The Airline Pricing Problem

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

von Diplom-Mathematiker Jonas Rauch
geb. am 11. Januar 1983 in Heidelberg

genehmigt durch die Fakultät für Mathematik
der Otto–von–Guericke–Universität Magdeburg

Gutachter: Prof. Dr. Sebastian Sager
Dr. Michael Frank

eingereicht am: 5. September 2017
Verteidigung am: 10. November 2017

DOI: 10.24352/UB.OVGU-2018-001

Abstract

Airline revenue management can be separated into two major areas — pricing and inventory control — with the joint objective function of maximizing revenue. Pricing defines optimal fare products, each being a combination of a price, segmentation rules and attributes such as rebooking conditions, and assigns each fare product to a booking class. Inventory control then optimally controls availability of each booking class as a function of expected demand and remaining inventory. Both have a strong impact on an airline's profit and therefore play a critical role in its success in a highly competitive market.

While inventory control has been the subject of extensive research in the past decade, the pricing side has gotten very little attention in the scientific RM literature. In fact, many authors use the term revenue management and inventory control synonymously. As a result, whereas capacity control is highly automated based on sophisticated forecasting and optimization methods, pricing decisions in industry practice are mostly taken manually with little decision support. One explanation for the underrepresentation of pricing in the RM literature is its complexity: It cannot be analyzed in isolation, but always has to be considered in combination with inventory control, because every pricing decision potentially changes the optimal booking class availabilities.

In this thesis we formulate the joint airline pricing and inventory control problem as a two-level optimization problem. Existing publications on this topic focus on analyzing structural properties of the problem under very limiting assumptions regarding the customer choice model and often using deterministic inventory control schemes. In contrast, in this thesis we analyze pricing for a general class of stochastic customer choice models and in combination with dynamic inventory control, with the goal of numerically solving the resulting pricing optimization problem.

To this end we conduct a sensitivity analysis of the customer choice model, allowing to numerically compute the gradient of booking probabilities with respect to prices and product attributes. In addition we derive an adjoint equation of the inventory control dynamic program for a single flight, which allows to efficiently compute the gradient of expected revenue with respect to the pricing and demand input parameters. Combining both we are then able to apply gradient-based optimization methods to solve the joint pricing and inventory control problem.

Transferring concepts form network inventory control, the pricing methodology is heuristically extended to the network case with a large number of flights connected by transfer traffic. As a by-product an improved optimization and control mechanism for network inventory control is derived, which in a simulation study shows significant revenue gains over the traditional method.

Zusammenfassung

Airline Revenue Management besteht aus zwei wesentlichen Teilen — Pricing und Kapazitätssteuerung — mit dem gemeinsamen Ziel, den Gesamtumsatz zu maximieren. Pricing definiert sogenannte Tarifprodukte, jeweils bestehend aus einem Preis, Segmentierungsregeln und Produkteigenschaften wie Umbuchungskonditionen, und ordnet jedes Produkt einer Buchungsklasse zu. Im Rahmen der Kapazitätssteuerung wird dann die Verfügbarkeit aller Buchungsklassen in Abhängigkeit von erwarteter zukünftiger Nachfrage und Restkapazität optimal gesteuert. Beides hat einen starken Einfluss auf den Profit einer Airline und damit auf ihren Erfolg in einem zunehmend kompetitiven Wettbewerbsumfeld.

Während die Kapazitätssteuerung über die vergangenen Jahrzehnte hinweg ausführlich studiert wurde, hat Pricing in der wissenschaftlichen Revenue Management–Literatur nur sehr geringe Aufmerksamkeit genossen. Viele Autoren verwenden den Begriff Revenue Management synonym mit Kapazitätssteuerung. Aus diesem Grund werden Pricingentscheidungen in der Industriepraxis häufig manuell und wenig Entscheidungsunterstützung getroffen, wohingegen die Kapazitätssteuerung auf Basis fortschrittlicher Prognose- und Optimierungsverfahren weitestgehend automatisiert ist. Eine Erklärung für die Unterrepräsentation von Pricing in der RM-Literatur ist dessen Komplexität: Pricing kann nicht isoliert analysiert werden, sondern muss immer in Kombination mit Kapazitätssteuerung betrachtet werden, da jede Preisentscheidung potentiell die optimale Buchungsklassenverfügbarkeit verändert.

In dieser Arbeit wird die Kombination von Pricing und Kapazitätssteuerung als zweistufiges Optimierungsproblem formuliert. Existierende Publikationen zu diesem Thema beschäftigen sich mit der struturellen Analyse des Problems unter stark einschränkenden Annahmen bezüglich des Kundenwahlmodells und verwenden deterministische Methoden für die Kapazitätssteuerung. Im Gegensatz dazu ist Gegenstand dieser Arbeit eine Analyse des Pricingproblems für eine allgemeine Klasse stochastischer Kundenwahlmodelle und unter Verwendung von dynamischer Kapazitätssteuerung mit dem Ziel, das resultierende Optimierungsproblem numerisch zu lösen.

Zu diesem Zweck wird eine Sensitivitätsanalyse des Kundenwahlmodells durchgeführt, die es ermöglicht, numerisch die Ableitungen von Buchungswahrscheinlichkeiten nach Preisen und Produkteigenschaften zu berechnen. Weiterhin wird die adjungierte Gleichung des dynamischen Kapazitässteuerungsproblems für Einzelflüge hergeleitet, die es erlaubt auf effiziente Art und Weise den Gradienten des erwarteteten Gesamtumsatzes nach Preis- und Nachfrageparametern zu berechnen. Durch Kombination von beidem können gradientenbasierte Optimierungsverfahren für die numerische Lösung des simultanen Pricing- und Kapazitätssteuerungsproblems angewendet werden.

Durch Übertragung und Weiterentwicklung von Konzepten aus der Kapazitätssteuerung für Netzwerke wird die Methode zur Pricingoptimierung heuristisch auf den Netzwerkfall mit einer Vielzahl von Flügen, verbunden durch Umsteigeverkehr, erweitert. Als Nebenprodukt wird ein verbesserter Optimierungs- und Steuerungsmechanismus für die Kapazitätssteuerung im Netzwerk hergeleitet, der in Simulationen zu signifikanten Ertragssteigerungen gegenüber traditionellen Methoden führt.

Danksagung

Ich bedanke mich herzlich bei allen Familienangehörigen, Freunden und Kollegen, die mich in den letzten Jahren unterstützt und immer wieder motiviert haben.

An erster Stelle gilt mein Dank Sebastian Sager, nicht nur für die fachliche Untersttzung und Betreuung dieser Arbeit und dafür, dass ich Mitglied seiner Arbeitsgruppe sein durfte, sondern auch für sein anhaltendes Vertrauen über diesen langen Zeitraum.

