, but as SUP-H1 has the lower convergence rate, it needs more iterations and approaches the result of SUP-H2 from above.

Opposed to the previous indicators, the MAPE of ASP diverges. It swings up and down between different solutions. From Figures 7.2 and 7.3 we see that the average updates are constant, indicating that the MAPE oscillates between the same solutions. Furthermore, the oscillation effect reduces the larger the scenario. The gap decreases from four percent in the scenario with two booking classes to less than 0.5 percent in the scenario with 12 booking classes.

We do not depict the convergence of the adjusted MAPE. Its pattern is similar to the one of the regular MAPE, except that the values are higher. As a final remark, note that the MAPE of the local solutions not including the code-share demand amounts to 60 percent. So, all methods including LF as the current industry standard, at least halve the deviation to the central shadow prices.

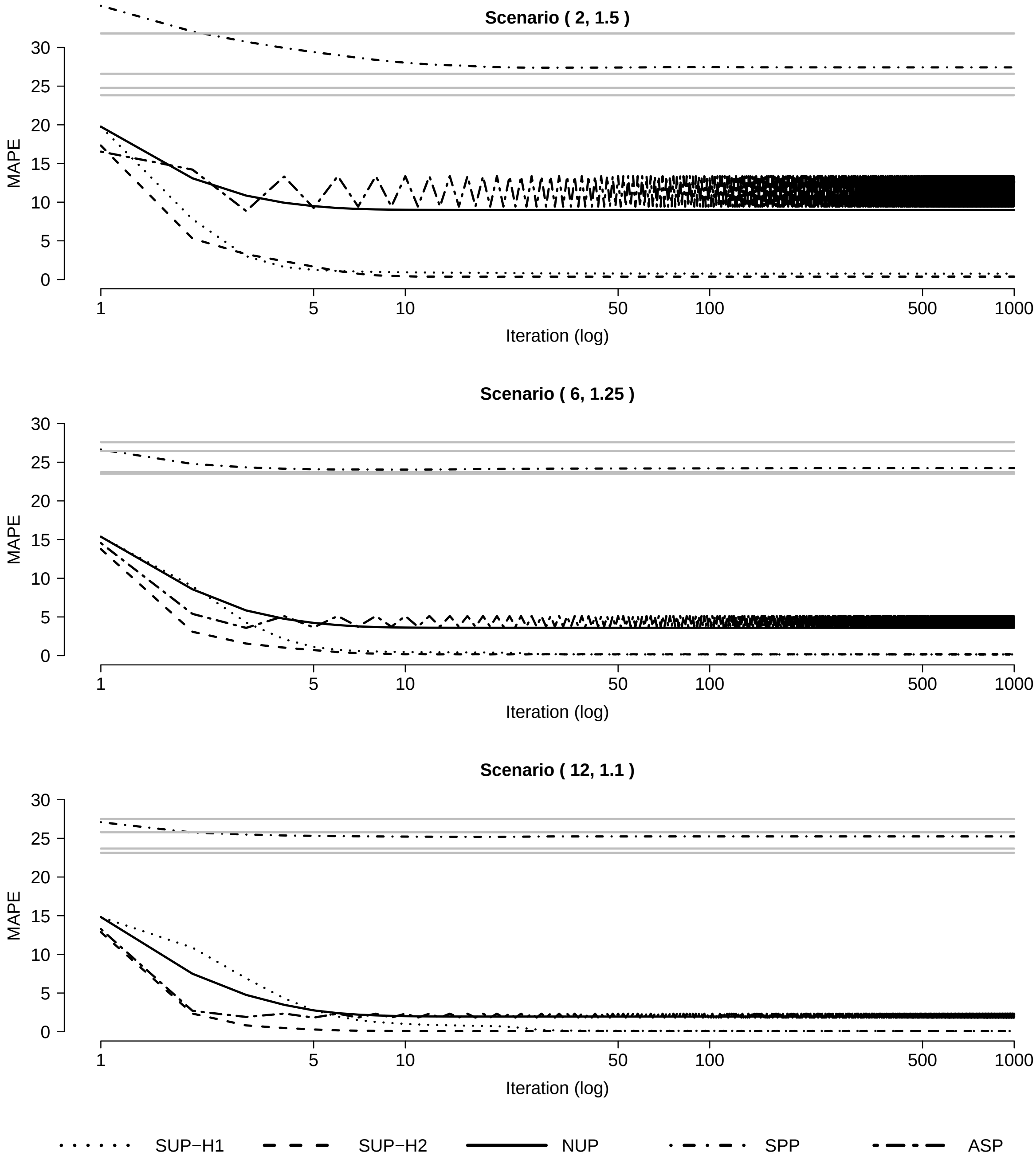

Figure 7.6.: MAPE – Exact Updates

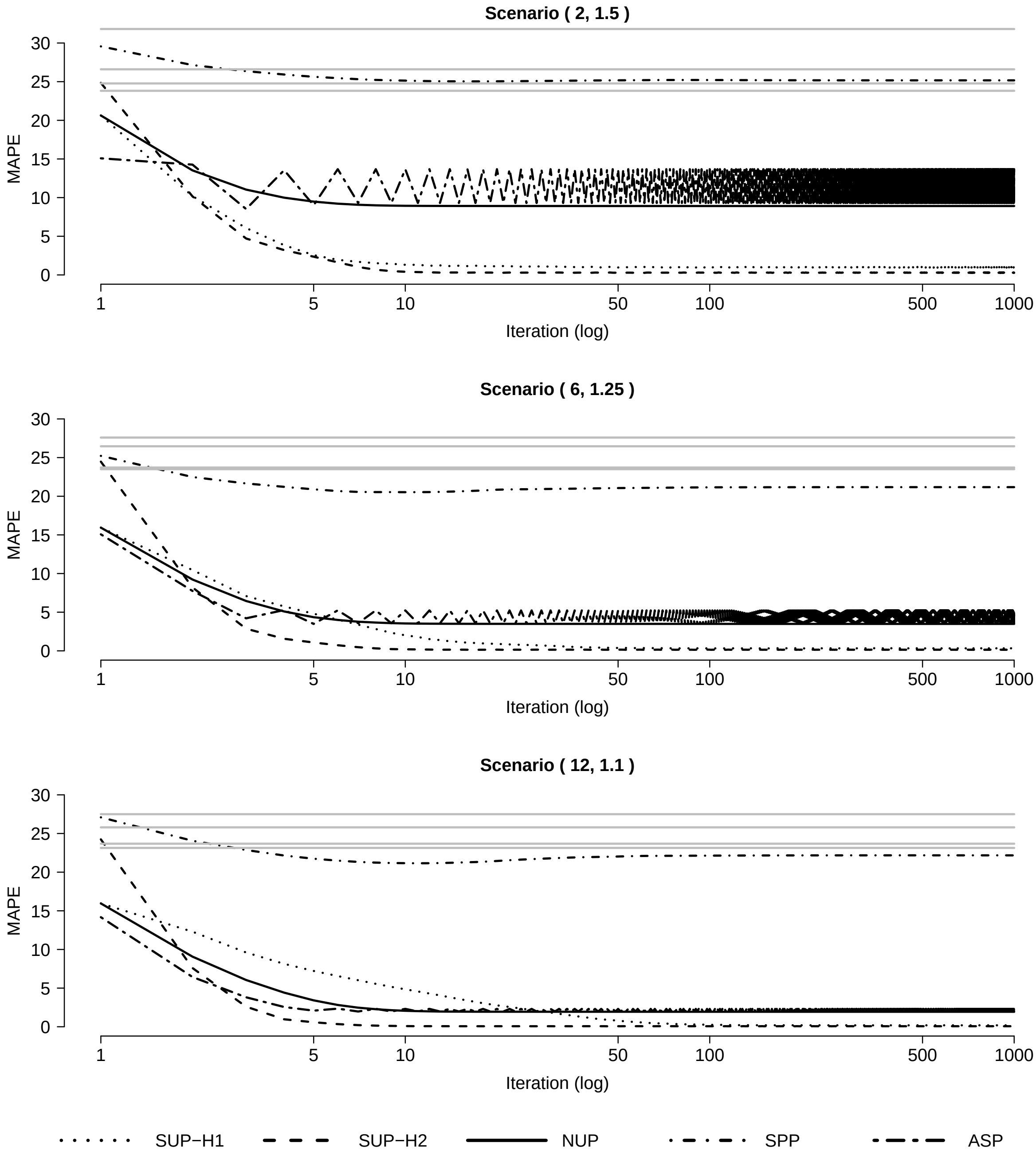

Figure 7.7.: MAPE – Approximate Updates

Finally, we examine the convergence of the aggregate revenue. It is calculated in percent of the central revenue and depicted in Figures 7.8 to 7.10. The results show that SPP has the biggest gap among the adaptation methods. Besides the first iteration of SUP-H2 with approximate updates, it always exceeds the gap of the others and improves only slightly.

The other schemes have higher convergence rates and approach the central solution within 10 iterations with. The initial deviation is minimized by SUP-H1 and NUP, while SUP-H2 has initially the biggest gap. However, it reduces within one iteration to the same level or even below the one of the other methods. Last, we note that the excess revenue reduces with the size of the scenario due to the denser price points.

The alliance revenue for ASP is plotted separately in Figure 7.10. On the contrary to the adaptation schemes, the adjustments in ASP change the aggregate value of the itineraries and therefore, a different scope is needed. Figure 7.10 is divided into two graphs, one for exact and for approximate updates. The results in both cases are very similar: The smallest gap comes from the scenario with two booking classes and it takes at most five iterations to reach the minimum. As with the MAPE, the alliance revenue determined by ASP does not strictly decrease. It diverges and oscillates between two solutions. The impact of this averaged over all instances is 0.17 to 1.15 percent of the central revenue.

Last, we note that the aggregate revenue without code-share demand amounts to about 85 percent averaged over all three scenarios. So, 15 percent of the total revenue is exclusively generated from code-share sales and cannot be compensated with intraline demand.