Weiterhin möchte ich mich bei Michael Frank dafür bedanken, dass er mich mit dem Gebiet des Revenue Managements in Kontakt gebracht und mir als externer Mitarbeiter bei Lufthansa frühzeitig spannende Aufgaben anvertraut hat, und für die Bereitschaft, als Gutachter für diese Arbeit zu fungieren. Ich bedanke mich bei der Deutschen Lufthansa für die Möglichkeit, im Rahmen einer Kooperation mit der Universität Heidelberg gleichzeitig forschen und wertvolle Praxiserfahrung sammeln zu können.

Mein Fachwissen im Bereich Revenue Management ist zu großen Teilen den vielen fruchtbaren Diskussionen mit meinen Kollegen bei der Lufthansa und SWISS zu verdanken. Hervorheben möchte ich Karl Isler und Stefan Pölt, die jederzeit bereit waren, ihr breites Wissen und ihre langjährige Erfahrung zu teilen, und Benedikt Zimmermann, dessen ausführliches Feedback sehr zur Qualität dieser Arbeit beigetragen hat. Darüber hinaus bedanke ich mich bei Christopher Alder, Thomas Fiig, Martin Friedemann, Christian Popp, Eric Ruhlin, Tobias Schröder und den vielen andern Mitgliedern der deutschen und internationalen Revenue Management Community für die zahlreichen konstruktiven Gespräche.

Danke an meine Eltern und meinen Bruder Sebastian, die mich immer wieder mit ermutigenden Worten auf Kurs gehalten haben. Ganz besonders bedanke ich mich bei Lucy für ihre schier unendliche Geduld in den vergangenen Jahren und dafür, dass sie mir die vielen Wochenenden und Abende am Schreibtisch nicht übel genommen hat, sondern mich im Gegenteil fortwährend bestärkt und unterstützt hat.

Contents

Introduction

Due to the nature of current distribution systems, airlines control their offer through two separate steps. First, they publish fares that describe the product the airline is selling and its price. Each fare is a combination of a travel itinerary, a price, additional services (e.g. baggage allowance, lounge access, etc.) and fare conditions (e.g. advance purchase restrictions) that control when and by whom the fare can be purchased. These fares, and in particular the prices attached to them, often remain constant for long periods of time. In order to have more control over the selling process, each fare is mapped onto one of a finite number of booking classes. Airlines then control availability of these booking classes through inventory control mechanisms. Booking class availability is controlled dynamically over the course of the booking horizon depending on capacity, a demand forecast, and the observed number of bookings so far, and therefore changes frequently, sometimes multiple times per day. In order to travel on a certain itinerary, the customer needs to purchase a fare for the respective travel path, i.e. pay the price of the fare and adhere to the fare conditions. In addition, booking has to be made for the itinerary in the corresponding booking class, which is only possible if the booking class needs is made available for booking by the airline. These two means that the airline can use to control their offer were historically introduced with two different goals in mind.

Availability control for booking classes was originally developed as a tool to optimally control limited capacity over the course of the booking horizon, and is motivated by the following two facts. Firstly, different customers pay different amounts for the same seat, and if capacity is too scarce to accommodate all potential customers, the airline would of course prefer to transport only the highest paying customers. Secondly, any inventory that has not been sold at the time of departure is spoiled, and the revenue of potential customers that were turned away earlier during the booking horizon is lost. In very simple terms, capacity control therefore has the goal to reserve a sufficient number of seats for late booking customers, who generally pay higher prices, while not sacrificing revenue by turning away too much demand early during the booking horizon and leaving empty seats. This *capacity control* or *inventory control* problem is well-studied in the scientific literature and is often called the Revenue Management (RM) or yield management problem. Advanced demand forecasting and optimization methods and corresponding inventory control schemes are widely applied in industry practice. There are many different formulations and solution algorithms for the capacity control problem, depending on network structure (single resource vs. network problem) and assumptions about customer demand (deterministic or stochastic demand). This work highly relies on existing theory and solution algorithms from the classic RM literature. The most important results are reviewed in detail in Chapter 3.

Complex fare structures with multiple price points and fare restrictions are a tool for price discrimination. They exploit the fact that different customer segments, which differ in their willingness to pay, have different travel patterns and preferences. For example, business travelers often have a higher willingness-to-pay than tourist. At the same time, tourists tend to stay longer at their destination that business travelers. This allows the airline to offer a cheap fare specifically to tourists by imposing a minimum stay condition, which states that the fare can only be bought if the time between the outbound and inbound flights is longer than a certain number of days, which prevents business travelers to use the discounted fare. In addition, even in the absence of fare restrictions the airline can benefit from having multiple price points as basis for capacity control: In the extreme case of only a single price point, there would be no capacity control. All customers pay the same, and the airline would simply accept bookings in a first-come-first-serve manner. On the other hand, if the airline has a large number of price points to choose from when making the availability decision it can exercise finer control over the booking process and react dynamically to variation in customer demand, closing booking classes when demand turns out to be higher than expected and vice versa. We call the problem of optimally defining fare structures the *pricing problem*. In practice, this is done manually by pricing analysts, who rely on their knowledge of a certain market, and particularly about the mix of different customer segments and the criteria by which they can be distinguished. In this thesis, we analyze the problem of choosing optimal fares (conditions and price points) from a mathematical optimization perspective and present numerical solution methods.

Clearly, expected revenue for a given set of fares greatly depends on how availability is controlled during the booking horizon. In other words, evaluating the objective function of the pricing problem, i.e. computing expected revenue that can be achieved given a certain set of fares, requires the solution of the corresponding RM problem. The pricing problem can therefore be seen as a two-level problem, where the outer problem is the one of choosing optimal fares, while the inner problem is the capacity control problem.

In the classic RM literature, it is usually assumed that prices and fare conditions are fixed. Furthermore, the demand forecasting models used for inventory control are built around the same assumption. They are based on booking classes and use historical observations of bookings and availability to predict future demand, assuming that fares and restrictions will remain unchanged. They generally do not explicitly model customer behavior in terms of a customer choice model, and can therefore not predict how demand changes depending on a change in prices or product characteristics. This means that for the purpose of the pricing problem, where one of the main questions is how demand reacts to varying prices and conditions, we cannot use any of the booking class based demand models that are standard in the RM literature. Instead, we have to apply more general customer choice models that model the customers' response to a change in price and product attributes. Chapter 2 contains an overview over existing results regarding customer choice modeling and introduces the notation and the model we use in the later chapters of this thesis.

In this thesis, we consider both pricing and capacity control as part of RM. The full RM problem, can be seen as a stochastic optimal control problem, where we are not only optimizing control functions that vary over the course of the booking horizon (dynamic capacity control), but also parameters of the system (prices and conditions), which remain constant over time. The theoretical underpinnings of our solution approach are based on standard results from the areas of optimization and optimal control. In Chapter 1 we therefore summarize important theoretical results from these areas as well as some related numerical methods, which we later use to practically solve the pricing problem computationally.

In the second part of the thesis, we formulate and analyze the airline pricing problem in detail. We combine results from all areas listed above in order to obtain theoretical insights into the structure of the problem, based on which we then present a computationally tractable numerical solution method for the pricing problem. In Chapter 4 we describe the problem, introduce the notation and describe our approach to modeling demand as well as the methods we use to solve the underlying inventory control problem. In order to adequately model customer demand and its dependence on prices and product attributes, we use a very general customer choice model that is closely related to the mixed logit choice model. In this model, every potential customer is described as a vector of personal preferences, which follow a given joint probability distribution. Each customer chooses from a given set of alternatives by maximizing their utility, which depends on both the personal preferences as well as the attributes of the alternatives. Furthermore, we assume that customer arrival during the booking horizon follows a Poisson arrival process. Consequently, we use a dynamic programming formulation of the stochastic inventory control problem.

In order to solve this problem numerically, first and foremost one needs a way to efficiently evaluate its objective function, and, in if gradient-based optimization methods are to be used, its gradient with respect to the optimization variables. This means that we need to be able to evaluate the objective function value of the dynamic network availability control problem as a function of prices and fare restrictions. This is a two-step process: Standard capacity control methods are built on the assumption that one has a model for demand per booking class as a function of availability. Therefore, our objective function is actually the composition of two functions: A function that maps prices and product attributes to demand per booking class, followed by the mapping of this demand to the optimal objective function value of the capacity control problem.

The mapping of fares plus and customer choice model to a model of demand per booking class is the so-called aggregation problem of customer choice. Given an offer set consisting of one or multiple products, each defined by their attributes, the probability that a random customer will purchase a certain product is the probability measure of the set of customer preference vectors for which this product will have the highest utility, and can be expressed as a higher dimensional integral over the space of preference vectors. For general models such as the one we are using this probability cannot be computed analytically, and in practice is often estimated using Monte-Carlo-simulation. However, the variance arising from the simulation approach and makes it difficult to compute the gradient of demand w.r.t. prices and product attributes and, in addition, can lead to non-convexity in the overall objective function. We therefore opt for a deterministic method to compute these expected values. In Chapter 7 we show that the gradient of expected demand w.r.t. to the fares can also be expressed as a higher dimensional integral, and describe how—for models of reasonably low complexity—both expected demand and its gradient can efficiently be computed deterministically using a higher-dimensional quadrature algorithm.

Viewing the inventory control problem as a parametric optimization problem, parameterized with demand and price for each booking class, we now need to be able to compute its optimal objective function value and its gradient with respect to the parameters. This of course requires solving the classic RM problem, which for our specific problem formulation involves the solution of a dynamic program.

In case of a single resource, the availability control dynamic program can be solved efficiently, but standard solution algorithms are not focused on sensitivity analysis with respect to demand and prices and therefore do not easily give access to the gradient of the objective function value. In Chapter 5 we present a detailed analysis of the single-leg dynamic availability control problem. We perform a sensitivity analysis of the problem w.r.t the constant parameters and show how both the objective function value and its gradient can be computed efficiently using numerical methods.

These results cannot be directly transferred to the multi-resource case, because the network dynamic program suffers from the curse of dimensionality and cannot be optimally solved for realistically sized networks. There exist a number of high quality heuristics, which are computationally efficient and are widely applied in industry practice. However, these heuristics only generate close-to-optimal controls, but no estimate for overall expected network revenue, which is the objective function value we would like to evaluate. We review the so-called LP-DP decomposition heuristic in Chapter 6. It works with the following three steps: First, one solves a deterministic approximation of the network problem. The dual solution of this problem is then used to decompose the network into a large number of single-resource problems, which can be solved independently via dynamic programming. The solutions of these dynamic programs are then combined in a control scheme that is used to steer booking class availability throughout the booking horizon. We then show how the solutions of the single-leg dynamic programs can be used to efficiently compute an estimate of overall network revenue and its gradient with respect to demand and prices per booking class. In addition we introduce an improved version of the decomposition, which is better able to capture the stochastic nature of demand and leads to significantly increased revenue. In a simulation study we compare our method of approximating expected overall network revenue with actual revenue that was achieved in the simulation. The results show that our method is able to predict expected revenue much more accurately than upper bounds that are a by-product of the decomposition and which we use as a comparison benchmark. The simulation also shows that the improved network decomposition not only significantly increases expected network revenue but also further improves the quality of the objective function value estimate.

In Chapter 8 we combine all methods presented in this thesis in order to solve an instance of the pricing problem. Here, we do not intend to show expected revenue improvements of our methods over the alternatives, because there is no objective comparison benchmark: We do not know of any alternative numerical solution methods to the network airline pricing problem, and in industry practice pricing is based on manual decisions made by experts. We therefore only solve a small number of problem instances, which only differ in the number of products they are using, and qualitatively analyze the results. We show that even a single-leg case problem with only a handful of products is highly non-convex.

Chapter 1

Numerical tools

In this chapter we will state some important results from the field of numerical analysis, which will be applied to the airline pricing problem in the later chapters. First, in Section 1.1 we will cover basic optimality conditions as well as some results on solution sensitivity for nonlinear optimization problems. In Section 1.2 we give an overview over the theory and algorithms for the numerical solution of differential equations. Section 1.3 combines both and summarizes important results concerning nonlinear optimal control problems.

Throughout this thesis, we will use the following notational conventions:

- If $f\colon \mathbb{R}^n \to R, x \mapsto f(x)$ is a continuously differentiable real-valued function, $\nabla f = \left(\frac{\partial f}{\partial x}\right)^\top$ will denote the gradient of f.
- If f is twice continuously differentiable, $\nabla^2 f = \frac{\partial^2 f}{\partial x^2}$ denotes the Hessian of f.
- If $f\colon \mathbb{R}^n \times \mathbb{R}^m \to R, (x, y) \mapsto f(x, y)$ is a continuously differentiable real-valued function of two variables, $\nabla_x f = \left(\frac{\partial f}{\partial x}\right)^\top$ will denote the gradient of f with respect to x. Likewise, $\nabla_y f$ will denote the gradient of f w.r.t. y and, of course, analogously for functions of more than two variables.
- If f is twice continuously differentiable, $\nabla_x^2 f = \frac{\partial^2 f}{\partial x^2}$ denotes the Hessian of f w.r.t. x and likewise for $\nabla_y^2 f$.
- If $f\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l, (x, y) \mapsto f(x, y)$ is a continuously differentiable vector-valued function of two variables, $f_x = \frac{\partial f}{\partial x}$ will denote the partial derivative of f with respect to x, and likewise for y and other arguments of f.

1.1 Nonlinear optimization

The results presented in this section are standard material and can be found in most textbooks and lecture notes on nonlinear programming (cf. [15, 10]).