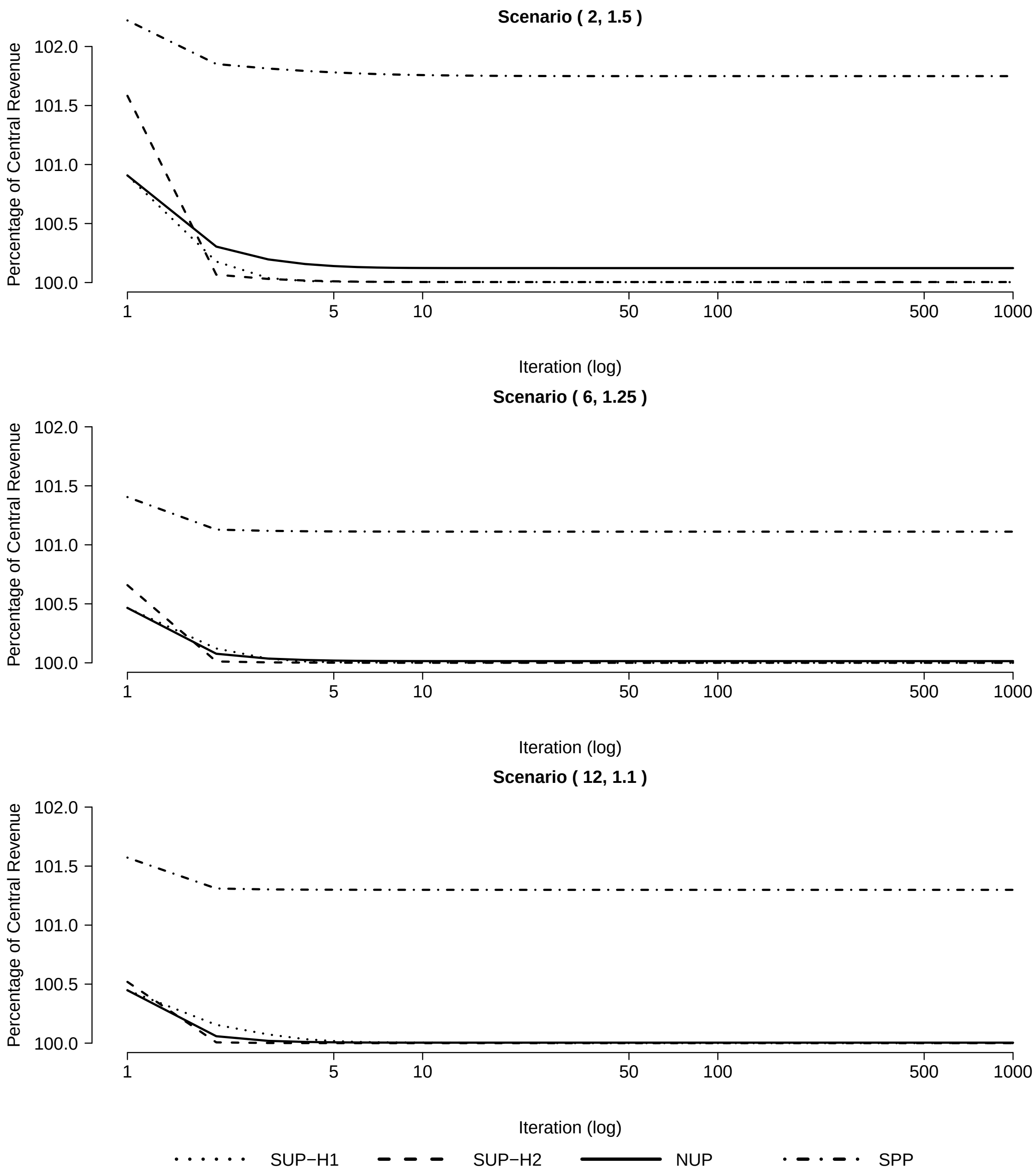

Figure 7.8.: Alliance Revenue – Exact Updates

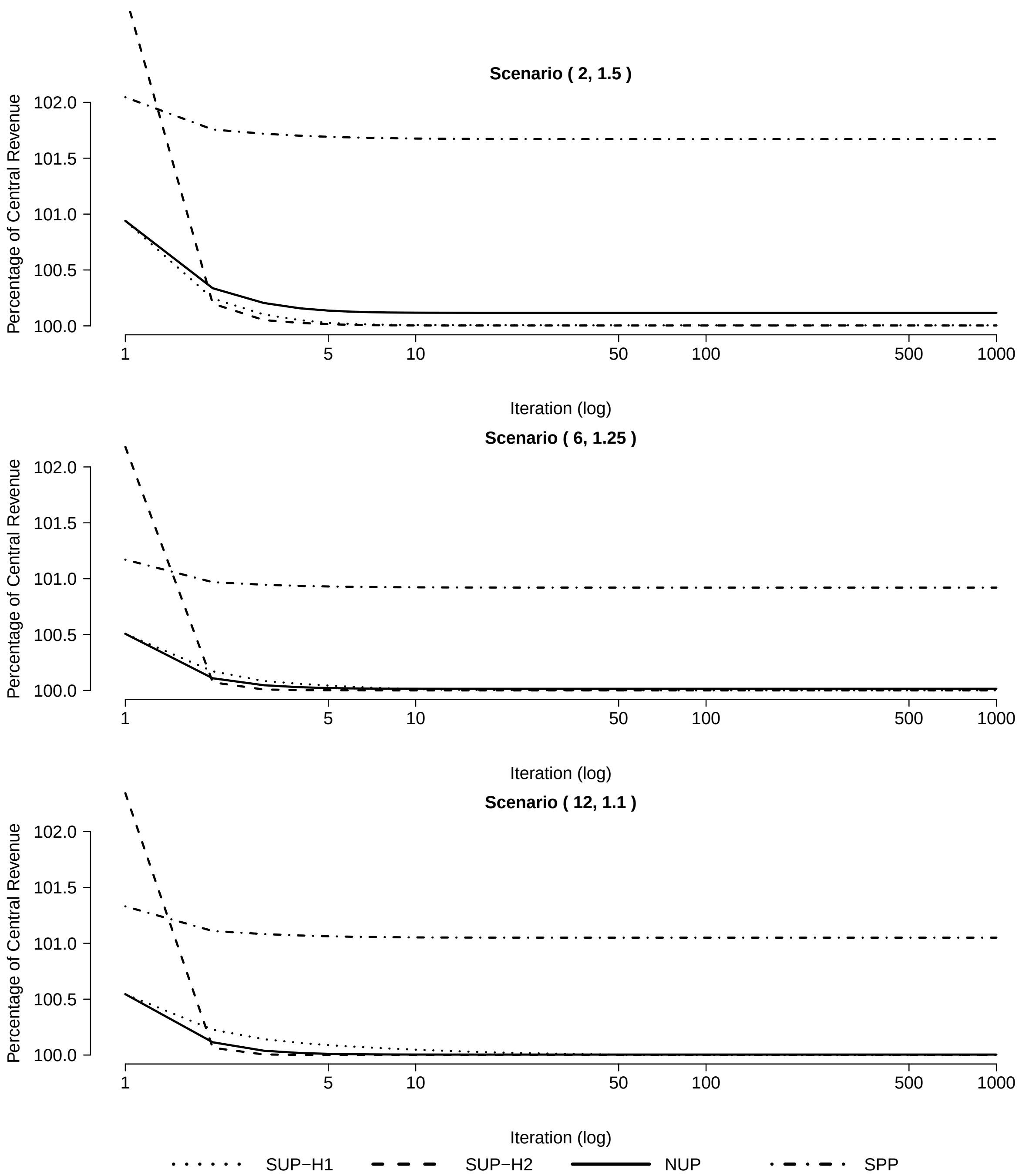

Figure 7.9.: Alliance Revenue – Approximate Updates

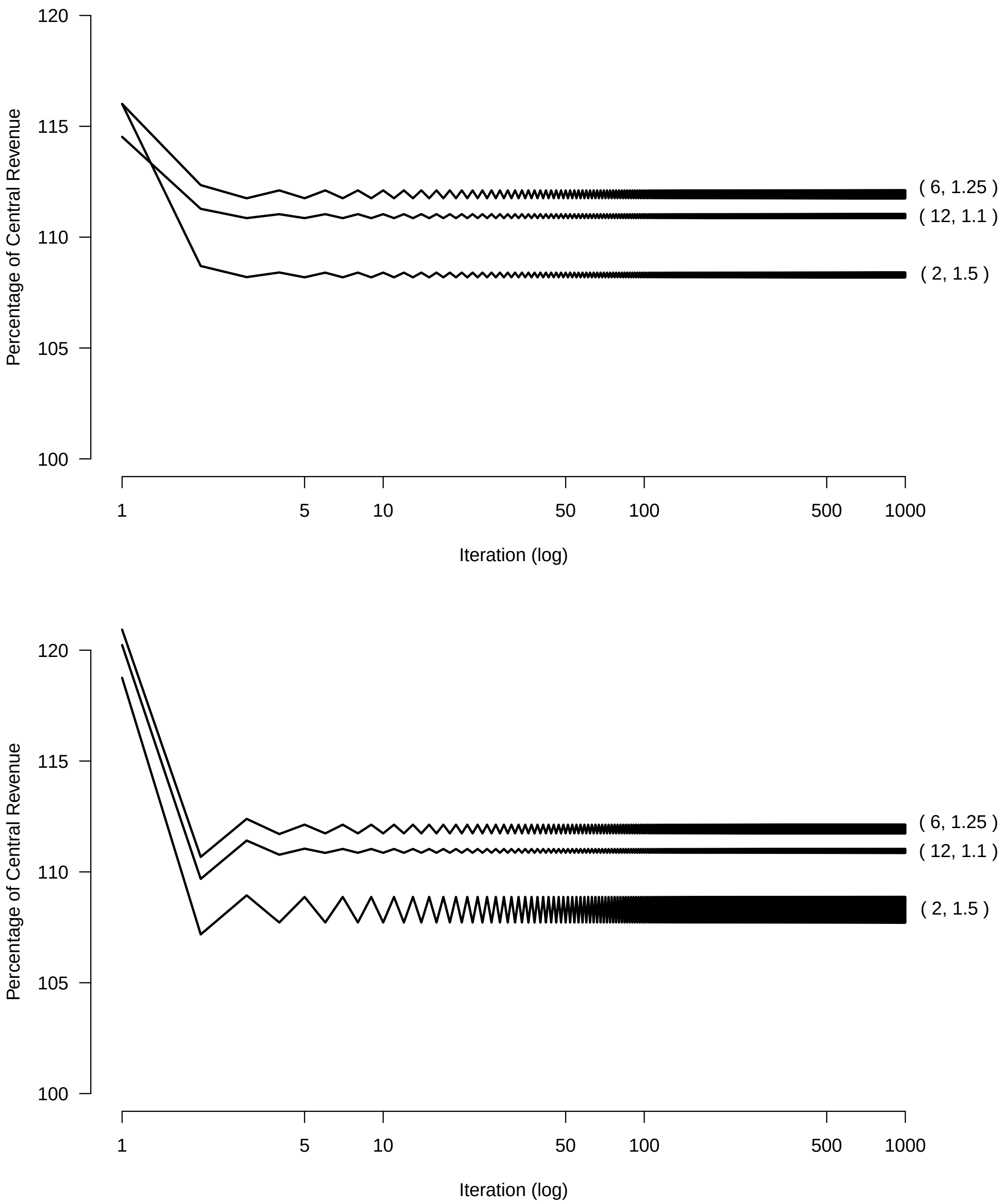

Figure 7.10.: Alliance Revenue – ASP with Exact and Approximate Updates

7.3. Discussion of the Results

Our results show that the two heuristics outperform the other methods. They achieve the highest percentage of central optimal instances and minimize the average deviation between the local and the central shadow prices. Nevertheless, they are heuristic: They do not guarantee to find the optimal outcome or may not converge to a stable solution. Therefore, they are not sufficient in a strict sense.

Comparing both heuristics with each other, we see that they provide similar results. Although SUP-H1 reaches the central optimum in slightly more instances, the MAPE and in particular the adjusted MAPE is significantly lower for SUP-H2. In addition, the second heuristic is computationally less expensive and converges faster. It achieves the highest number of optimal shadow prices after 10 iterations, while the convergence of SUP-H1 degrades with the scenario size. For the scenario with many price points and approximate updates – the situation most likely occurring in practice – SUP-H2 outperforms SUP-H1.

The necessary procedure NUP is simpler to implement, ensures convergence and provides good results. Moreover, its performance improves with the size of the scenario: The more price points there are, the more accurate the local solutions become as the negative effect from borderline allocations diminishes. By further dividing the demand into sufficiently small parts, NUP asymptotically approaches the central solution.

In contrast to NUP, the performance of SPP is disappointing. Although the updates have the same structure, the results are significantly worse and there is no visible effect across the scenarios. Because the optimality conditions are less restrictive, the updating only leads to small improvements and the MAPE is even higher than in some of the static methods. A main issue causing the weak performance of SPP are flights with excess capacity. Their shadow prices are zero and by (6.8) the code-share valuations also become zero such that the shadow prices do not change.

While SPP at least finds a stable solution, ASP causes a large share of instances to diverge, what makes this scheme less predictable. Nevertheless, the quality of the solutions is satisfying and it is as simple to implement as NUP. Also, the results improve with more price points and therefore, ASP may also work well in practice.

Last but not least, the static schemes are outperformed by the dynamic ones. Their results are consistently worse and the gap that they produce is about the same, independent of the scenario size. Their main advantage is their simplicity and predictability. Furthermore, they do not require any information exchange among the carriers.

Across all methods there are only minor differences between exact and approximate updates. Both variants provide similar results and not resolving the linear programs well approximates the exact solution. In some instances, it is even better because exact updates are also not optimal. This observation is important in practice. With hundreds or thousands of code-share itineraries, regular re-optimization becomes impossible and efficiency losses would make the methods unattractive in comparison to other schemes, as for example the static ones.

Another important feature is that the performance of the adaptation schemes (except SPP) increases with the number of booking classes. In practice, airlines use about 20 booking classes and hub-and-spoke networks have dozens of routes traversing an individual flight. As a result, the schemes promise close-to-optimal results in real-world networks.

Overall, the methods show distinct characteristics and may be combined to achieve the best results. Nevertheless, our findings are purely theoretical: External influences such as volatile demand may affect the calculation of the code-share valuations and distort the performance observed in practice.

Part III.

Practical Implications and Simulation Study

8. Practical Implications

The theoretical derivations as well as the numerical experiments provided an indication on the performance of the various valuation schemes. In practice, however, there are several other factors impacting the effectiveness of code-share control methods, as for example the high complexity of the revenue management process as well as the interaction of the individual steps. Due to interdependencies and the heuristic nature of most revenue management methodologies, it may happen that simple procedures work better than theoretical more advanced approaches. This holds for the proration schemes developed in this thesis as well. Before implementing them in the industry, it is essential to identify possible pitfalls and to run extensive studies that verify how well these methods integrate in the existing information systems structures. For this purpose, this chapter provides an overview of aspects related to their practical realization and proposes different variants on how the adaptation process can be implemented. In the subsequent chapters, we use these ideas to embed our approach in a large-scale simulation environment.

8.1. General Considerations

The approach developed in this thesis attempts to imitate the central solution in the local optimization problems of the alliance carriers. In order to actually implement the central solution, however, it does not suffice to find the central shadow prices, but it is necessary to imitate the central procedure at all levels – from the construction of the forecasts to the acceptance of the customer requests.

For the Dynamic Programming Decomposition, for example, this means to imitate the fare adjustment done between the linear program and the dynamic programs. On the one hand, the prorated valuations in the linear program sum up to the full itinerary fare and coordinate the carriers' decisions. On the other hand, displacement adjusted fares are used in the dynamic programs. Implementing the fare adjustment in addition to the fare proration is essential to correctly imitate the central procedure. Without a consistent implementation the benefits achieved at one level may be eroded at another. For example, using one type of fares in both steps over- or underestimates the value of code-share itineraries and leads to deviations in the solutions. So, finding the better shadow prices may not have a positive effect when the same valuations are used in the dynamic programs. It can even worsen the overall outcome.

In the process, the assumption on common demand forecasts, saying that all airlines expect the same code-share demand, is similarly important. Although airlines can easily identify code-share bookings, for example from the passenger name records (Vinod, 2005), they need the same forecasting and unconstraining methodology as well as the same product structure to reach the optimal outcome. In theory this is possible. In practice, however, independent carriers rarely satisfy this requirement. The problem of asymmetric products can be reduced by being granted partial immunity[13] and introducing common fare structures. Moreover, due to the limited number of revenue management solutions, some airlines may end up with similar methodologies. Both helps to align the forecasting processes, leading to common demand forecasts. Without such conditions, coordination schemes on the optimization level are likely to fail.

Another practical challenge associated with the implementation of the adaptation process is the size and complexity of the network. For a large network carrier it can become huge and increases the underlying optimization problem that needs to be evaluated. This

[13] For example the A++ agreement among several Star Alliance members on transatlantic routes.

aspect is particularly important for methods requiring to solve the optimization problem multiple times, e.g. for every code-share itinerary. Such approaches are difficult, if not impossible, to realize in practice and we do not expect them to get implemented in practice given the available technology. Only significant improvements in the solution quality may justify the higher computational load.

To achieve the optimal outcome, it is further necessary to time the airlines' optimization processes and to determine exact updates by frequently resolving the linear programs. With partners spread across the world the first requirement provides an additional coordination problem. The second requirement is unlikely to realize in practice due to the high number of code-share itineraries.

Besides these possible problems, a central feature of the adaptation methods is that they do not require much information. The necessary updating process described by Theorem 4 solely needs the disclosure of shadow prices, while the sufficient procedures require the shadow prices and the capacity allocations. Already today, this information is frequently exchanged among carriers. Capacity allocations can be interpreted as AVS messages and they are currently industry standard for code-share control. Moreover, many alliance partners enjoy partial immunity and may share their bid prices for booking control. Our results showed that this information suffices to apply sophisticated coordination schemes and to achieve close-to-optimal results.

8.2. Implementation of the Adaptive Valuation Schemes

This section proposes different variants for the implementation of the valuation schemes from Chapter 6. We describe how the information exchange among the carriers is organized and discuss their advantages and drawbacks.

8.2.1. Variant 1 – Iterative Updating

The first variant directly imitates the updating procedure from Chapter 6. Both airlines iteratively solve the linear programs and exchange the shadow prices in between. For practical reasons the number of iterations may be restricted e.g. to 10. Our experimental results in Chapter 7 show that with a large number of price points on every flight, the resulting shadow prices quickly become reasonably close to the optimum. With the final shadow prices the carriers apply the fare adjustment described by (5.2) and solve the dynamic programs. The respective information exchange is schematically depicted in Figure 8.1.

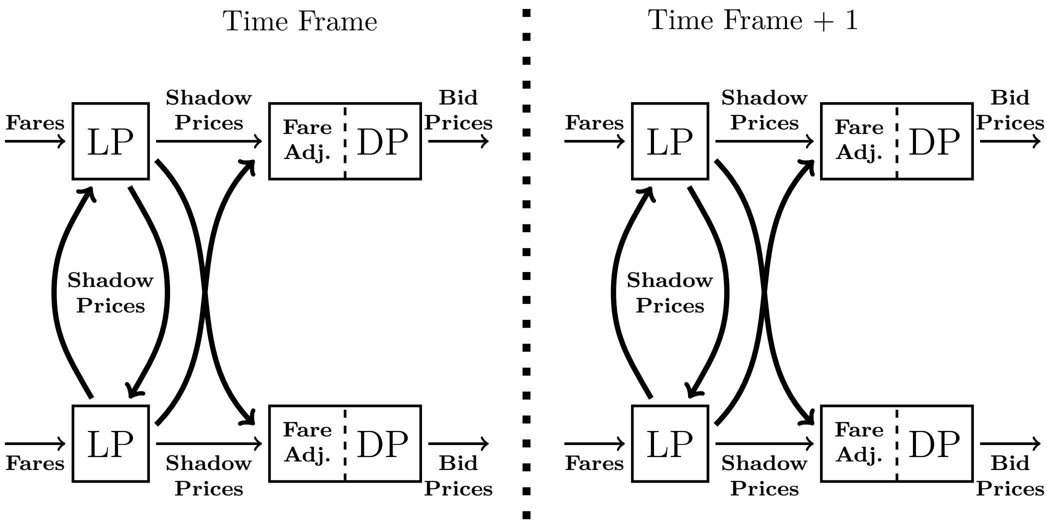

Figure 8.1.: Information Exchange for Code-Share Valuations – Variant 1

The first variant with iterative updates is closest to the ideal process described in Chapter 6, however it is also the most difficult one to implement in practice. It requires the alliance members to align their optimization processes and regularly exchange the shadow prices. Such strong coordination limits the sovereignty of the carriers and is more likely to be used by close partners such as after a merger. Without perfect synchronization, the process does not work and the results may not be accurate.