Definition 1.1.1 (Nonlinear optimization problem) A *nonlinear optimization problem* or Nonlinear Program (NLP) is the problem of minimizing (or maximizing) an *objective function* $f\colon \mathbb{R}^n \to \mathbb{R}$ over all admissible choices for the optimization variable $\mathbf{x} \in \mathbb{R}^n$ in the *feasible set* $X \subseteq \mathbb{R}^n$. The NLP is usually formulated as

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) = 0 \\ & h(\mathbf{x}) \le 0, \end{aligned} \tag{NLP}$$

where the feasible set X is described by nonlinear vector-valued *equality* and *inequality constraint* functions $g\colon \mathbb{R}^n \to \mathbb{R}^{n_{\text{eq}}}$ and $h\colon \mathbb{R}^n \to \mathbb{R}^{n_{\text{ineq}}}$ respectively. The inequality constraints h are meant component-wise and often contain *simple bounds* on the variable $\mathbf{x}$ of the type

$$\mathbf{x}_j - \mathbf{x}_j^{\mathrm{ub}} \leq 0 \tag{1.1a}$$

$$\mathbf{x}_j^{\mathrm{lb}} - \mathbf{x}_j \leq 0. \tag{1.1b}$$

Considering that maximization can be achieved by minimizing the negative of the objective function, we restrict ourselves to the minimization problem. For the rest of this chapter we will assume that the functions f, g and h are continuously differentiable.

Definition 1.1.2 (Solutions of an NLP) A *solution* of (**NLP**) is a vector $\mathbf{x} \in \mathbb{R}^n$ that satisfies the constraints $g(\mathbf{x}) = 0$ and $h(\mathbf{x}) \leq 0$. We say that $\mathbf{x}$ is *feasible* for (**NLP**).

A solution $\mathbf{x}^*$ is *optimal* or *globally optimal* for (**NLP**), if

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \tag{1.2}$$

for every solution $\mathbf{x}$.

A solution $\mathbf{x}^*$ is *locally optimal*, if there exists a $\delta > 0$, such that Eq. (1.2) holds for every solution $\mathbf{x}$ that satisfies

$$\|\mathbf{x}^* - \mathbf{x}\| \leq \delta. \tag{1.3}$$

A solution $\mathbf{x}^*$ is a strict (local) optimum, if Eq. (1.2) holds with strict inequality for every $\mathbf{x} \neq \mathbf{x}^*$.

A solution $\mathbf{x}^*$ is an isolated local optimum, if there exists a neighborhood U of $\mathbf{x}^*$, such that $\mathbf{x}^*$ is the only local optimum of (**NLP**) in U.

For a given solution, it is often necessary to distinguish between inequality constraints that are satisfied with equality and those that have *slack*.

Definition 1.1.3 (Active constraints, active set) Let $\mathbf{x}$ be a solution of (**NLP**). The k-th inequality constraint h_k is *active* at $\mathbf{x}$, if $h_k(\mathbf{x}) = 0$, and *inactive* if $h_k(\mathbf{x}) < 0$.

The *active set* at $\mathbf{x}$ is the set of all active constraints at $\mathbf{x}$:

$$I(\mathbf{x}) = \{k = 1, \ldots, n_{\mathrm{ineq}} \mid h_k(\mathbf{x}) = 0\}. \tag{1.4}$$

1.1.1 Optimality conditions

When studying necessary and sufficient optimality conditions for a solution of an NLP, we require the constraints to satisfy certain regularity conditions. The strongest (and simplest) are the following:

Definition 1.1.4 (Linear independence constraint qualification) Let $\mathbf{x}$ be a solution of **NLP**. Linear Independence Constraint Qualification (LICQ) holds at $\mathbf{x}$, if the gradients

$$\nabla g_k(\mathbf{x}),\ k = 1, \ldots n_{\mathrm{eq}} \tag{1.5}$$

$$\nabla h_k(\mathbf{x}),\ k \in I(\mathbf{x}) \tag{1.6}$$

of the equality constraints and the active inequality constraints are linearly independent.

Definition 1.1.5 (Lagrangian) We call the function

$$\mathcal{L}\colon \mathbb{R}^n \times \mathbb{R}^{n_{\mathrm{eq}}} \times \mathbb{R}^{n_{\mathrm{ineq}}} \to \mathbb{R} \tag{1.7a}$$

$$(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mapsto \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := f(\mathbf{x}) + \boldsymbol{\lambda}^\top g(\mathbf{x}) + \boldsymbol{\mu}^\top h(\mathbf{x}) \tag{1.7b}$$

the *Lagrangian* function of (**NLP**). The vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are called the *Lagrange multipliers* associated to the equality and inequality constraints respectively.

Theorem 1.1.6 (Karush-Kuhn-Tucker (KKT) conditions)
*Let $\mathbf{x}^*$ be a locally optimal solution of (**NLP**), such that LICQ holds at $\mathbf{x}^*$. Then there exist KKT-multipliers $\boldsymbol{\lambda}^* \in \mathbb{R}^{n_{eq}}$ and $\boldsymbol{\mu}^* \in \mathbb{R}^{n_{ineq}}$, such that the triple $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ satisfies*

- *the* stationarity *condition*

$$0 = \frac{\partial \mathcal{L}}{\partial \mathbf{x}}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = f_{\mathbf{x}}(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} g_{\mathbf{x}}(\mathbf{x}^*) + \boldsymbol{\mu}^{*\top} h_{\mathbf{x}}(\mathbf{x}^*), \tag{1.8a}$$

- *the* primal feasibility *conditions*

$$0 = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = g(\mathbf{x}^*), \tag{1.8b}$$

$$0 \geq \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = h(\mathbf{x}^*), \tag{1.8c}$$

- *the* dual feasibility *conditions*

$$0 \leq \boldsymbol{\mu}^*, \tag{1.8d}$$

- *and the* complementary slackness *condition*

$$0 = \boldsymbol{\mu}^{*\top} h(\mathbf{x}^*). \tag{1.8e}$$

A proof was first published by Kuhn and Tucker in 1951 [70], but can also be found in Karush's 1939 masters thesis [67].

Remark 1.1.7 The combination of Eqs. (1.8c) and (1.8d) and the complementary slackness condition Eq. (1.8e) implies that

$$0 = \boldsymbol{\mu}_k^* h_k(\mathbf{x}^*) \tag{1.9}$$

holds component-wise, i.e. for every $k = 1, \ldots n_{\text{ineq}}$: $\boldsymbol{\mu}_k^* = 0$ or $h_k(\mathbf{x}^*) = 0$.

Definition 1.1.8 (Strict Complementary Slackness (SCS)) Let $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ satisfy the KKT conditions for (**NLP**). We say that *strict complementary slackness* holds, if for every $k = 1, \ldots, n_{\text{ineq}}$: $\boldsymbol{\mu}_k = 0 \Rightarrow h_k(\mathbf{x}) > 0$ or, in other words, for every k exactly one of $\boldsymbol{\mu}_k$ and $h_k(\mathbf{x})$ is equal to zero.

Remark 1.1.9 In the absence of inequality constraints, the KKT conditions are just the well-known *Lagrange conditions* for equality constrained optimization problems.

In Theorem 1.1.6, the condition that LICQ holds at $\mathbf{x}^*$ can be replaced by weaker assumptions. Generally, we call any set of alternative sufficient conditions for the statement of Theorem 1.1.6 *Constraint Qualifications (CQ)*. Notable alternative CQs are the Constant Rank Constraint Qualification (CRCQ) and the Mangasarian-Fromovitz Constraint Qualification (MFCQ):

Definition 1.1.10 (CRCQ) Let $\mathbf{x}$ be a solution of **NLP**. CRCQ holds at $\mathbf{x}$, if there exists an open neighborhood U of $\mathbf{x}$, such that for all subsets $J \subseteq \{1, \ldots, n_{\text{eq}}\}$ and $I' \subseteq I(\mathbf{x})$ of the constraints, the corresponding subset of gradient vectors of the constraints has constant rank on U, in other words the family of vectors

$$\begin{aligned} &\nabla g_k(\mathbf{x}'), \, k \in J \\ &\nabla h_k(\mathbf{x}'), \, k \in I' \end{aligned}$$

has the same rank (which depends on J, I') for all $\mathbf{x}' \in U$.

Definition 1.1.11 (MFCQ) Let $\mathbf{x}$ be a solution of **NLP**. MFCQ holds at $\mathbf{x}$, if

(a) the gradients of the equality constraints are linearly independent, i.e. $\operatorname{rank}(\nabla g(\mathbf{x})) = n_{\text{eq}}$,

(b) there exists a vector $v \in \mathbb{R}^n$, such that

$$\begin{aligned} v^\top \nabla g(\mathbf{x}) &= 0, \\ v^\top \nabla h_k(\mathbf{x}) &> 0 \qquad \forall k \in I(\mathbf{x}). \end{aligned}$$

We say that the gradients of the equality constraints and the active inequality constraints are *positively linearly independent.*

For proof that MFCQ and CRCQ are indeed sufficient for Theorem 1.1.6 the reader is referred to the articles of Mangasarian and Fromovitz [83] and Janin [65] respectively.

Remark 1.1.12 If LICQ holds at a local optimum $\mathbf{x}^*$, it can be shown that the Lagrange multipliers $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$ are uniquely defined by the KKT conditions.

In addition to the first order KKT-conditions, we can state the following Second Order Necessary Conditions (SONCs) for local optimality:

Theorem 1.1.13 (SONCs)
Let $\mathbf{x}^*$ be a local minimum of ($\boldsymbol{NLP}$) and assume that LICQ holds at $\mathbf{x}^*$. Let $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$ be the Lagrange multipliers, such that the KKT conditions Eq. (1.8) *are satisfied. Furthermore, assume that f, g and h are twice continuously differentiable in a neighborhood of $\mathbf{x}^*$. Then*

$$v\nabla^2_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)v \geq 0 \tag{1.10a}$$

for every $v \in \mathbb{R}^n$ that satisfies

$$v^\top \nabla g_k(\mathbf{x}^*) = 0 \qquad \forall k = 1, \dots, n_{eq} \tag{1.10b}$$
$$v^\top \nabla h_k(\mathbf{x}^*) \leq 0 \qquad \forall k \in I(\mathbf{x}^*) \tag{1.10c}$$
$$v^\top \nabla h_k(\mathbf{x}^*) = 0 \qquad \forall k\colon \boldsymbol{\mu}_k^* < 0. \tag{1.10d}$$

A triple $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$, which satisfies Eqs. (1.8) and (1.10) only provides a *candidate* $\mathbf{x}^*$ for a local minimum, but can be a local maximum or a saddle point as well. Local optimality can be verified under additional assumptions.

Theorem 1.1.14 (Second Order Sufficient Conditions (SOSCs))
Let $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ satisfy Eq. (1.8) *and assume that f, g and h are twice continuously differentiable in a neighborhood of $\mathbf{x}^*$. Moreover, assume that*

$$v\nabla^2_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)v > 0 \tag{1.11}$$

for every $0 \neq v \in \mathbb{R}^n$ that satisfies Eqs. (1.10b) *to* (1.10d)*. Then $\mathbf{x}^*$ is a local minimum of ($\boldsymbol{NLP}$).*

The assumption that LICQ holds can be replaced by weaker conditions [36, 106].
For proofs the reader is referred the book if Fiacco and McCormick [37].

1.1.2 Sensitivity analysis

In this section we will summarize some important results regarding solution sensitivity for the *parametric* NLP

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}, \mathbf{p}) \\ \text{s.t.} \quad & g(\mathbf{x}, \mathbf{p}) = 0 \\ & h(\mathbf{x}, \mathbf{p}) \leq 0, \end{aligned} \tag{$\mathbf{NLP}(\mathbf{p})$}$$

where the objective function $f\colon \mathbb{R}^n \times \mathbb{R}^{n_\mathrm{p}} \to \mathbb{R}$, equality constraints $g\colon \mathbb{R}^n \times \mathbb{R}^{n_\mathrm{p}} \to \mathbb{R}^{n_\mathrm{eq}}$, and inequality constraints $h\colon \mathbb{R}^n \times \mathbb{R}^{n_\mathrm{p}} \to \mathbb{R}^{n_\mathrm{ineq}}$ depend on a parameter vector $\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}}$. In other words, to each choice of a parameter vector $\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}}$ we associate an instance of ($\mathbf{NLP}$), which we will denote by ($\mathbf{NLP}(\mathbf{p})$).

For the remainder of this section, assume that f, g and h are twice continuously differentiable and let $\mathbf{p}_0$ be a fixed parameter vector. Let $\mathbf{x}_0^*$ be a local minimum of the associated problem ($\mathbf{NLP}(\mathbf{p}_0)$), and let the triple $(\mathbf{x}_0^*, \boldsymbol{\lambda}_0^*, \boldsymbol{\mu}_0^*)$ satisfy the KKT conditions. Under suitable regularity conditions, the perturbed problem ($\mathbf{NLP}(\mathbf{p})$), for $\mathbf{p}$ in a small neighborhood of $\mathbf{p}_0$, has a local minimum in the neighborhood of $\mathbf{x}_0^*$. Moreover, both the local minimizer $\mathbf{x}^*$ and the multipliers $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$ are continuously differentiable w.r.t. $\mathbf{p}$, if certain constraint qualifications hold.