8.2.2. Variant 2 – Last Shadow Prices

Instead of using the shadow prices from the same optimization step and solving the linear programs iteratively, the second variant uses the last shadow prices – the ones determined during the previous optimization round. Following the idea in Jain (2011), the shadow prices from the last optimization approximate the updated shadow prices and can be used for the proration of the code-share fares before the next optimization. Assuming that the shadow prices remain relatively constant over time[14], convergence may happen over the booking horizon and between the runs. Figure 8.2 shows the respective information flow.

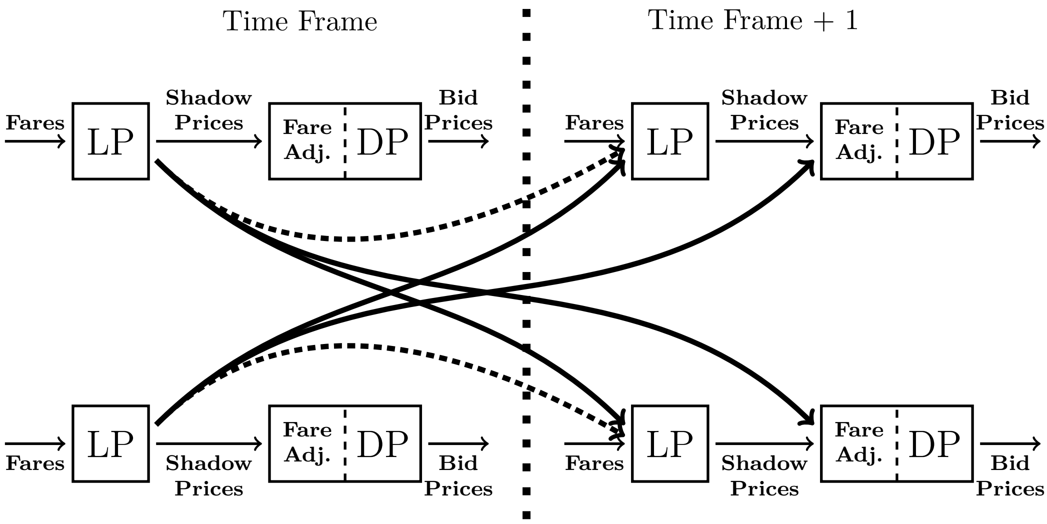

Figure 8.2.: Information Exchange for Code-Share Valuations – Variant 2

The second variant is less accurate than the first one due to the time lag. Nevertheless, for the same reason it is far easier to implement in practice. The airlines do not need to coordinate their revenue management processes and simply approximate their partners current status with the information from the last optimization. Besides the timing issue, the linear programs are not solved iteratively in the this variant, what makes this approach computationally less expensive.

[14]For example see the martingale result in Akan and Ata (2009) as well as the respective discussion in Wright (2010, p. 63).

8.2.3. Variant 3 – Stochastic Bid Prices

The third variant uses the stochastic bid prices instead of the deterministic shadow prices. The adaptation process is this case remains the same as in Variant 2. The code-share fares are prorated before the optimization starts as well as in the fare adjustment. During the latter step the own shadow prices and the partner's bid prices are used together to adjust multi-leg fares. The complete information exchange is depicted in Figure 8.3.

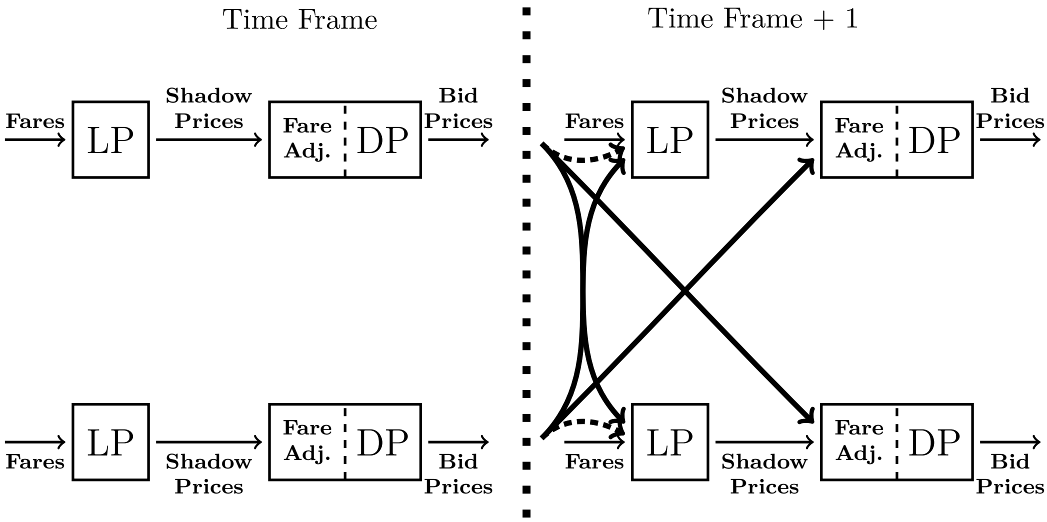

Figure 8.3.: Information Exchange for Code-Share Valuations – Variant 3

Potential inconsistencies may arise because the bid prices are determined by the dynamic programs, which again depend the solution of the linear programs and the fare adjustments. The advantages of bid prices are that they are available at every point in time and do not need to be approximated with old data. The carriers can share them whenever needed and the bid prices incorporate the latest booking information. Furthermore, the carriers that use BPS for code-share booking control, already exchange their bid prices and there is no additional information sharing.

8.2.4. Variant 4 – AVS Allocations

The fourth variant uses neither the bid prices nor the shadow prices and is intended for carriers with AVS-based code-share control. As AVS messages contain booking class availabilities, the code-share valuations should be updated whenever the availabilities on the individual code-share flights do not coincide. In the process, it is necessary to estimate the current bid prices from the AVS messages. One possible approach for this is to set the bid prices equal to the fare of the lowest available class. Having estimated the partner's bid prices, the necessary update rule can be applied as in the regular case, but with the estimated bid prices instead of the real bid prices or the shadow prices.

8.3. Implementation of the Alternative Schemes

The implementation of static valuation schemes is much simpler. The fares or the flight distances are publicly known or at least simple to estimate. Each carrier can collect this information and individually prorate the code-share fares. The resulting valuations are used throughout the optimization process, i.e. in the linear program as well as the dynamic programs, and do not change. Code-share itineraries are thereby treated as genuine intraline itineraries.

The only dynamic scheme proposed in the literature is the Dynamic Valuation policy by Jain (2011). It corresponds to the absolute shadow price adjustment mentioned in Chapter 7, except that it uses bid prices and not shadow prices. As for the static schemes, Dynamic Valuation makes only a single adjustment at the beginning. The resulting fares are used during the entire optimization process. The suggested implementation possibilities for the adaptive schemes, in contrast, prorate the code-share fares once before the optimization begins and a second time during the fare adjustment between the linear program and the dynamic programs. For a more detailed description of Dynamic Valuation, the reader may consult the work by Jain (2011).

9. Revenue Management Simulator REMATE

REMATE ('Revenue Management Training for Experts') is a comprehensive simulation environment developed by Lufthansa German Airlines and several cooperating universities (Frank et al., 2010; Cleophas, 2012; Gerlach et al., 2013). Its design is based on the principles presented in Frank et al. (2008). To our knowledge, the only comparable revenue management simulator is the *Passenger-Origin-Destination-Simulator* (PODS) operated at the Massachusetts Institute of Technology (Gorin and Belobaba, 2004; Hopperstad, 2000; Lee, 1998; Skwarek, 1997; Boing, 1993).

Revenue management simulations are important to evaluate forecasting and optimization techniques in realistic conditions. By generating artificial passenger demand, the performance of different control methods can be compared across various market situations. This reduces the risk of unexpected outcomes and may expose possible pitfalls, which are crucial to know for a successful implementation in practice.

REMATE is one of the most sophisticated simulators in the industry. Besides research, it is used for training purposes and serves as a decision-support tool for market analysts. It supports various state-of-the-art forecasting and optimization methods as well as other features such as cancellations, no-shows and point-of-sale distinction. Similar to PODS, REMATE utilizes a complex customer-choice model that allows passengers to valuate and compare travel options based on a set of preferences. Moreover, many common user influences are supported that enable user interaction with the system, as it is frequently observed in practice. In the following paragraphs, we provide a brief description of the REMATE architecture and in particular the simulation setup and scenarios used in this thesis.

9.1. General Architecture

REMATE simulates the interaction between customers and airline revenue management systems. Its general architecture is schematically depicted in Figure 9.1.

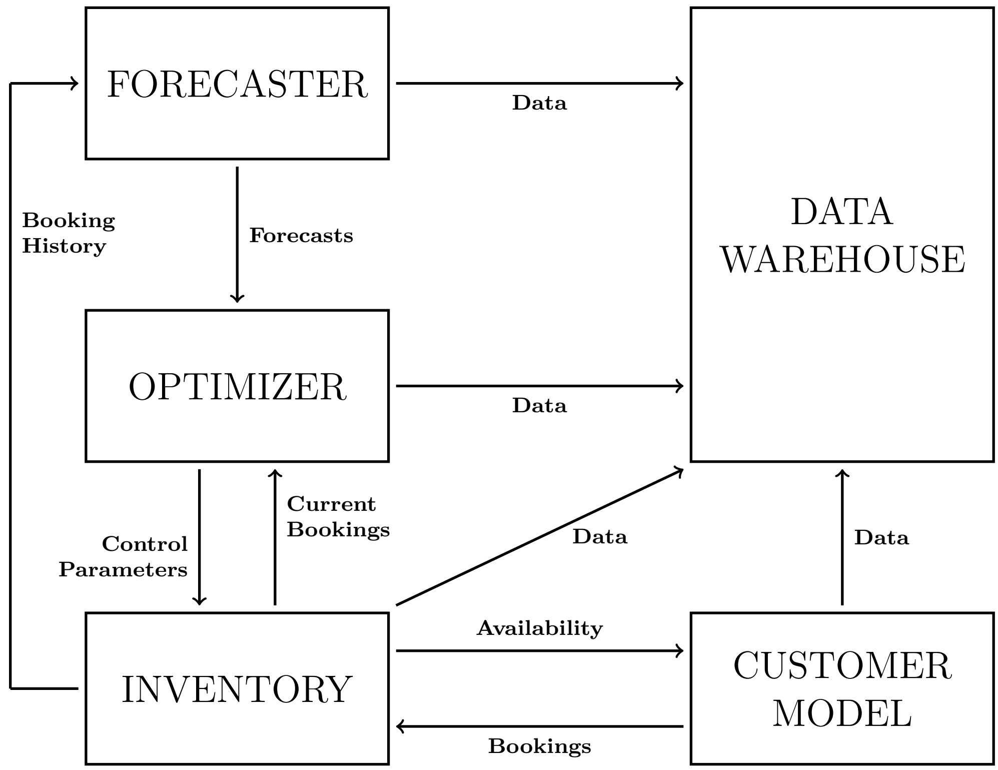

Figure 9.1.: Schematic Structure of REMATE (cp. Zimmermann et al., 2011))

The first step (not depicted in Figure 9.1) is the setup of the supply. Supply covers the flight schedule, the booking classes, prices, and aircraft capacities. This information is uploaded by the user and provides the basis for the subsequent steps.

Given the supply information, the airlines initialize their revenue management systems and passenger requests are generated according to the customer characteristics defined in the customer model. In the process of the simulation, the customers arrive and consume the capacity offered by the airlines. The airlines observe the booking behavior and update their control decisions accordingly. For that purpose, they utilize full-scale revenue management systems, modeling the complete process from data preparation to final availabilities. All data generated and used throughout the simulation is stored in the data warehouse.

REMATE simulates flights on up to seven consecutive departure days. The booking period for every flight is 360 days and is divided into 24 time frames, also called *data collection points* (DCP). The first DCP is at 360 days-before-departure (DBD). The 23rd is at DBD 0 (the departure day). The 24th is the actual departure of a flight. At each DCP except the 24th, forecasts are updated and control parameters optimized.

The customer model in REMATE generates random customers request on product level. The total number of customers is divided over distinct customer types and changes in every run following a normal distribution. Every customer type has unique characteristics, which determine the choice behavior among the acceptable and affordable products. The most important characteristics include the average *willingness-to-pay*, the *preferred departure time*, the *time of the request*, the *cancellation probability* as well as the *disutilities against product restrictions*.

- **Willingness-to-Pay:** The willingness-to-pay of a customer states the maximum price that this customer accepts for the requested product. Offers with higher prices are immediately rejected. The willingness-to-pay of an individual customer is normally distributed around the mean of the respective customer type.
- **Preferred Departure Time:** Every customer has a preferred departure time and only accepts flights departing within this departure window.

- **Request and Cancellation Time:** The request and cancellation dates are the days before departure when customers request or cancel their tickets. Cancellations at DCP 24 are considered no-shows. Whether or not a customer cancels the ticket is determined randomly by the cancellation probability, which is the average cancellation rate for customers of a specific type. Arrival and cancellation patterns are defined on customer type level.

- **Disutility Against Restrictions:** Disutilities govern the purchase decisions whenever customers face multiple feasible options. For every individual customer and every restriction the disutilities are independently drawn from a normal distribution and the mean disutilities are defined in the customer type.

Simulations are initialized with arbitrary forecast values and then executed over several runs. Every simulation run simulates the complete booking horizon of all departure days and all flights. Between successive runs the history building updates the forecast references such that the initial forecasts are progressively replaced by the actual observations in the simulation. To avoid that the initial forecasts distort the outcomes, the first runs can be discarded and only the final runs may be used as results.

REMATE supports four different forecasting methods, namely flight, network, hybrid and market-sensitive forecasting. Optimization-wise, two methods are implemented: EMSRb for flight optimization and Dynamic Programming Decomposition for network optimization. Inventory control can either be done with protection levels or with bid prices. The main characteristics of all these methods are described in Section 2.2.

After a simulation has terminated, different indicators can be displayed on various levels. The most important ones, which are also widely used in practice as well as in this thesis, are bookings (BKD; the number of booked passengers), revenue (REV; the total money that customers pay), seat load factor (SLF; the bookings on flight level over capacity) and yield (the average revenue per booked passenger).

9.2. Scenario Setup

This sections presents of the setup of the scenarios used in this thesis. It contains four subsections covering the supply, the demand, the revenue management model and the alliance setup.

9.2.1. Network Design and Product Structure

Our analysis is based on a stylized hub-and-spoke network with two airlines, which we refer to XX and YY, and 12 flights – four long-haul and eight short-haul flights. All flights depart or arrive at a single hub used by both carriers and there are 12 different spoke airports. The two airlines operate halve the flights each and have distinct (complementary) networks such that there is no direct competition. The network is depicted in Figure 9.2.

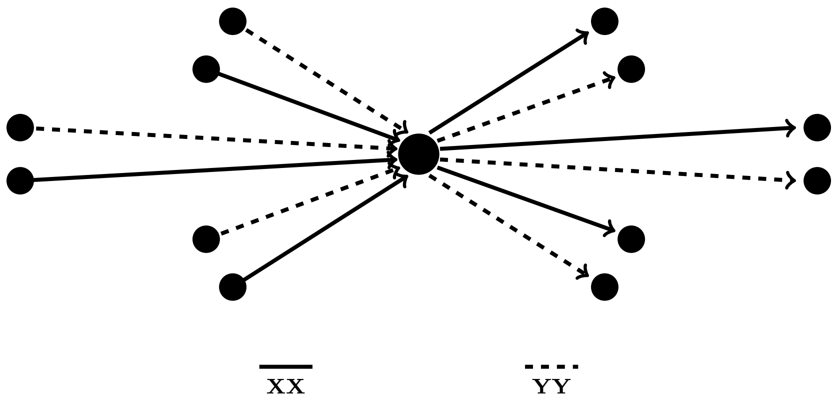

Figure 9.2.: Network with two Airlines and 12 Flights

Each airlines offers 15 intraline itineraries and an additional 18 code-share itineraries arise by connecting the two sub-networks. The capacity on long-haul flights is 400 seats, the capacity on short-haul flights 200 seats. There is a single compartment and six booking classes. We further use five restrictions such that each booking class has a unique combination of restrictions, as shown in Table 9.1.