To convey the idea underlying NLP sensitivity analysis, we will first consider a problem without inequality constraints

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x},\mathbf{p}) \\ \text{s.t.} \quad & g(\mathbf{x},\mathbf{p}) = 0 \end{aligned} \tag{\textbf{NLP'}(\mathbf{p})}$$

and prove some basic results. Let $\mathbf{z} = (\mathbf{x}, \boldsymbol{\lambda})$. The Lagrangian function is then given by

$$\mathcal{L}(\mathbf{z},\mathbf{p}) = \mathcal{L}(\mathbf{x},\boldsymbol{\lambda},\mathbf{p}) = f(\mathbf{x},\mathbf{p}) + \boldsymbol{\lambda}^\top g(\mathbf{x},\mathbf{p}). \tag{1.12}$$

Theorem 1.1.15
*If, for $\mathbf{p}_0$ and $\mathbf{z}_0 = (\mathbf{x}_0^*, \boldsymbol{\lambda}_0^*)$, the KKT conditions and the SOSCs for **NLP'**($\mathbf{p}_0$) are satisfied and LICQ holds at $\mathbf{x}_0^*$, then there exists a unique continuously differentiable function*

$$\mathbf{z}^*\colon U \subset \mathbb{R}^{n_p} \to \mathbb{R}^n \times \mathbb{R}^{n_{eq}} \tag{1.13a}$$

$$\mathbf{p} \mapsto \mathbf{z}^*(\mathbf{p}) = (\mathbf{x}^*(\mathbf{p}), \boldsymbol{\lambda}^*(\mathbf{p})) \tag{1.13b}$$

defined on a neighborhood U of $\mathbf{p}_0$, such that

(a) $\mathbf{z}^*(\mathbf{p}_0) = \mathbf{z}_0^*$,

(b) $\mathbf{x}^*(\mathbf{p})$ *and* $\boldsymbol{\lambda}^*(\mathbf{p})$ *satisfy the KKT conditions for Eq.* (**NLP'**($\mathbf{p}$)),

(c) $\mathbf{x}^*(\mathbf{p})$ *is a local minimum of Eq.* (**NLP'**($\mathbf{p}$)).

Proof For every $\mathbf{p}$, the KKT conditions for a local optimum $\mathbf{z}^* = (\mathbf{x}^*, \boldsymbol{\lambda}^*)$ of Eq. (**NLP'**($\mathbf{p}$)) are given by the system of nonlinear equations

$$0 = G(\mathbf{z},\mathbf{p}) = \nabla_{\mathbf{z}}\mathcal{L}(\mathbf{z}^*,\mathbf{p}) = \begin{pmatrix} \nabla_{\mathbf{x}} f(\mathbf{x}^*,\mathbf{p}) + \sum_{k=1}^{n_{\text{eq}}} \boldsymbol{\lambda}_k^* \nabla_{\mathbf{x}} g_k(\mathbf{x}^*,\mathbf{p}) \\ g(\mathbf{x}^*,\mathbf{p}) \end{pmatrix}. \tag{1.14}$$

In particular, because the KKT conditions hold at $\mathbf{p}_0$ and $\mathbf{z}_0 = (\mathbf{x}_0^*, \boldsymbol{\lambda}_0)$, $G(\mathbf{z}_0^*, \mathbf{p}_0) = 0$.

Taking the derivative of Eq. (1.14) w.r.t. $\mathbf{z}^*$, we obtain

$$G_{\mathbf{z}^*}(\mathbf{z}^*,\mathbf{p}) = \nabla_{\mathbf{z}}^2\mathcal{L}(\mathbf{z}^*,\mathbf{p}) = \begin{pmatrix} \nabla_{\mathbf{x}}^2 f(\mathbf{x}^*,\mathbf{p}) & g_{\mathbf{x}}^\top(\mathbf{x}^*,\mathbf{p}) \\ g_{\mathbf{x}}(\mathbf{x}^*,\mathbf{p}) & 0 \end{pmatrix}. \tag{1.15}$$

By assumption, the SOSCs hold at $\mathbf{p}_0$ and $\mathbf{z}_0 = (\mathbf{x}_0^*, \boldsymbol{\lambda}_0)$. In other words, $\nabla_{\mathbf{x}}^2 f(\mathbf{x}_0^*, \mathbf{p}_0)$ is positive definite on the kernel of $g_{\mathbf{x}}(\mathbf{x}_0^*, \mathbf{p}_0)$. In particular, $\nabla_{\mathbf{x}}^2 f(\mathbf{x}_0^*, \mathbf{p}_0)$ is linearly independent of the rows of $g_{\mathbf{x}}(\mathbf{x}_0^*, \mathbf{p}_0)$. Moreover, because LICQ holds, $g_{\mathbf{x}}(\mathbf{x}_0^*, \mathbf{p}_0)$ has full rank and therefore $G_{\mathbf{z}^*}(\mathbf{z}_0^*, \mathbf{p}_0)$ has full rank as well.

As a consequence, the implicit function theorem locally guarantees the existence of a continuously differentiable function $\mathbf{z}^*$ of the form Eq. (1.13), such that for every $\mathbf{p} \in U\colon G(\mathbf{z}^*(\mathbf{p}), \mathbf{p}) = 0 \Rightarrow$ (b). Clearly, $\mathbf{z}^*(\mathbf{p}_0) = \mathbf{z}_0^* \Rightarrow$ (a). Furthermore, the assumption that $\nabla_{\mathbf{x}}^2 f$ is positive definite on the kernel of $g_{\mathbf{x}}$ is an open condition and therefore holds in an open neighborhood V of $(\mathbf{x}_0^*, \mathbf{p}_0)$. Since $\mathbf{x}^*(\mathbf{p})$ is continuous in $\mathbf{p}$, for a sufficiently small neighborhood U of $\mathbf{p}_0$ we have $\mathbf{x}^*(U) \times U \subset V$. Therefore, for all $\mathbf{p} \in U\colon \mathbf{z}^*(\mathbf{p})$ satisfies the SOSCs $\Rightarrow$ (c). $\square$

Of course, the derivatives of $\mathbf{x}^*$ and $\boldsymbol{\lambda}^*$ w.r.t. $\mathbf{p}$ can be deduced from the implicit function theorem. Because the airline pricing problem will later be formulated as a two-tier hierarchical model, in which the controls for the outer problem occur as fixed parameters for the inner problem, we are mainly concerned with the sensitivity of the optimal objective function value, and not so much with the sensitivity of the solution vectors.

Therefore, for an open neighborhood U of $\mathbf{p}_0$, consider the *locally optimal value function*

$$\begin{aligned} F\colon U \subset \mathbb{R}^{n_\mathrm{p}} &\to \mathbb{R} \\ \mathbf{p} &\mapsto F(\mathbf{p}) \coloneqq f(\mathbf{x}^*(\mathbf{p}), \mathbf{p}), \end{aligned}$$

mapping a value of $\mathbf{p}$ near $\mathbf{p}_0$ to the objective function value at the local minimum constructed in Theorem 1.1.15. Because $\mathbf{x}^*(\mathbf{p})$ and $f(\mathbf{x}, \mathbf{p})$ are continuously differentiable, so is F.

Proposition 1.1.16
With the same notation as above,

$$\frac{\partial F}{\partial \mathbf{p}}(\mathbf{p}_0) = \frac{\partial \mathcal{L}}{\partial \mathbf{p}}(\mathbf{z}_0, \mathbf{p}_0) = \frac{\partial f}{\partial \mathbf{p}}(\mathbf{x}_0^*, \mathbf{p}_0) + \boldsymbol{\lambda}_0^{*\top} \frac{\partial g}{\partial \mathbf{p}}(\mathbf{x}_0^*, \mathbf{p}_0). \tag{1.16}$$

Proof By construction, for all $\mathbf{p} \in U$, the solution $\mathbf{x}^*(\mathbf{p})$ is feasible for ($\mathbf{NLP}(\mathbf{p})$) and therefore $g(\mathbf{x}^*(\mathbf{p}), \mathbf{p}) \equiv 0$ is constant. Taking the derivative w.r.t. $\mathbf{p}$ at $\mathbf{p} = \mathbf{p}_0$ yields

$$\begin{aligned} 0 &= \frac{\partial g}{\partial \mathbf{x}}(\mathbf{x}^*(\mathbf{p}_0), \mathbf{p}_0) \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}}(\mathbf{p}_0) + \frac{\partial g}{\partial \mathbf{p}}(\mathbf{x}^*(\mathbf{p}_0), \mathbf{p}_0) && (1.17\text{a}) \\ &= g_\mathbf{x}(\mathbf{x}_0^*, \mathbf{p}_0)\mathbf{x}_\mathbf{p}^*(\mathbf{p}_0) + g_\mathbf{p}(\mathbf{x}_0^*, \mathbf{p}_0). && (1.17\text{b}) \end{aligned}$$

Furthermore, because $\mathbf{x}_0^*$ is locally optimal for $\mathbf{NLP'}(\mathbf{p}_0)$, he have the stationarity condition

$$0 = f_\mathbf{x}(\mathbf{x}_0^*, \mathbf{p}_0) + \boldsymbol{\lambda}_0^{*\top} g_\mathbf{x}(\mathbf{x}_0^*, \mathbf{p}_0). \tag{1.18}$$

Applying the chain rule we have

$$\begin{aligned} \frac{\partial F}{\partial \mathbf{p}}(\mathbf{p}_0) &= \frac{\partial f}{\partial \mathbf{p}}(\mathbf{x}^*(\mathbf{p}_0), \mathbf{p}_0) + \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}^*(\mathbf{p}_0), \mathbf{p}_0) \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}}(\mathbf{p}_0) \\ &= f_\mathbf{p}(\mathbf{x}_0^*, \mathbf{p}_0) + f_\mathbf{x}(\mathbf{x}_0^*, \mathbf{p}_0)\mathbf{x}_\mathbf{p}^*(\mathbf{p}_0) \\ &= f_\mathbf{p}(\mathbf{x}_0^*, \mathbf{p}_0) - \boldsymbol{\lambda}_0^{*\top} g_\mathbf{x}(\mathbf{x}_0^*, \mathbf{p}_0)\mathbf{x}_\mathbf{p}^*(\mathbf{p}_0) \\ &= f_\mathbf{p}(\mathbf{x}_0^*, \mathbf{p}_0) + \boldsymbol{\lambda}_0^{*\top} g_\mathbf{p}(\mathbf{x}_0^*, \mathbf{p}_0), \end{aligned}$$

where the last two equalities follow from Eqs. (1.18) and (1.17b) respectively □

Remark 1.1.17 Note that the proof of Proposition 1.1.16 does not require any knowledge about the sensitivities of the controls $\mathbf{x}^*$, other than the fact that they exist. Similarly, Eq. (1.16) allows us to compute $\frac{\partial F}{\partial \mathbf{p}}$ without having to evaluate $\frac{\partial \mathbf{x}^*}{\partial \mathbf{p}}$.

Equation (1.16) is an extension of the well-known envelope theorem from unconstrained optimization, which states that

$$\left.\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}_0} \max\{f(\mathbf{x}, \mathbf{p}) \mid \mathbf{x} \in \mathbb{R}^n\} = \frac{\partial f}{\partial \mathbf{p}}(\mathbf{x}_0^*, \mathbf{p}_0),$$

where $\mathbf{x}_0^* = \arg\max\{f(\mathbf{x}, \mathbf{p}_0) \mid \mathbf{x} \in \mathbb{R}^n\}$ is the optimum of the unperturbed problem.

In the presence of inequality constraints, similar results still hold under suitable conditions, and the analysis is very similar to the equality constrained case. However, care has to be taken concerning the behavior of the active set. We only give a rough informal overview and refer the reader to the relevant literature for a detailed treatment. Under the strongest conditions, we obtain a straight-forward generalization of Theorem 1.1.15:

Theorem 1.1.18
*If, for $\mathbf{p}_0$ and $\mathbf{z}_0 = (\mathbf{x}_0^*, \boldsymbol{\lambda}_0^*, \boldsymbol{\mu}_0^*)$, the KKT conditions and the SOSCs for **NLP**($\mathbf{p}_0$) are satisfied, LICQ holds at $\mathbf{x}_0^*$, and the SCS condition holds, then there exists a unique continuously differentiable function*

$$\begin{aligned} \mathbf{z}^*\colon U \subset \mathbb{R}^{n_p} &\to \mathbb{R}^n \times \mathbb{R}^{n_{eq}} \times \mathbb{R}^{n_{ineq}} && (1.19\text{a}) \\ \mathbf{p} &\mapsto \mathbf{z}^*(\mathbf{p}) = (\mathbf{x}^*(\mathbf{p}), \boldsymbol{\lambda}^*(\mathbf{p}), \boldsymbol{\mu}^*(\mathbf{p})) && (1.19\text{b}) \end{aligned}$$

defined on a neighborhood U of $\mathbf{p}_0$, such that *(a)* $\mathbf{z}^*(\mathbf{p}_0) = \mathbf{z}_0^*$,

(b) $\mathbf{x}^*(\mathbf{p})$, $\boldsymbol{\lambda}^*(\mathbf{p})$ *and* $\boldsymbol{\mu}^*(\mathbf{p})$ *satisfy the KKT conditions for (**NLP**($\mathbf{p}$)),*

(c) $\mathbf{x}^*(\mathbf{p})$ *is a local minimum of (**NLP**($\mathbf{p}$)).*

Proof See [38]. □

The proof relies on the fact that, under these strong assumptions, the active set does not change for small perturbations of $\mathbf{p}_0$ and $\mathbf{x}_0^*$. This allows to (locally) treat the active inequalities as equality constraints and ignore the inactive constraints, therefore reducing the problem to the situation of Theorem 1.1.15. Again the sensitivities of $\mathbf{x}^*(\mathbf{p})$, $\boldsymbol{\lambda}^*(\mathbf{p})$ and $\boldsymbol{\mu}^*(\mathbf{p})$ at $\mathbf{p} = \mathbf{p}_0$ can be derived using the implicit function theorem.

The assumptions in Theorem 1.1.18 can be relaxed in several ways. Jittorntrum [66] and Robinson [106] show that SCS is not necessary if the SOSCs are replaced by slightly stronger assumptions. As shown by Kojima [69], the LICQ condition can be dropped as well if the SOSCs are strengthened further.