Table 9.1.: Booking Class Structure in the REMATE Scenarios

Booking Class	R1	R2	R3	R4	R5
A					
B	x				
C	x	x			
D	x	x	x		
E	x	x	x	x	
F	x	x	x	x	x

The prices are based on the distance of the itinerary and a booking class specific multiplier. The multipliers vary between long- and short-haul routes and were derived by regressions over real-world price structures. They consist of a constant and a percentage of the full fare, as depicted in Table 9.2. To get the actual prices, the product of both is multiplied with the square root of the distance.

Table 9.2.: Fare Levels on Long- and Short-Haul Routes

Booking Class	Long-Haul	Short-Haul
A	100%	100%
B	80%	80%
C	65%	65%
D	50%	50%
E	35%	40%
F	20%	30%
Constant	17.0	13.5

Finally, the locations of the airports are chosen such that both airlines fly the same distances. We achieve this by putting pairs of airports on the same geographical location and letting each airline fly to one of them. As a consequence, the airlines have the same prices on these routes and their expected market share across the entire network is 50% in terms of bookings as well as revenue.

9.2.2. Customer Model

We define three different customer types that we refer to as 'high', 'medium' and 'low'. The high-value customer type represents typical business passengers with strong product preferences and high willingness-to-pay. They prefer less restricted products corresponding to the two or three highest booking classes. The low-value customers are the exact opposite and represent the leisure segment. They have a low willingness-to-pay and look for the cheapest prices. Consequently, they solely buy the two or three lowest booking classes. The medium customer type is a combination of the previous two. Depending on the available products and the randomly chosen disutilities, the customers may either buy the cheapest product or pick from the set of affordable products. This customer type covers the medium priced products and represents a mixture of low-value business passengers and high-value leisure passengers. Customers in this segment are difficult to distinct and their decisions vary a lot with the available products as well as personal preferences.

Next to the differences in willingness-to-pay and the product choice behavior, we differentiate between demand for long- and short-haul routes. First, long-haul passengers tend to book earlier than short-haul passengers. Second, the customer preferences vary with the type of route. For example, high-value customers have stronger product preferences on long-haul flights than on short-haul flights. Last, the product preferences are adjusted to the different price structures used in the respective markets.

The customer model is defined by a combination of the three customer types and a demand factor. The demand factor states the total expected demand relative to the capacity offered by the airlines. We create three typical market situations. They differ by the distribution of demand over the three customer types and the choice of the demand factor. We call them 'High Value Business', 'Average Performer' and 'High Volume Leisure'. All three have excess demand and the demand distribution varies from business to leisure oriented. The exact setup of the markets is shown in Table 9.3.

Table 9.3.: Demand Constellations in the Three Markets

Market	Demand Factor	Customer Type Distribution		
		High	Medium	Low
High Value Business	1.2	35%	35%	30%
Average Performer	1.5	20%	40%	40%
High Volume Leisure	1.8	10%	35%	55%

The total number of customers determined by the above mentioned parameters must further be distributed over the various routes. This is done with the percentages shown in Table 9.4. 'LH' thereby stands for long-haul flights and 'SH' for short-haul flights. The combination 'LH-SH' refers to an itinerary consisting of a long- and a short-haul flight. In the second column, we translate the percentages into the average number of requests given the capacity on the respective route type.

Table 9.4.: Distribution of Customers over the Different Routes

Route Type	Percentage of Customers	Average Number of Requests
LH-LH	10%	20
LH-SH	50%	50
SH-SH	20%	10
LH	40%	160
SH	30%	60

To make a booking, the customers search for available products below their maximum willingness-to-pay. The value of a product corresponds to its price plus the sum of all disutilities associated with the restrictions of the booking class. Among these options the customers pick the one that minimizes their personal disutility. If the aggregate disutility exceeds their willingness-to-pay, they do not accept the product. If there is no feasible travel option at all, the customers do not buy any ticket and they are lost for the airlines. There are no second requests.

To simplify the model, there are neither cancellations nor no-shows. As we focus on the optimization and valuation of code-share itineraries, these features would complicate our analysis and the results would depend on how well cancellations and no-shows are handled in the systems, for example how well they are forecasted.

The setup of the customer model was motivated through various studies conducted by market analysts and revenue management experts at Lufthansa. Furthermore, industry data from Lufthansa was used to calibrate the customer model with respect to the booking class mix and the booking curve.

9.2.3. Revenue Management Method

The forecasting method in our scenarios is a standard network forecast. It assumes independent demand and determines separate forecasts for every booking class and every itinerary. The initial forecasts are in the form of random values determined by equally distributing the total expected number of passengers over all products. The history building uses exponential smoothing to update the reference forecasts and progressively replaces the initial values such that they do not defer the results. The optimization method is the Dynamic Programming Decomposition described in Chapter 5.

The inventory of both carriers is controlled with bid prices. There is a bid price determined for every flight and a customer is excepted if its value exceeds the sum of the bid prices for the seats that it consumes. Additive bid prices are common among network carriers in practice (Talluri and van Ryzin, 1998). Moreover, the dynamic programs in the optimization directly return bid prices and we do not convert them to protection levels.

9.2.4. Code-Sharing

The implementation of code-sharing in REMATE is closely adapted from industry practice. It follows the general procedure outlined in Section 3.2 with airlines separately forecasting and optimizing code-share itineraries. They observe the same code-share demand and apply the same forecasting and history building logic with the same parameters. As demand is assumed to be independent, the code-share forecasts thereby match without exchanging them or fixing them to a common value. In addition, the separation of local and code-share itineraries allows to assign different valuations to code-share itineraries.

The code-share flights and itineraries are defined by the user in the supply. With respect to the customer model, it remains the same and customers on code-share routes have the same characteristics as intraline customers. When both airlines offer the same itinerary for the same price, customers are indifferent among the offers and choose randomly with equal probability.

Both availability exchange methods, AVS and BPS, are available in REMATE and we refer to Section 3.2.3 for their description. Whenever a customer request arrives, the airlines exchange their current availabilities and the marketing carrier decides upon the aggregated availability whether a customer is accepted or rejected. The marketing carrier's accept-reject decision is based on single availability (as opposed to dual availability) and the operating carrier has no veto opportunity.

As the implementation of code-sharing in REMATE imitates several important aspects from current industry practice, our findings can be transfered back to real-world markets. Furthermore, the overall scope of REMATE as well as its features provide a good basis for a more realistic analysis than the one presented in Chapter 7.

9.3. Valuation Schemes

After having explained the general structure of REMATE as well as the setup of the scenarios, this section discusses the implementation of the valuation schemes. We implement three dynamic ones and four static ones. All dynamic schemes are based on the information exchange pattern depicted as Variant 3 in Section 8.2. The carriers exchange their latest bid prices and use them to update the code-share valuations and to do the fare adjustment. Moreover, they use BPS to control the code-share booking process, while for the static schemes we apply both inventory control methods (AVS and BPS) and compare their performance. The seven valuation schemes are:

- Necessary Update Rule (NUP)
  The valuations are updated with the necessary rule and equal surplus sharing.
- Bid Price Proration (BPP)
  The full fare is prorated by the ratio of the current bid prices.
- Absolute Bid Price Adjustment[15] (ABP)
  The full fare is adjusted by the current bid price of the partner.
- Local Fare Valuation (LF)
- Local Fare Proration (LFP)
- Distance Proration (DP)
- Full Fare Valuation (FF)

The four static schemes – DP, LFP, LF and FF – are implemented by uploading the respective valuations in the supply of the scenarios. As there are no updates, these valuations remain the same over the entire booking horizon and all runs. The valuations are used in the linear programs as well as dynamic programs.

[15] This method corresponds to the Dynamic Valuation scheme proposed by Jain (2011).

For the dynamic schemes both carriers exchange their bid prices each time they start their optimization. They calculate the prorated valuations using one of the three methods and run the linear programs. Afterwards they do the fare adjustment as before, but for code-share itineraries they take the full itinerary fare and subtract the current bid price of the partner. Thereby, they imitate the central process, but approximate the missing shadow price with the bid price (Variant 3).

The full itinerary fares observed by the customers are the same ones throughout all methods. This ensures that the customer behavior is not influenced by the valuation schemes. The proration methods solely impact the optimization result – first the shadow prices and then the bid prices. For customers only the availability of the products changes. In case of AVS, the availabilities are determined by comparing the prorated fares against the local bid prices. In case of BPS, the full itinerary fare is compared against the sum of the bid prices. As both carriers use the same code-share fares and the same bid prices, their offers always match.

The sufficient procedure is not implemented. Our results in Chapter 7 show that the quality of the shadow prices improves less, the more price points there are. Therefore, it does not promise superior results in the practical setup, but would increase the computational load. Last, we restrict our analysis to approximate updates, i.e. all valuations are based on the same set of bid prices, and we do not simulate exact updates. Our numerical experiments have shown that exact updates do not give significant improvements.

10. Simulation Results and Analysis

This chapter presents the results from the simulation study introduced in the previous chapter. Each of the scenarios is executed over 150 runs. The first 50 are neglected to avoid distortions by the initial forecasts and the results are averaged over the final 100 runs. The customers are generated randomly for every of the three markets based on the parameters defined in the respective customer model. The customer requests remain exactly the same for all valuation schemes, i.e. customers are created for every market and not for every scenario. So, for a specific market all valuation schemes face the same customers with the same request times and the same attributes. Consequently, the results are solely impacted by the valuation of the code-share itineraries and the resulting availabilities and bookings.

Throughout our analysis we use valuation by local fares (LF) with AVS information exchange as baseline. This is the most frequently observed combination in practice and represents the current code-share revenue management logic in the industry (see Chapter 3). The results of all other valuation and information exchange schemes are presented as percentage changes to the base case. After introducing the base case in Section 10.1, Section 10.2 begins with the analysis of the static methods. We examine the results using AVS and BPS. Section 10.3 continues the investigation for the dynamic methods. For all schemes we report the revenue (REV) and the bookings (BKD) from each of the three markets.

10.1. Base Method

The base case corresponds to the current code-share practice in the industry, where code-share bookings are counted, valued and forecasted as local bookings. Because this setup is used as baseline for our analysis, we provide a short overview of the results before we continue. We approximate the gap to the central solution and present the booking curve as well as the booking class mix.

In contrast to the theoretical experiments in Chapter 7, we need to approximate the central solution. The supply of the airlines changes and therefore, the customers have to be generated separately, causing deviations in the requests. The approximated gaps for the three markets are depicted in Figure 10.1. The difference is smallest in Scenario 1 with 5.2 to 6.4 percent, while it increases to 6.2 to 7.2 percent and 6.6 to 7.6 percent in Scenarios 2 and 3, respectively. These gaps are significant and encourage the investigation of improved valuation schemes. For a more in-depth discussion on the performance of code-sharing in practice we refer to the paper by Gerlach et al. (2013).

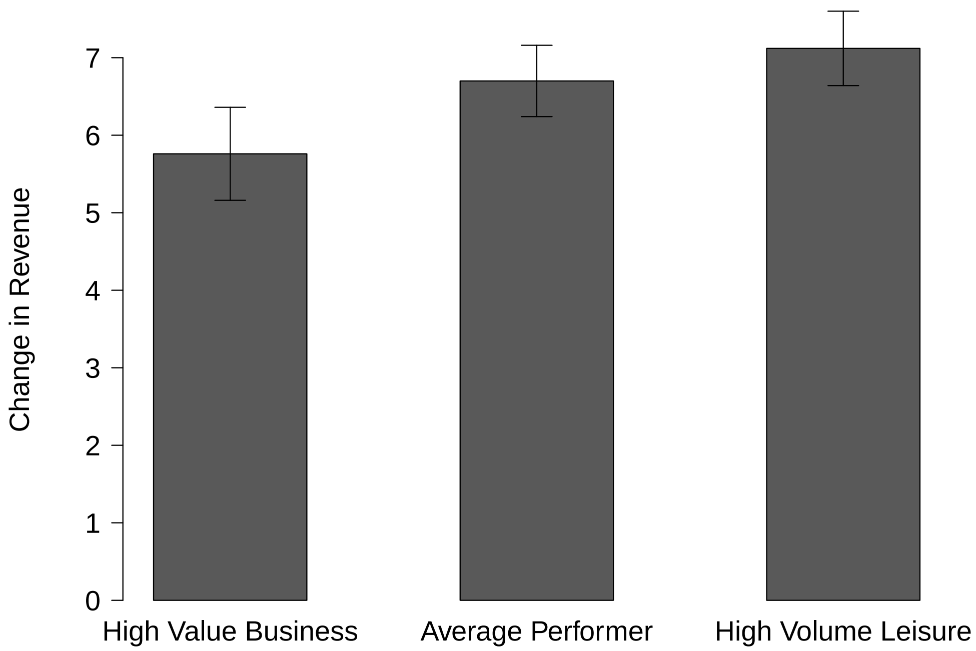

Figure 10.1.: Approximate Gap Between the Central and the Baseline Solution

Next, we look at the booking curves represented by the seat load factors. They are plotted over the DCPs in Figure 10.2.

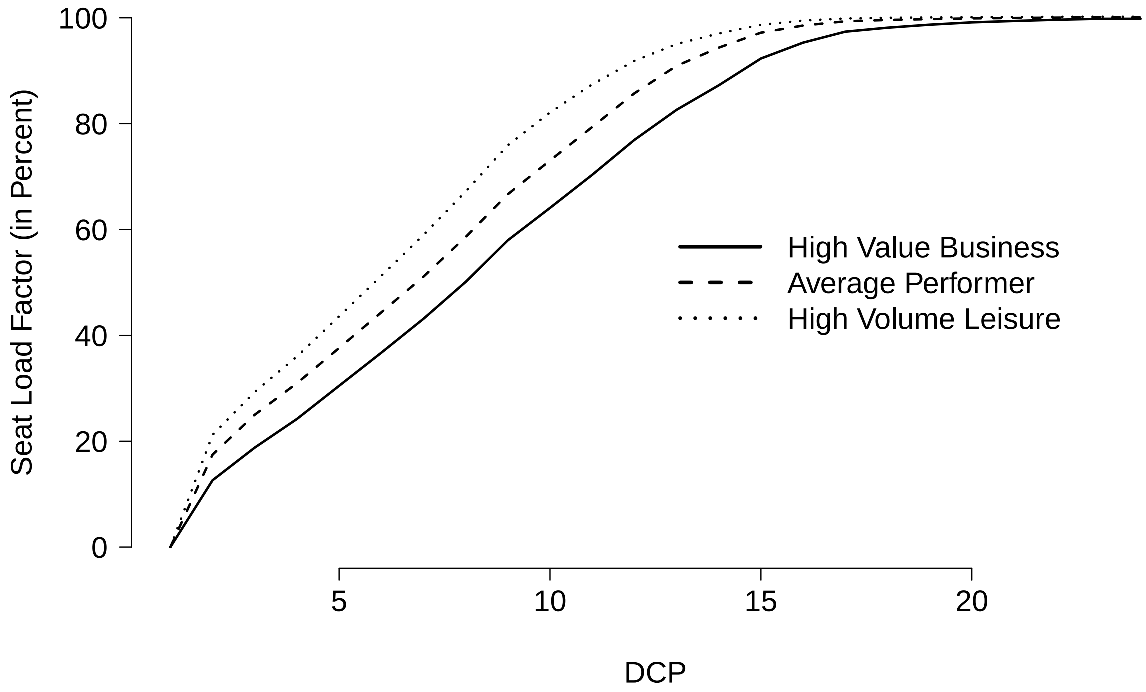

Figure 10.2.: Booking Curves of the Three Scenarios

All three scenarios achieve average seat load factors between 95 and 100 percent. Such high values are due to excess demand combined with the lack of cancellations, no-shows and competition. Moreover, we see that customers in Scenario 3 book earlier. With a high share of leisure demand, many requests arrive at the beginning of the booking horizon, while in Scenario 1 more bookings occur during the last two weeks.

Finally, we look at the booking class mix – the bookings distributed over the six booking classes. Figure 10.3 shows that, not surprisingly, the more high-value demand there is, the more bookings occur in the higher booking classes. Overall, about 60 percent of the bookings occur in the three lowest booking classes and 22 to 28 percent in the two highest classes.

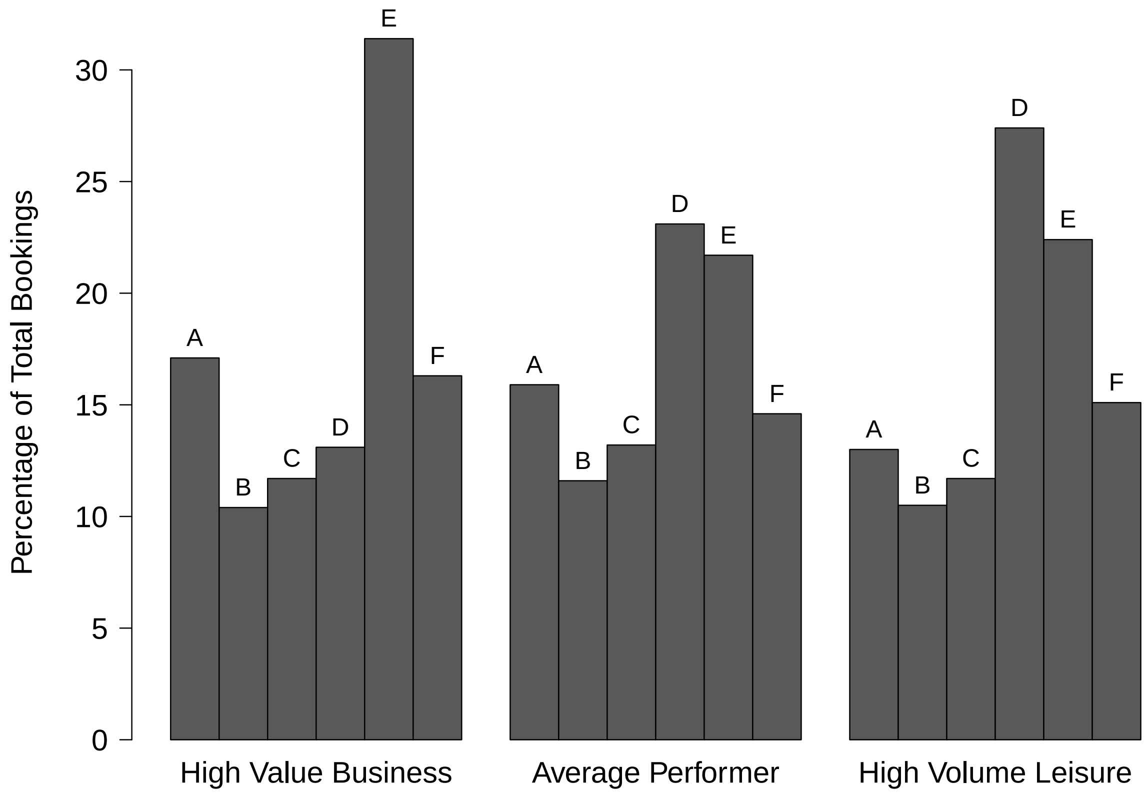

Figure 10.3.: Booking Class Mix in the Three Scenarios

10.2. Static Methods

Having examined the performance of the base cases as well as the differences between the three markets, our analysis begins with the static valuation schemes LF, LFP, DP and FF. The results are divided by the type of availability exchange. Table 10.4 in Sections 10.2.1 depicts the revenue and bookings for AVS information exchange. Table 10.5 in Section 10.2.2 depicts the results for BPS. They are all plotted as percentage changes to the base case using AVS with LF valuations.

10.2.1. AVS Information Exchange

The effect on revenue depends very much on the valuation method. For LFP it is negative in Scenario 1, but becomes positive in the other two. For DP it is consistently negative, for FF consistently positive. The bad performance of DP can be explained by the unbalanced valuations. For short-haul flights the code-share valuations become very small, leading to different availabilities on short- and long-haul flights. As AVS only accepts a passenger when the respective booking class is available on each flight, code-share demand is turned away and the free capacity occupied by less valuable intraline passengers. For the same reason, FF outperforms the base case: It overvalues the itineraries and encourages code-share bookings. Because they are on average more valuable to the network than intraline itineraries, the overall revenue increases. The impact of LFP is ambiguous and relatively small (at most 0.8 percent). In the scenarios with high demand it has a slightly positive effect.

In terms of bookings we see that they behave contrary to the revenue. For LFP and DP they increase, for FF they decrease. This indicates that in the absence of competition, higher code-share valuations lead to more restrictive booking control through higher bid prices. This promotes code-share sales and increases the revenue. As the bookings change sometimes more and sometimes less than the revenue, the effect on yield is ambiguous. In a few cases it increases, in others it decreases.

Last, we note that the biggest revenue changes combined with the smallest differences in bookings occur in Scenario 1. With a high share of business demand and strong product preferences, the impact on revenue is more severe and the valuation schemes are financially riskier. In Scenario 3, on the contrary, bookings change more than revenue. So, with a high share of low-value demand the valuation methods have a stronger effect on bookings.

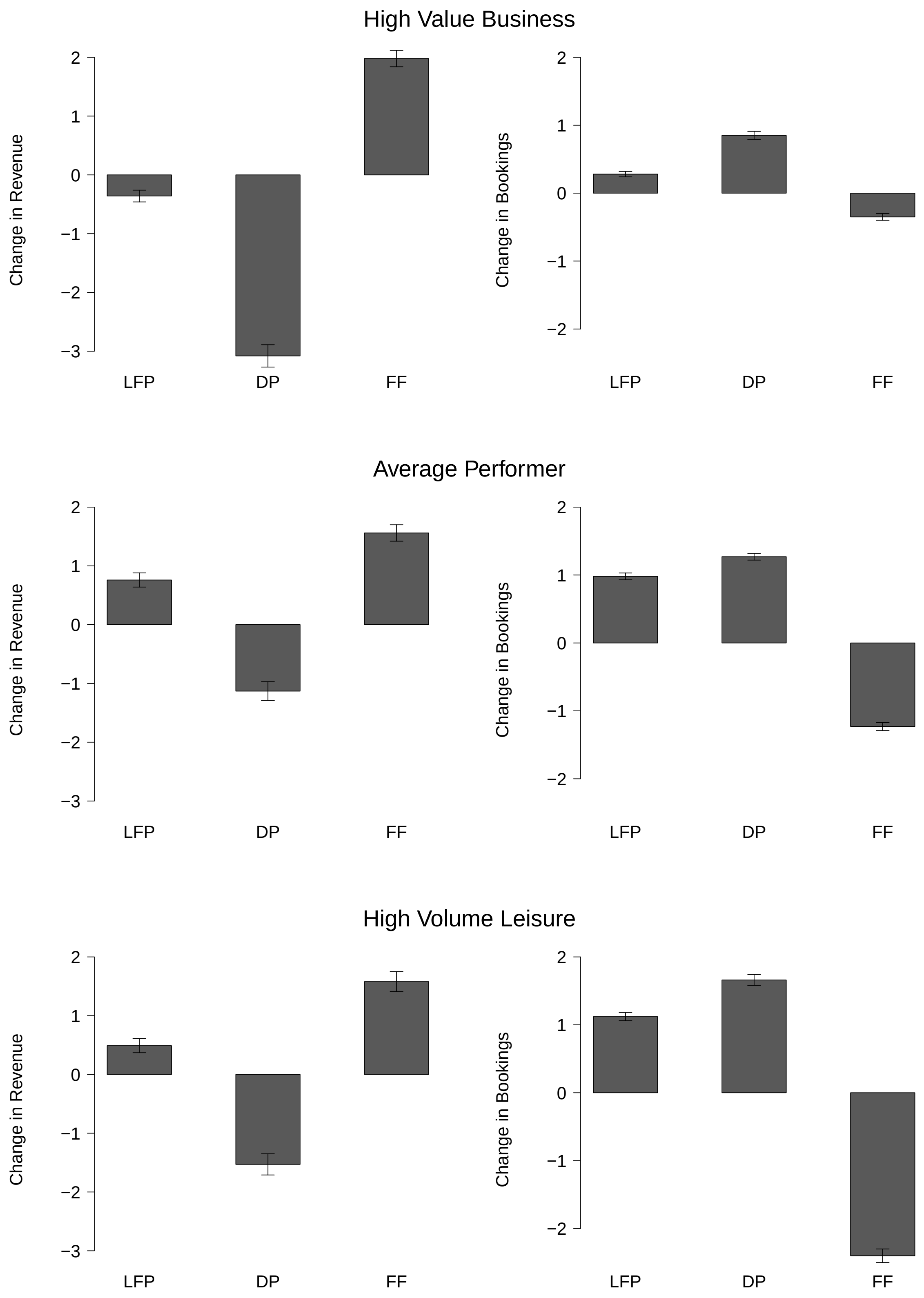

Figure 10.4.: Revenue and Bookings for the Static Valuation Schemes with AVS

10.2.2. BPS Information Exchange

First of all, the revenue increases across all valuation schemes and all scenarios relative to the base case. BPS is more accurate and provides superior results despite the actual code-share valuations. For LF, LFP and DP the improvement is about the same, amounting to 0.3 to 0.9 percent of the base revenue. The biggest improvements come from FF. They reach up to four percent in Scenario 3. As with AVS, the high overvaluation of code-share demand encourages more valuable code-share bookings at the expense of less valuable intraline bookings.

Second, we see by comparing Figures 10.4 and 10.5 that BPS outperforms AVS. For the same valuation method, the revenue determined with BPS consistently exceeds the one from the same scenario with AVS. On average the gap is more than one percent, in some cases even close to four percent. This finding is supported by other authors such as Jain (2011) and Darot (2001) and can be explained with the acceptance logic. While AVS rejects code-share bookings as soon as one carrier closes the respective booking class, BPS compares the code-share fare to the sum of bid prices. The individual bid prices do not matter that much and thereby, BPS balances out unequal code-share valuations such that bad valuations have a lower impact. This makes BPS more robust toward the proration schemes.

Third, the effect on bookings is rather small. It is at most 0.5 percent in Scenario 3 and even smaller − between 0.1 and 0.2 percent − in the scenarios with less demand. So, the revenue effect is mostly driven by shifts in the booking mix. The number of customers remains about the same, but more of them buy a higher class, and consequently, the average yield increases.

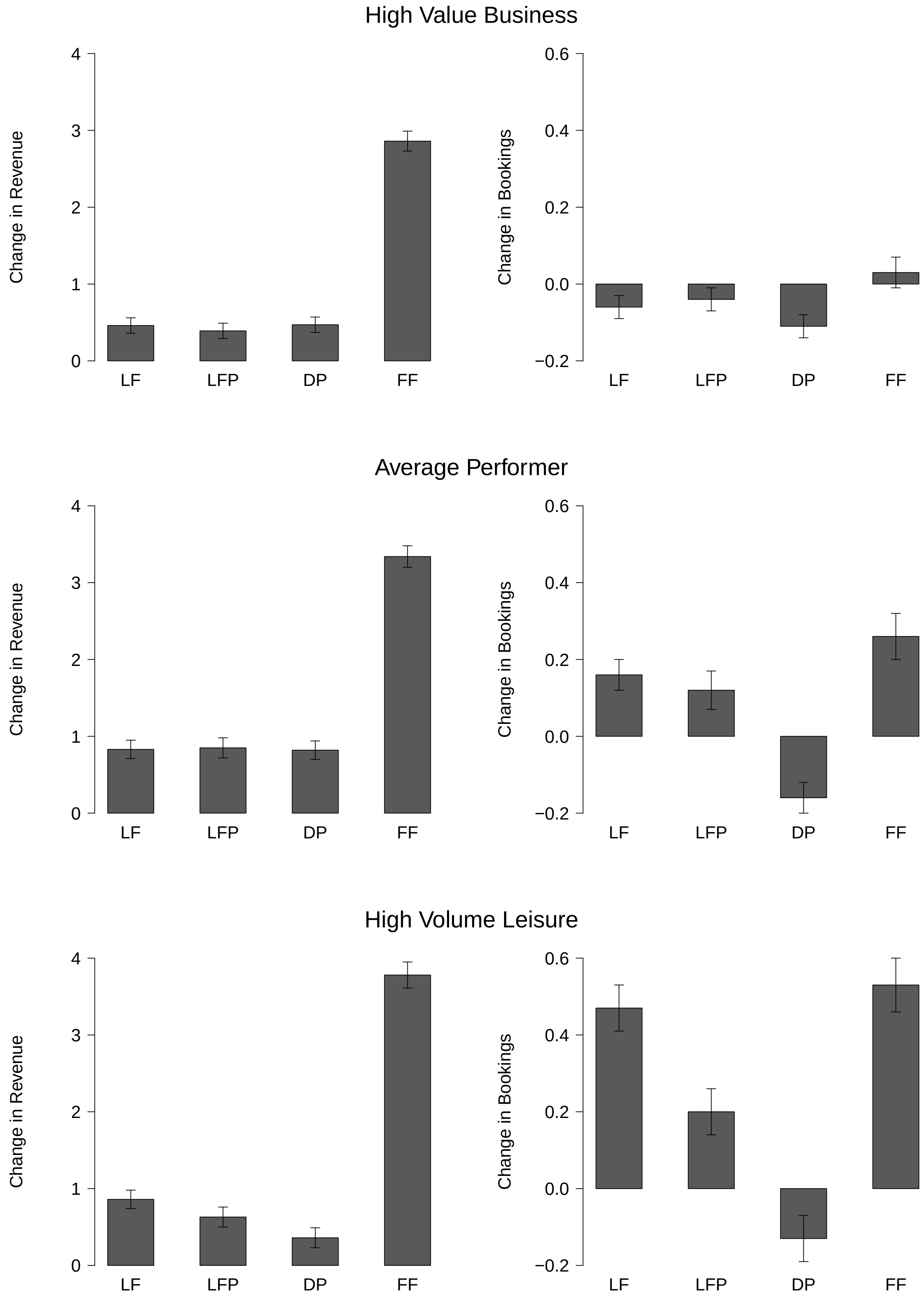

Figure 10.5.: Revenue and Bookings for the Static Valuation Schemes with BPS

10.3. Dynamic Methods

For the dynamic methods we always use BPS to control the code-share process. In view of our previous results, we refrain from testing the dynamic schemes with AVS: BPS outperforms AVS and the bid prices are used anyways to update the valuations. The results for the dynamic schemes are depicted in Figure 10.6. The lower gray line in the revenue graphs corresponds to the maximum revenue achieved by the best static scheme (FF), the upper gray line to the approximated central revenue.

First of all, the graphs show that the revenue achieved by the dynamic schemes consistently exceeds the one from the static schemes. It is between two and four percent higher. Second, NUP and BPP realize revenue gains of about seven percent in Scenarios 2 and 3, outperforming ABP by about one percent. In these scenarios, their performance reaches the level of the central solution and even slightly exceeds it. This is possible because (1) the central solution is calculated with different customer requests and is only an estimation, and (2) the central model is an heuristic itself and may not necessarily provide the real optimum. In Scenario 1, the three dynamic methods perform similarly well with gains of around five percent – 0.5 percent below the approximated central solution. Last, the revenue of the three schemes increases with the overall demand. The gains are smallest for ABP, but reach about two percent between Scenarios 1 and 3 for NUP and BPP.

The number of bookings decreases up to 1.5 percent when using NUP or BPP. For ABP, on the contrary, they slightly increase in Scenarios 1 and 2 and slightly decrease in Scenario 3. The biggest drop in bookings stems from BPP. It is the most restrictive scheme. In Scenarios 2 and 3 with a lot of demand, this is beneficial and leads to the highest revenue increases. In Scenario 1, however, there is less demand and the control strategy is over-restrictive, resulting in smaller revenue gains compared to the other dynamic schemes.

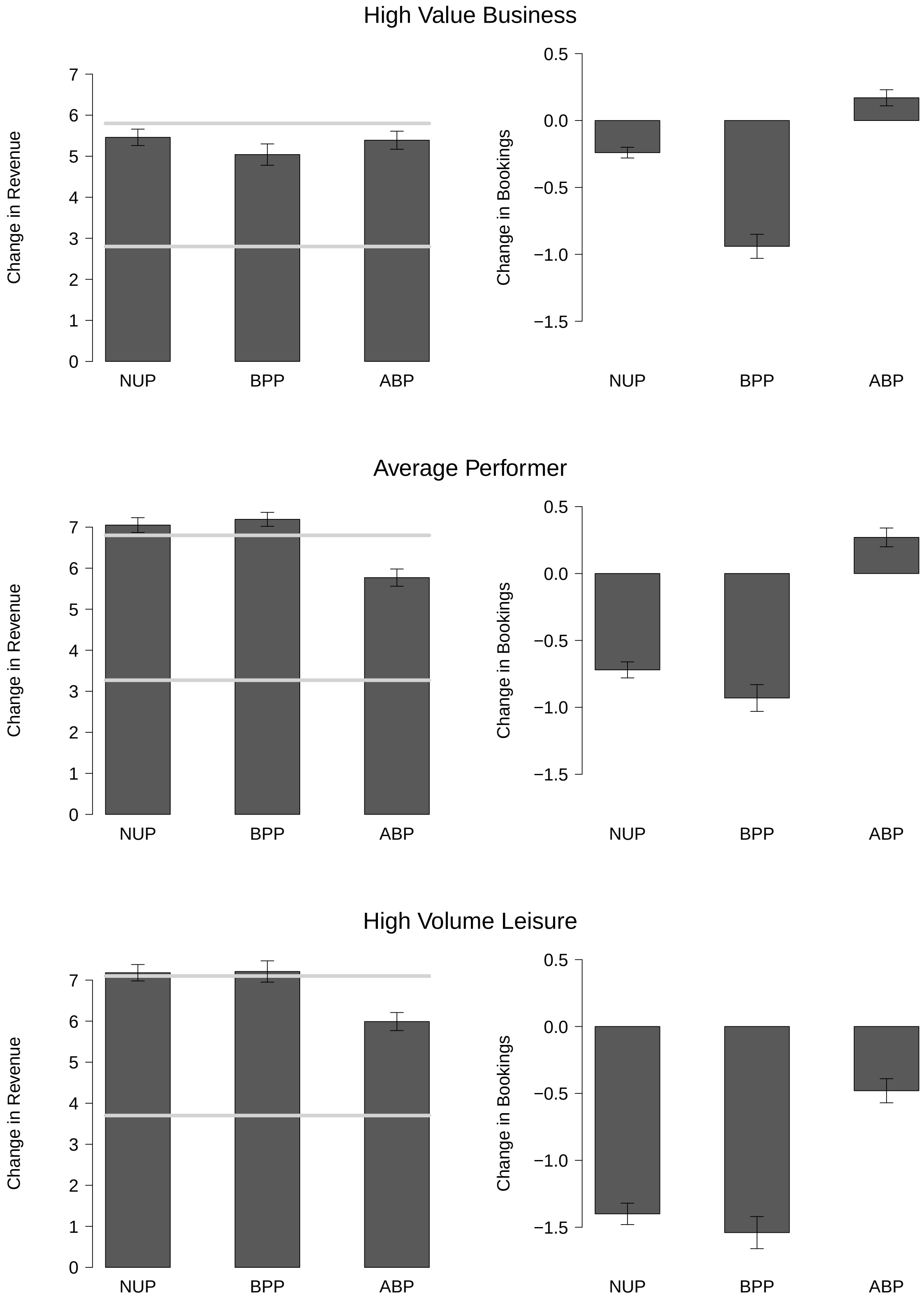

Figure 10.6.: Revenue and Bookings for the Dynamic Valuation Schemes with BPS

The drop in bookings together with the increase in revenue leads to an even stronger effect on yield, which is depicted in Figure 10.7. The greatest increases stem from BPP and amount to more than five percent in Scenario 1, nearly eight percent in Scenario 2 and slightly above eight percent in Scenario 3. These are the highest yields observed throughout our analysis and they are roughly 0.25 percent above the ones determined by NUP. The yield associated with ABP is consistently below NUP and BPP and changes less across the three scenarios. It is between five to six percent above the baseline yield.

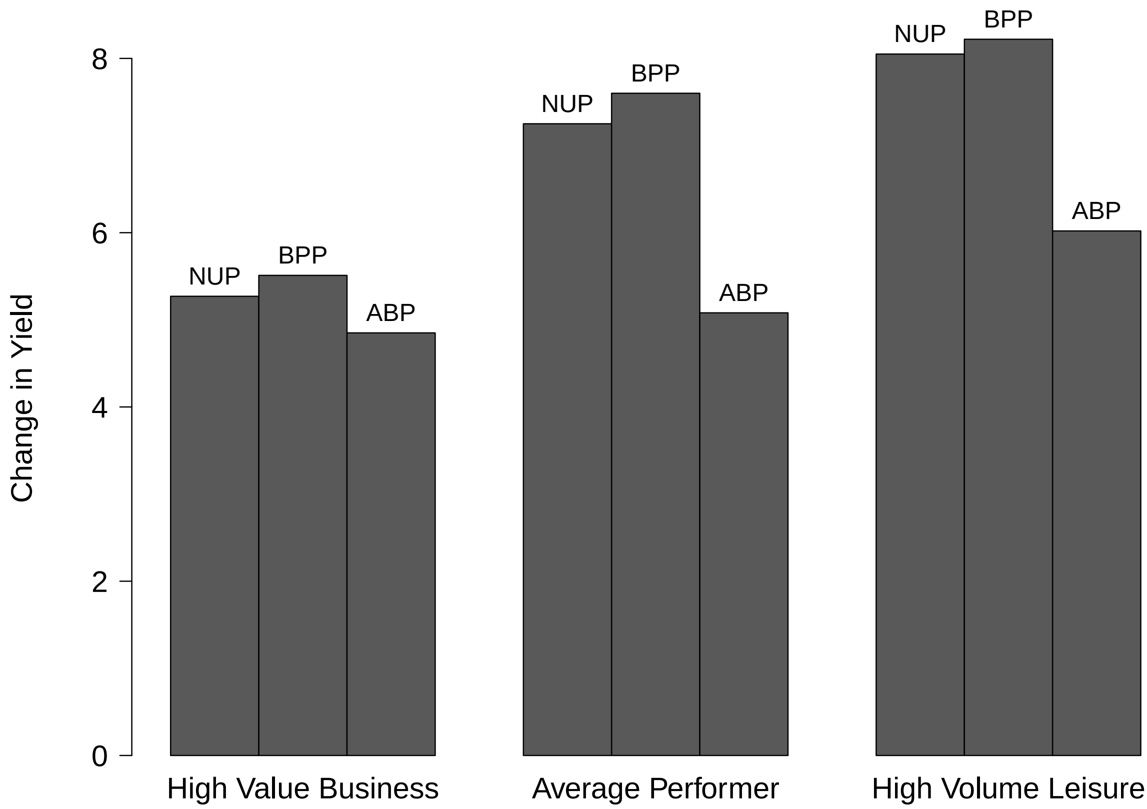

Figure 10.7.: Yield for the Dynamic Valuation Schemes with BPS

The high yield increases by NUP and BPP are attributed to the more-restrictive booking control as well as shifts in the booking mix. The latter is depicted in Figure 10.8. As the changes in the booking mix are very similar for the three dynamic methods, we only show the result for NUP.

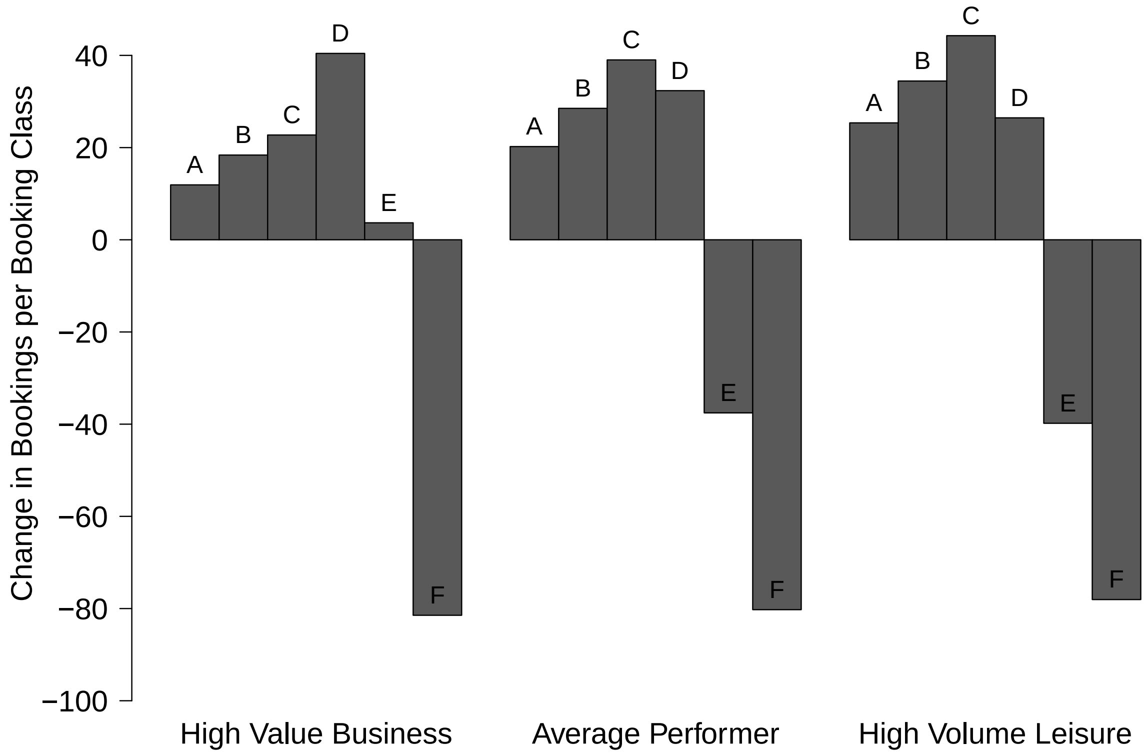

Figure 10.8.: Booking Class Mix for NUP

The dynamic valuation schemes have the greatest impact on the lowest booking class F. The bookings in this class drop by up to 80 percent. At the same time, the bookings in the four highest classes increase and occupy the capacity that is freed up. Comparing the three scenarios, the increases in the three highest classes A, B and C get bigger, the more demand there is, while the bookings in the three lower classes D, E and F decrease more and more. This indicates that the optimization process becomes the efficient with high demand. Thereby, the potential to create up-sell by forcing customers to buy a higher class increases. As a consequence, the improved code-share valuations have a stronger impact in these scenarios.

10.4. Discussion of the Results

From our results we can derive a few main results: First, BPS outperforms AVS and is more suited for network control. Second, dynamic schemes are superior to static schemes and increase the overall revenue by two to four percent in our monopoly market because they better adapt to the demand situation on each flight. Third, the performance of the schemes depends strongly on the composition of the demand. On the one hand, the higher the average value of the customers, the riskier it becomes to accept the wrong one. On the other hand, the higher the overall demand, the more customers are affected.

With respect to the adaptation scheme, we only implemented a single variant based on bid prices, which is the most realistic one from a practical perspective. Bid price are frequently used for inventory control and serve as a common currency for information sharing. Moreover, bid prices are available at every point in time and reflect the latest booking situation. The shadow prices, in contrast, are only updated during the optimization and may not incorporate recent bookings. Furthermore, the carriers need to align their entire revenue management processes to implement the ideal information exchange with iterative updates. Our results show that the simpler variant based on bid prices reaches the central outcome in two of the three scenarios and hence, provides an interesting alternative for implementation in practice. The theoretical benefits of the optimal valuations seem to vanish during the revenue management process and due to the stochastic customer demand such that the approximation performs similarly well.

Last, we note that Dynamic Valuation (ABP) also outperforms the static schemes, but is dominated by the adaptation schemes. The more accurate imitation of the central process calculates more restrictive steering parameters and increases the revenue as well as the yield in comparison to ABP.

11. Conclusion and Outlook

Our introductory discussion in Chapter 1 highlights the importance of code-sharing in today's airline industry as well as the challenges that it creates on information systems and the revenue management process. We find that the number of code-shared flights increased tremendously over the last decade and that code-share bookings provide a significant source for revenue. However, the important role of code-sharing from a business perspective is not adequately reflected in the revenue management systems. Airlines focus on their intraline networks and implement simple procedures to enable code-share sales. This leads to suboptimal booking control and jeopardizes their financial performance. Based upon these observations and the literature review in Chapters 2 and 3, we propose a theoretical framework to study the interaction of alliance members. As our goal is to provide solutions that are implementable in practice, our fundamental assumptions are decentralization and incomplete information. So, the airlines can neither treat the alliance as a single airline, nor can they share all information necessary to calculate the optimal alliance solution. Instead the partners operate independent revenue management systems and exchange minimum information.

Under these assumptions we examine the control decisions of the partners and propose an innovative as well as practical approach to govern code-share itineraries. We show under which conditions the proposed methods achieve optimal alliance revenues and thoroughly quantify their performance. The details of our findings are summarized in this chapter. Furthermore, we discuss our contribution in view of the related literature and provide directions for future research.

11.1. Summary of the Findings

The central result of this thesis is presented in Chapter 6: An adaptation algorithm that updates code-share valuations such that the local solutions of the independent airlines converge to the central solution. Using the code-share revenue management model introduced in Chapter 5, we theoretically derive necessary and sufficient conditions for central optimality as well as adaptation rules. The optimality conditions are in the form of boundaries on the code-share valuations used in the optimizer and the capacity allocations. Code-share valuations are crucial for the outcome of the optimization and link the airlines' decisions. Finding proper values is a simple way to determine the optimal alliance solution. As the carriers may not have such values directly, we introduce update rules, which change the valuations whenever they are not optimal.

The performance of the algorithm is evaluated in two ways: On the one hand, we perform theoretical experiments in a deterministic setting, solely modeling the linear programs. All other steps of the revenue management process are neglected and there is no customer interaction. On the other hand, we run stochastic simulations to verify the performance in a real-world setting with customer interaction and embedded in a full-scale revenue management system.

The theoretical experiments are presented in Chapter 7. We develop a small sample alliance network with two airlines and four flights. In three scenarios differing by the number of booking classes, we execute Monte Carlo Simulations with 10,000 randomly generated instances. Our results show that the adaptation algorithm outperforms all other commonly used valuation schemes. The two heuristics approximating the sufficient optimality conditions converge to the central solution in more than 96 percent of the instances and minimize the expected error between the local and the central shadow prices. The performance of the necessary procedure is slightly worse, providing the central optimal shadow prices in 60–70 percent of the instances. Furthermore, we notice that the more booking classes there are, the smaller is the gap between the local and central shadow prices. From this observation we conclude that in a real-world setting with possibly hundreds of price points on every flight, valuations satisfying the necessary conditions are sufficiently close to the central solution and we do not expect significant benefits from applying the sufficient procedure.

Having established these theoretical results, we examine their applicability to practice in Chapter 8 and conduct a series of stochastic simulations. We evaluate the performance of the update rules in a stochastic setting with customer interaction. For this purpose we introduce the revenue management simulation environment REMATE in Chapter 9. The results of the different code-share valuation policies are analyzed across typical demand constellations in Chapter 10. Looking at the results of the static schemes, we find that BPS outperforms AVS independently of the underlying valuation method. Moreover, a consistent overvaluation of code-share itineraries may be beneficial when their value exceeds the average value of intraline itineraries. With respect to the dynamic schemes, we implement three variants all using the stochastic bid prices. Two are based on the adaptation concept, the third corresponds to the Dynamic Valuation idea developed by Jain (2011). Our results show that (1) dynamic schemes are superior to static ones, and (2) the adaptation schemes outperform Dynamic Valuation. The revenue is one percent higher in two of the three scenarios, while they perform equally well in the third scenario. Last, we note that using bid prices instead of shadow prices provides an effective way to implement the adaptation idea. The results are close to the central optimum.

11.2. Research Contributions

This thesis contributes to the alliance revenue management literature in several ways:

- Industry data from Lufthansa shows that the majority of flights of large network carriers is affected by code-sharing and that code-share bookings account for a significant share of the total bookings. Thereby, they also have a severe impact on internal revenue management processes and the control decisions.

- Alliances cannot implement central revenue management control due to legal, technical and organizational burdens. This leads to a situation of decentralization and limited information exchange.

- Current revenue management methodologies used in the industry do not explicitly consider code-share itineraries. Airlines forecast and optimize code-share bookings together with local bookings, and the majority of alliance partners use AVS to exchange flight availabilities.

- Separation of local and code-share itineraries is necessary to implement sophisticate code-share control methods. The simplest parameter to control the revenue management decisions on shared routes is the prorated code-share revenue – the values associated with the individual segments of the itinerary.

- The prorated code-share valuations must satisfy certain necessary conditions to implement the central solution.

- To guarantee the implementation of the central optimal solution in the local linear programs, the local optimal code-share allocations must be central feasible.

- It is necessary to imitate the central procedure at all stages of the revenue management process in order to maximize the joint performance of the alliance members.

- BPS outperforms AVS for code-share control and is more robust toward the valuation methods. In addition, it minimizes the effect on bookings.

- With one exception, all valuation schemes improve the current setup applied in the industry.

- Dynamic code-share valuations outperform static ones.

- The schemes based on the adaptation algorithm perform best and provide results close to the central optimum.

Overall, our work provides a comprehensive investigation of code-share control methods. In contrast to most related contributions, we assume a strictly decentralized setting and provide a theoretical as well as practical analysis. The results that we establish, give new insights into the dynamics of code-share alliances, but also apply to similar problems outside the field of airline alliances and revenue management. Potential applications cover all kinds of situations characterized by four main conditions: (1) Multiple suppliers control distinct sets of resources with finite capacity. (2) Products use resources from different suppliers. (3) Each supplier decides upon the distribution of its resources among the products. (4) The quantity that is produced of a specific product, depends on the amount of resources provided by each supplier.

11.3. Directions for Future Research

The research presented in this thesis may be extended in several directions. The same idea of comparing and updating local optimization results can be applied to alliances using a soft-blocking environment. Instead of capacity allocations, the marginal seat revenue or the bid prices from the capacity fragments need to be compared and the allocation decisions updated whenever they do not match. The equilibrium conditions for this situation are described in Boyd (1998) and Vinod (2005). Similarly, adaptation ideas may be analyzed in the context of the monopolistic network revenue management problem. The solutions on individual flights may be compared and adapted such that leg-based optimization techniques converge to the network optimum.

Our analysis is based on the network linear program. Although it is frequently used in practice and advantageous due to its mathematical tractability, a similar approach may be studied with alternative revenue management methodologies. These could be the network dynamic program presented in Talluri and van Ryzin (1998) or the randomized linear program by Talluri and van Ryzin (1999). Moreover, additional features may be added to the model such as customer choice behavior or dependent demand.

As a big body of potential research, the practical implementation of our as well as related code-share approaches may be further investigated and their performance tested with simulations. Part III of this thesis could serve as a starting point for such an analysis and detailed evaluation criteria are outlined in Gerlach et al. (2013). There are several possible extensions to our work that could be evaluated: For example, we assume that alliance partners are completely symmetrical. If the carriers have different data structures or model assumptions, the code-share transactions need to be translated into the data structures of the partners or preprocessed to match the requirements of the different systems. The airlines may also use different revenue management setups as for example different system parameters. Thereby, the results become incomparable. Overall, studying the impact of the valuation schemes in an asymmetric setting would foster their understanding and indicate their robustness.

Another practical challenge are heterogeneous products and fares. Code-share valuations may need to be adjusted for these differences. For example, two different fare products should be valued differently than two equal products. Related to this, the impact of incorrect booking class mappings on airline revenues as well as on the long-term behavior of the revenue management process could be evaluated. In this context, it may be promising to further investigate the behavior of the valuation schemes in combination with alternative forecasting and history building methodologies as well as across different demand situations. Our simulation results indicate that the outcome strongly depends on the underlying demand situation and that it is also affected by the other steps of the revenue management process. A systematic analysis of these dependencies would be useful to decide which method should be implemented in practice. Finally, it could be interesting to test the other implementation variants of the adaptation scheme mentioned in Chapter 8. They are more challenging to realize, but promise to be more accurate.

Our analysis focuses on the operational level of the alliance game and neglects the contractual level. Analyzing revenue sharing schemes in terms of their quality as well as their impact on the performance and decisions of the individual airlines would provide another promising field for future research. Several game-theoretic ideas and a large body of business literature may help to tackle the problem and could provide new insights into the structure of revenue sharing schemes in resource exchange alliances, but also among departments within the same company.

In the context, it would be worthwhile to examine the impact of code-sharing on the individual carriers. Throughout our analysis we strictly consider the alliance result, i.e. the aggregate over all partners. We do not look at the individual carriers, as they are symmetric. Nevertheless, a similar simulation study with carriers facing asymmetric demand or having different aircraft capacities may be an interesting extension to our work. Furthermore, we do not analyze the results per route type (e.g. code-share versus intraline routes). The choice of the code-share valuation scheme has a direct effect on code-share itineraries, but the different booking behavior on these itineraries also leads to an indirect effect on the booking behavior on other itineraries, which could be evaluated.

This work investigates optimal revenue management controls for complementary code-share flights. Our discussion does not include parallel nor virtual code-shares. However, they are equally important in the industry and raise similar questions regarding their optimal control and the fair compensation of the alliance members. Both aspects may provide interesting research topics as well.

Bibliography

Abdelghany, A., Sattayalekha, W. and Abdelghany, K. (2009). On airlines code-share optimization: a modelling framework and analysis, *International Journal of Revenue Management* **3**(3): 307–330.

Adler, N. and Smilowitz, K. (2007). Hub-and-spoke network alliances and merges: Price-location competition in the airline industry, *Transportation Research Part B* **41**: 394–409.

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

Akan, M. and Ata, B. (2009). Bid-price controls for network revenue management: Martingale characterization of optimal bid prices, *Mathematics of Operations Research* **34**(11): 912–936.

Alstrup, J., Andersson, S.-E., Boas, S., Madsen, O. and Vidal, V. (1989). Booking control increases profit at scandinavian airlines, *Interfaces* **19**(4): 10–19.

Altman, E., Basar, T., Jiménez, T. and Shimkin, N. (2001). Routing into two parallel links: Game-theoretic distributed algorithms, *Journal of Parallel and Distributed Computing* **61**: 1367–1381.

Altman, E., Boulogne, T., El-Azouzi, R., Jiménez, T. and Wynter, L. (2006). A survey on networking games in telecommunications, *Computers & Operations Research* **33**(2): 286–311.

Arshinder, Kanda, A. and Deshmukh, S. G. (2008). Supply chain coordination: Perspectives, empirical studies and research directions, *International Journal of Production Economics* **115**(2): 316–335.

Balachandran, K. R. and Ronen, J. (1989). Incentive contracts when production is subcontracted, *European Journal of Operational Research* **40**: 169–185.

Bamberger, G., Carlton, D. and Neumann, L. (2004). An empirical investigation of the competitive effects of domestic airline alliances, *Journal of Law and Economics* **47**: 195–222.

Belobaba, P. (1987). *Air Travel Demand and Airline Seat Inventory Management*, PhD thesis, Massachusetts Institute of Technology.

Belobaba, P. (1989). Application of a probabilistic decision model to airline seat inventory control, *Operations Research* **37**(2): 183–197.

Bertsekas, D. P. (1999). *Nonlinear Programming: 2nd Edition*, Athena Scientific, New Hampshire.

Bertsimas, D. and de Boer, S. (2005). Simulation-based booking limits for airline revenue management, *Operations Research* **53**(1): 90–106.

Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization*, Athena Scientific, New Hampshire.

Bhandari, A. and Secomandi, N. (2011). Revenue management with bargaining, *Operations Research* **59**(2): 498–506.

Boing (1993). Decision window path preference methodology time mode, *Technical report*, Boing Commercial Airplane Group.

Bondareva, O. N. (1963). Some applications of linear programming methods to the theory of cooperative games, *Problems of Cybernetics* **10**: 119–140.

Boulogne, T., Altman, E. and Pourtallier, O. (2002). On the convergence to nash equilibrium in problems of distributed computing, *Annals of Operations Research* **109**: 279–291.

Boyd, A. (1998). Airline alliances, *OR/MS Today* **25**(5).

Boyd, A. and Bilegan, I. C. (2003). Revenue management and e-commerce, *Management Science* **49**(10): 1363–1386.

Boyd, A. and Kallesen, R. (2004). The science of revenue management when passengers purchase the lowest available fare, *Journal of Revenue and Pricing Management* **3**(2): 171–177.

Brandenburger, A. (2007). Cooperative game theory. Working Paper, New York University.

Brueckner, J. K. (2001). The economics of international codesharing: an analysis of airline alliances, *International Journal of Industrial Organization* **19**: 1475–1498.

Brueckner, J. K. (2003). International airfares in the age of alliances: the effects of codesharing and antitrust immunity, *The Review of Economics and Statistics* **85**(1): 105–118.

Brueckner, J. K. and Whalen, W. T. (2000). The price effects of international airline alliances, *Journal of Law and Economics* **43**: 503–545.

Brumelle, S. and McGill, J. I. (1993). Airline seat allocation with multiple nested fare classes, *Operations Research* **41**(1): 127–137.

Cachon, G. P. and Lariviere, M. A. (2005). Supply chain coordination with revenue-sharing contracts: Strengths and limitations, *Management Science* **51**(1): 30–44.

Cachon, G. P. and Netessine, S. (2004). *Game Theory in Supply Chain Analysis*, Handbook of Quantitative Supply Chain Analysis: Modeling in the eBusiness Era, Kluwer Academic Publishers, chapter 2.

Çetiner, D. and Kimms, A. (2012). Approximate nucleolus-based revenue sharing in airline alliances, *European Journal of Operational Research* **220**: 510–521.

Chakravorty, J. (2010). Airline alliances becoming the new 'mergers', http://www.reuters.com/article/idUSTRE61P4BP20100226. Last Accessed on July 4, 2013.

Chatterjee, K. and Lilien, G. L. (1984). Efficiency of alternative bargaining procedures, *Journal of Conflict Resolution* **28**(2): 270–295.

Chatterjee, K. and Samuelson, W. (1983). Bargaining under incomplete information, *Operations Research* **31**(5): 835–851.

Chen, H.-Y. and Wu, H.-C. (2008). Decentralized supply chain coordination with revenue sharing mechanism: Transfer pricing heuristics and revenue share rates. Working Paper, Chaoyang University of Technology.

Chua, C. L., Kew, H. and Yong, J. (2005). Airline code-share alliances and costs: Imposing concavity on translog cost function estimation, *Review of Industrial Organization* **26**(4): 461–487.

Chun, S. Y., Kleywegt, A. J. and Shapiro, A. (2011). Revenue management in resource exchange seller alliances. Working Paper, Georgia Institute of Technology.

Cléaz-Savoyen, R. L. (2005). *Airline revenue management methods for less restricted fare structures*, Master's thesis, Massachusetts Institute of Technology.

Cleophas, C. (2009). *Simulation-Based Analysis of Forecast Performance Evaluations for Airline Revenue Management*, PhD thesis, Faculty of Business Administration and Economics, University of Paderborn.

Cleophas, C. (2012). Assessing multi-agent simulations - inspiration through application, *Highlights on Practical Applications of Agents and Multi-Agent Systems* pp. 163–170.

Cooper, W. L., de Mello, T. H. and Kleywegt, A. J. (2006). Models of the spiral-down effect in revenue management, *Operations Research* **54**(5): 968–987.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C. (1990). *Introduction to Algorithms*, The MIT Press.

Cross, R. G. (1997). *Revenue Management*, Broadway Books, New York.

Currie, C. S. M., Cheng, R. C. H. and Smith, H. K. (2008). Dynamic pricing of airline tickets with competition, *Journal of the Operational Research Society* **59**(8): 1026–1037.

Curry, R. E. (1990). Optimal airline seat allocation with fare classes nested by origins and destinations, *Transportation Science* **24**(3): 193–204.

Dantzig, G. B. (1963). *Linear Programming and Extensions*, Princeton University Press, New Jersey.

Dantzig, G. B. and Wolfe, P. (1960). Decomposition principle for linear programs, *Operations Research* **8**(1): 101–111.

Darot, J. F. J. (2001). *Revenue management for airline alliances: Passenger origin-destination simulation analysis*, Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

de la Torre, P. E. F. (1999). *Airline alliances: The airline perspective*, Master's thesis, Massachusetts Institute of Technology.

Economides, N. and Salop, S. C. (1992). Competition and integration among complements and network market structure, *The Journal of Industrial Economics* **40**(1): 105–123.

Feldman, J. M. (1998). Making alliances work, *Air Transport World* **35**(6): 26–31.

Fiig, T., Isler, K., Hopperstad, C. and Belobaba, P. (2010). Optimization of mixed fare structures: Theory and applications, *Journal of Revenue and Pricing Management* **9**(1): 152–170.

Frank, M., Cleophas, C., Schröder, T. and Gerlach, M. (2010). Introduction of REMATE, Presentation at the Annual AGIFORS Reservation and Yield Management Study Group Meeting.

Frank, M., Friedemann, M. and Schroeder, A. (2008). Principles for simulations in revenue management, *Journal of Revenue and Pricing Management* **7**(1): 7–16.

Gagnepain, P. and Marín, P. L. (2010). The effects of airline alliances: What do the aggregate data say?, *Journal of the Spanish Economic Association* **1**(3): 251–276.

Gallego, G. and Hu, M. (2009). Dynamic pricing of perishable assets under competition. Working Paper, Columbia University.

Gallego, G. and van Ryzin, G. J. (1994). Optimal dynamic pricing of inventories with stochastic demand over finite horizons, *Management Science* **40**(8): 999–1020.

Gallego, G. and van Ryzin, G. J. (1997). A multiproduct dynamic pricing problem and its applications to network yield management, *Operations Research* **45**(1): 24–41.

Gayle, P. G. (2007). Is virtual codesharing a market segmenting mechanism employed by airlines?, *Economics Letters* **95**(1): 17–24.

Gerlach, M., Cleophas, C. and Kliewer, N. (2013). Airline codeshare alliances – marketing boon and revenue management information systems challenge, *Business & Information Systems Engineering* **5**(3): 153–163.

Giannoccaro, I. and Pontrandolfo, P. (2004). Supply chain coordination by revenue sharing contracts, *International Journal of Production Economics* **89**: 131–139.

Glover, F., Glover, R., Lorenzo, J. and McMillan, C. (1982). The passenger-mix problem in the scheduled airlines, *Interfaces* **12**(3): 73–80.

Goerzen, A. (2005). Managing alliance networks: Emerging practices of multinational corporations, *Academy of Management Executive* **19**(2): 94–107.

Goh, K. and Uncles, M. (2003). The benefits of airline global alliances: an empirical assessment of the perceptions of business travelers, *Transportation Research Part A* **37**(6): 479–497.

Goldman, P., Freling, R., Pak, K. and Piersma, N. (2001). Models and techniques for hotel revenue management using a rolling horizon. Econometric Institute Report EI 2001-46, Erasmus University of Rotterdam.

Gorin, T. and Belobaba, P. (2004). Impacts of entry in airline markets: Effects of revenue management on traditional measures of airline performance, *Journal of Air Transport Management* **10**: 259–270.

Graf, M. and Kimms, A. (2001). An option-based revenue management procedure for strategic airline alliances, *European Journal of Operational Research* **215**: 459–469.

Greenberg, H. J. (1986). An analysis of degeneracy, *Naval Research Logistics Quarterly* **33**: 635–655.

Groves, T. and Loeb, M. (1979). Incentives in a divisionalized firm, *Management Science* **25**(3): 221–230.

Gulati, R. (1998). Alliances and networks, *Strategic Management Journal* **19**: 293–317.

Ho, Y.-C., Luh, P. B. and Olsder, G. J. (1982). A control-theoretic view on incentives, *Automatica* **18**(2): 167–179.

Hopperstad, C. (2000). Passenger origin destination simulator technical specification. Working Paper, Seattle.

Houghtalen, L., Ergun, Ö. and Sokol, J. (2011). Designing mechanisms for the management of carrier alliances, *Transportation Science* **45**(4): 465–482.

Hu, X., Caldentey, R. and Vulcano, G. (2013). Revenue sharing in airline alliances, *Management Science* **59**(5): 1177–1195.

Ireland, R. D., Hitt, M. A. and Vaidyanath, D. (2002). Alliance management as a source of competitive advantage, *Journal of Management* **28**(3): 413–446.

Isler, K. and Imhof, H. (2008). A game theoretic model for airline revenue management and competitive pricing, *Journal of Revenue and Pricing Management* **7**(4): 384–396.

Ito, H. and Lee, D. (2007). Domestic codesharing, alliances and airfares in the U.S. airline industry, *The Journal of Law and Economics* **50**: 355–380.

Jain, H. (2011). *Alliance revenue management in practice: Techniques and simulation analysis*, Master's thesis, Massachusetts Institute of Technology.

Kadatz, D. M. (2011). *Revenue sharing in airline alliances: A game-theoretic approach*, Master's thesis, Freie Universität Berlin.

Kasilingam, R. G. (1996). Air cargo revenue management: Characteristics and complexities, *European Journal of Operational Research* **96**: 36–44.

Killing, J. P. (1988). *Understanding Alliances: The Role of Task and Organizational Complexity*, Cooperative Strategies and International Business, Lexington Books, Lexington, MA, chapter 3, pp. 55–67.

Kim, E. H. and Singal, V. (1993). Mergers and market power: Evidence from the airline industry, *The American Economic Review* **83**(3): 549–569.

Kim, S. H. (1991). Coordination of multiagent systems through explicit valuation of action, *Robotics and Computer-Integrated Manufacturing* **8**(4): 265–291.

Kleymann, B. and Seristö, H. (2001). Levels of airline alliance membership: Balancing risks and benefits, *Journal of Air Transport Management* **7**: 303–310.

Kunnumkal, S. and Topaloglu, H. (2009). A new dynamic programming decomposition method for the network rm problem with customer choice behavior. Working Paper, Indian School of Business, Hyderabad.

LaRoche, M.-A., Gamache, M. and Olivier-Ouellet, J. (2010). Dynamic optimization of codeshare flight selection for an airline company. Working Paper, École Polytechnique de Montréal.

Lee, A. O. (1990). *Airline reservations forecasting: probabilistic and statistical models of the booking process*, PhD thesis, Massachusetts Institute of Technology.

Lee, A. Y. H. (1998). *Investigation of competitive impacts of origin-destination control using pods*, Master's thesis, Massachusetts Institute of Technology.

Lee, T. C. and Hersh, M. (1993). A model for dynamic airline seat inventory control with multiple seat booking, *Transportation Science* **27**(3): 252–265.

Littlewood, K. (1972). Forecasting and control of passenger bookings, *In the Proceedings of the Annual AGIFORS Reservation and Yield Management Study Group Meeting*, Nathanya (Israel).

Luo, L. and Peng, J. (2007). Dynamic pricing model for airline revenue management under competition, *Systems Engineering - Theory & Practice* **27**(11): 15–25.

Maglaras, C. and Meissner, J. (2006). Dynamic pricing strategies for multiproduct revenue management problems, *Manufacturing & Service Operations Management* **8**(2): 136–148.

Maruna, M. and Morrell, P. (2010). Mergers: After the honeymoon, http://www.flightglobal.com/articles/2010/07/29/345465/mergers-after-the-honeymoon.html. Last Accessed on October 15, 2012.

McAfee, R. P. and te Velde, V. (2006). *Dynamic Pricing in the Airline Industry*, Elseviers Science Publishers.

McGill, J. I. and van Ryzin, G. J. (1999). Revenue management: Research overview and prospects, *Transportation Science* **33**(2): 233–256.

Meissner, J. and Strauss, A. K. (2011). Network revenue management with inventory-sensitive bid prices and customer choice. Working Paper, Lancaster University.

Michaels, L. (2007). Significant gains through an integrated view by origin-destination market, *Journal of Revenue and Pricing Management* **6**(4): 274–278.

Morrish, S. and Hamilton, R. (2002). Airline alliances - who benefits?, *Journal of Transport Management* **8**: 401–407.

Nagarajan, M. and Bassok, Y. (2008). A bargaining framework in supply chains: The assembly problem, *Management Science* **54**(8): 1482–1496.

Nash, J. F. (1950). The bargaining problem, *Econometrica* **18**(2): 155–162.

Netessine, S. and Shumsky, R. A. (2005). Revenue management games: Horizontal and vertical competition, *Management Science* **51**(5): 813–831.

Nicolaou, A. I. and Christ, M. H. (2011). Integrated information systems, risk sharing and alliance risk. Working Paper, Bowling Green State University.

Oberstar, J. L. (2008). Airline consolidation? hell no. Business Week, February 15.

O'Neal, J. W., Jacob, M. S., Farmer, A. K. and Martin, K. G. (2007). Development of a codeshare flight-profitability system at delta air lines, *Interfaces* **37**(5): 436–444.

Orda, A., Rom, R. and Shimkin, N. (1993). Competitive routing in multi-user communication networks, *IEEE/ACM Transactions on Networking* **1**(5): 510–521.

Osborn, R. N. and Hagedoorn, J. (1997). The institutionalization and evolutionary dynamics of interorganizational alliances and networks, *The Academy of Management Journal* **40**(2): 261–278.

Oum, T. H., Park, J.-H. and Zhang, A. (1996). The effects of airline codesharing agreements on firm conduct and international air fares, *Journal of Transport Economics and Policy* **30**(2): 187–202.

Pak, K. and Piersma, N. (2002). Airline revenue management: An overview of OR techniques 1982-2001. Econometric Institute Report EI 2002-03, Erasmus University of Rotterdam.

Park, J.-H. (1997). The effects of airline alliances on markets and economic welfare, *Transportation Research Part E* **33**(3): 181–195.

Pölt, S. (1998). Forecasting is difficult - especially if it refers to the future, *In the Proceedings of the Annual AGIFORS Reservation and Yield Management Study Group Meeting*, Melbourne (Australia).

Queenan, C. C., Ferguson, M., Higbie, J. and Kapoor, R. (2007). A comparison of unconstraining methods to improve revenue management systems, *Productions and Operations Management* **16**(6): 729–746.

Randall, P. (2010). How will continental and united merge their or methodologies?, http://reflectionsonor.wordpress.com/2010/10/21/how-will-continental-and-united-merge-their-or-methodologies/. Last Accessed on October 15, 2012.

Reyes, M. H. (2006). *Hybrid forecasting for airline revenue management in semi-restricted fare structures*, Master's thesis, Massachusetts Institute of Technology.

Ronen, J. and Balachandran, K. R. (1988). An approach to transfer pricing under uncertainty, *Journal of Accounting Research* **26**(2): 300–314.

Rothstein, M. (1971). An airline overbooking model, *Management Science* **5**(2): 180–192.

Saharidis, G. K., Dallery, Y. and Karaesmen, F. (2006). Centralized versus decentralized production planning, *RAIRO Operations Research* **40**: 113–128.

Shapley, L. (1967). On balanced sets and cores, *Naval Research Logistics Quarterly* **14**: 435–460.

Shubik, M. (1962). Incentives, decentralized control, the assignment of joint costs and internal pricing, *Management Science* **8**(3): 325–343.

Shumsky, R. A. (2006). The southwest effect, airline alliances and revenue management, *Journal of Revenue and Pricing Management* **5**(1): 83–89.

Simpson, R. W. (1989). Using network flow techniques to find shadow prices for market and seat inventory control. Memorandum M89-1, MIT Flight Transportation Laboratory.

Sivakumar, R. (2003). Codeshare optimizer - maximizing codeshare revenues, *In the Proceedings of the Annual AGIFORS Reservation and Yield Management Study Group Meeting*, Paris (France).

Skwarek, D. K. (1997). *Competitive impacts of yield management system components: Forecasting and sell-up models*, Master's thesis, Massachusetts Institute of Technology.

Smith, B. C., Leimkuhler, J. F. and Darrow, R. M. (1992). Yield management at american airlines, *Interfaces* **22**(1): 8–31.

Smith, B. C. and Penn, C. (1988). Analysis of alternative origin-destination control strategies, *In the Proceedings of the Annual AGIFORS Reservation and Yield Management Study Group Meeting*, pp. 123–144.

Srinivasan, A. (2004). Optimizing codeshare revenues, *In the Proceedings of the Annual AGIFORS Reservation and Yield Management Study Group Meeting*, Singapore.

Suzuki, Y. (2006). The net benefit of airline overbooking, *Transportation Research Part E* **42**(1): 1–19.

Talluri, K. T. and van Ryzin, G. J. (1998). An analysis of bid-price controls for network revenue management, *Management Science* **44**(11): 1577–1593.

Talluri, K. T. and van Ryzin, G. J. (1999). A randomized linear programming method for computing network bid prices, *Transportation Science* **33**(2): 207–216.

Talluri, K. T. and van Ryzin, G. J. (2004). *The Theory and Practice of Revenue Management*, Kluwer Academic Publishers.

Tomkins, C. (2001). Interdependencies, trust and information in relationships, alliances and networks, *Accounting, Organizations and Society* **26**: 161–191.

Topaloglu, H. (2009). Using lagrangian relaxation to compute capacity-dependent bid prices in network revenue management, *Operations Research* **57**(3): 637–649.

Topaloglu, H. (2012). A duality based approach for network revenue management in airline alliances, *Journal of Revenue and Pricing Management* **11**(5): 500–517.

Tretheway, M. W. (2004). Distortions of airline revenues: Why the network airline business model is broken, *Journal of Air Transport Management* **10**(1): 3–14.

van Ryzin, G. J. (2005). Models of demand, *Journal of Revenue and Pricing Management* **4**(2): 204–210.

van Ryzin, G. J. and Vulcano, G. (2008). Simulation-based optimization of virtual nesting controls for network revenue management, *Operations Research* **56**(4): 865–880.

Vickrey, W. (1972). Airline overbooking: Some further solutions, *Journal of Transport Economics and Policy* **6**(3): 257–270.

Vinod, B. (2005). Alliance revenue management, *Journal of Revenue and Pricing Management* **4**(1): 66–82.

Vinod, B. (2010). The complexities and challenges of the airline fare management process and alignment with revenue management, *Journal of Revenue and Pricing Management* **9**: 137–151.

Waslander, S. L., Inalhan, G. and Tomlin, C. J. (2003). Decentralized optimization via nash bargaining, *Conference on Cooperative Control and Optimization*.

Weatherford, L. and Belobaba, P. (2002). Revenue impacts of fare input and demand forecast accuracy in airline yield management, *Journal of the Operational Research Society* **53**(8): 811–821.

Weatherford, L. and Pölt, S. (2002). Better unconstraining of airline demand data in revenue management systems for improved forecast accuracy and greater revenues, *Journal of Revenue and Pricing Management* **1**(3): 234–254.

Williamson, E. L. (1992). *Airline Network Seat Inventory Control: Methodologies and Revenue Impacts*, PhD thesis, Massachusetts Institute of Technology.

Wollmer, R. D. (1992). An airline seat management model for a single leg route when lower fare classes book first, *Operations Research* **40**(1): 26–37.

Wright, C. P. (2010). *Modeling and Analysis of Revenue Management in Airline Alliances*, PhD thesis, Simon Graduate School of Business Administration, University of Rochester.

Wright, C. P., Groenevelt, H. and Shumsky, R. A. (2010). Dynamic revenue management in airline alliances, *Transportation Science* **44**(1): 15–37.

Zeni, R. H. (2001). *Improved Forecast Accuracy in Revenue Management by Unconstraining Demand Estimates from Censored Data*, PhD thesis, Rutgers Graduate School, University of New Jersey.

Zhang, A. and Zhang, Y. (2006). Rivalry between strategic alliances, *International Journal of Industrial Organization* **24**(2): 287–301.

Zhang, D. (2011). An improved dynamic programming decomposition approach for network revenue management, *Manufacturing & Service Operations Management* **13**(1): 35–52.

Zhang, D. and Cooper, W. L. (2009). Pricing substitutable flights in airline revenue management, *European Journal of Operational Research* **197**: 848–861.

Zhang, D. and Kallesen, R. (2008). Incorporating competitive price information into revenue management, *Journal of Revenue and Pricing Management* **7**(1): 17–26.

Zhang, Z. and Douligeris, C. (1992). Convergence of synchronous and asynchronous greedy algorithms in a multiclass telecommunications environment, *IEEE Transactions on Communications* **40**(8): 1277–1281.

Zickus, J. S. (1998). *Forecasting for airline network revenue management: Revenue and competitive impacts*, Master's thesis, Massachusetts Institute of Technology.

Zimmermann, B., Cleophas, C. and Frank, M. (2011). Universeller simulator für revenue management, *Internationales Verkehrswesen* **63**(3): 76–84.

“Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen und Hilfsmitteln wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.”

Ort, Datum

Unterschrift