Under weaker assumptions, the solution $\mathbf{x}^*(\mathbf{p})$ and the multipliers $\boldsymbol{\lambda}^*(\mathbf{p})$ and $\boldsymbol{\mu}^*(\mathbf{p})$ are not necessarily continuously differentiable or even unique. Ralph and Dempe [103] that they are continuous and PC^1, which is a kind of higher-dimensional piecewise differentiability[1], if $\mathbf{x}_0^*$ is an isolated local optimum of $\mathbf{NLP}(\mathbf{p}_0)$ and ($\mathbf{NLP}(\mathbf{p})$) has a unique local optimum near $\mathbf{x}_0^*$ for small perturbations of $\mathbf{p}$ around $\mathbf{p}_0$. Further generalizations with multi-valued maps $\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$ have been developed, an overview can be found in the review article of Bonnans and Shapiro [13].

In particular, as a generalization of Proposition 1.1.16, under mild regularity assumptions the optimal value function is PC^1 and directionally differentiable. Where it exists, its gradient is given by

$$\frac{\partial F}{\partial \mathbf{p}}(\mathbf{p}_0) = \frac{\partial \mathcal{L}}{\partial \mathbf{p}}(\mathbf{x}_0^*, \boldsymbol{\lambda}_0^*, \boldsymbol{\mu}_0^*, \mathbf{p}_0) \tag{1.20a}$$

$$= \frac{\partial f}{\partial \mathbf{p}}(\mathbf{x}_0^*, \mathbf{p}_0) + \boldsymbol{\lambda}_0^{*\top} \frac{\partial g}{\partial \mathbf{p}}(\mathbf{x}_0^*, \mathbf{p}_0) + \boldsymbol{\mu}_0^{*\top} \frac{\partial h}{\partial \mathbf{p}}(\mathbf{x}_0^*, \mathbf{p}_0). \tag{1.20b}$$

The reader is referred to Bonnans and Shapiro [13] for a proof and further references. Again, if the Lagrange multipliers are known, the gradient of the optimal value function F w.r.t. the parameters $\mathbf{p}$ can be computed without knowledge of the sensitivities of the optimal controls $\mathbf{x}^*$.

1.2 Differential equations

In this section we will review some of the basic techniques that are used to numerically solve ordinary differential equations (ODEs). More specifically, we consider initial value problems (IVPs) of the form

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t)) \tag{1.21a}$$

$$\mathbf{y}(t_0) = \mathbf{y}_0, \tag{1.21b}$$

where $\mathbf{y}\colon [t_0, T] \subset \mathbb{R} \to \mathbb{R}^{n_y}$ are the states and the right-hand side (RHS) is given by a function $\mathbf{f}\colon \mathbb{R} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_y}$. We will always assume that $\mathbf{f}$ is continuous in t and Lipschitz continuous in $\mathbf{y}$. Under these assumptions, the Picard-Lindelöf theorem ensures that Eq. (1.21) has a unique solution, which we want to approximate numerically. Proofs for the statements in this section and a more detailed treatment of the matter can be found in standard textbooks on numerical analysis [113, 16].

We will denote the solution of Eq. (1.21) with initial value $\mathbf{y}_0$ at t_0 by $\mathbf{y}(t; t_0, \mathbf{y}_0)$ when the dependence of $\mathbf{y}$ on the initial values are of importance.

[1] A function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is PC^1, if it is continuous and for every x_0 in its domain there exists an open neighborhood $U \subset \mathbb{R}^n$ of x_0 and a finite collection of continuously differentiable functions $f_i\colon U \to \mathbb{R}^m$, $i = 1, \dots, k$ such that $f(x) \in \{f_1(x), \dots f_k(x)\}$ for every $x \in U$. Properties of PC^1 functions are for example described by Chaney [26].

Definition 1.2.1 A *one-step method* for the solution of the IVP Eq. (1.21) is defined by an *increment function* $\Phi(t, h, u, \mathbf{f})$. For a given discretization of time with step sizes $h^{(i)}$, the method iteratively computes estimates $u^{(i)} \approx \mathbf{y}(t^{(i)})$ using the recursion

$$t^{(0)} = t_0 \tag{1.22}$$
$$u^{(0)} = \mathbf{y}_0 \tag{1.23}$$
$$t^{(i+1)} = t^{(i)} + h^{(i)} \tag{1.24}$$
$$u^{(i+1)} = u^{(i)} + h\Phi\left(t^{(i)}, h^{(i)}, u^{(i)}, \mathbf{f}\right). \tag{1.25}$$

Of course, in order to be of any practical use, the integration scheme should yield good estimates for the solution of Eq. (1.21).

Definition 1.2.2 A numerical integration method is called convergent, if the approximate trajectory converges to the exact solution as step size goes to 0, in other words if

$$\lim_{h\to 0} \max_i \left\| u^{(i)} - \mathbf{y}(t^{(i)}) \right\| = 0.$$

It is called convergent of order p, if

$$\max_i \left\| u^{(i)} - \mathbf{y}(t^{(i)}) \right\| = O(h^p)$$

as $h \to 0$.

The method is called consistent, if the *local error*

$$\delta^{(i)}(h) = u^{(i)} - \mathbf{y}(t^{(i)}; t^{(i-1)}, u^{(i-1)}) \tag{1.26}$$

for a step of length $h^{(i)} = h$ satisfies

$$\lim_{h\to 0} \frac{\delta^{(i)}(h)}{h} = 0. \tag{1.27}$$

The method is said to be *consistent of order* $p \in \mathbb{N}$[2], if $\delta^{(i)}(h) = O(h^{p+1})$ for $h \to 0$.

Remark 1.2.3 If a one-step method is consistent of order $p+1$, and $\mathbf{f}$ and the increment function Φ are Lipschitz continuous, then the method is convergent of order p.

The most common class of one-step methods are the methods of the Runge-Kutta family, first introduced by Runge [107] and Kutta [72].

Definition 1.2.4 A *Runge-Kutta method* with $s \in \mathbb{N}$ stages computes the increment as

$$\Phi(t, h, u, \mathbf{f}) = \sum_{i=1}^{s} c_i k_i, \tag{1.28}$$

where the intermediate values $k_1, \ldots, k_s$ are defined by the system of nonlinear equations

$$k_i = \mathbf{f}\left(t + a_i h, \mathbf{y} + h \sum_{j=1}^{s} b_{i,j} k_j\right). \tag{1.29}$$

The coefficients c, a and b are often arranged in a so-called *Butcher tableau* [23]

$$\begin{array}{c|ccc} a_1 & b_{1,1} & \cdots & b_{1,s} \\ \vdots & \vdots & & \vdots \\ a_s & b_{s,1} & \cdots & b_{s,s} \\ \hline & c_1 & \cdots & c_s \end{array}.$$

[2]Throughout this thesis the set of natural numbers $\mathbb{N}$ includes zero.

The method is called explicit, if the stages can be rearranged s.t. the matrix of coefficients $B = (b_{i,j})$ is lower-triangular:

$$\begin{array}{c|cccc} a_1 & & & & \\ a_2 & b_{2,1} & & & \\ \vdots & \vdots & \ddots & & \\ a_s & b_{s,1} & \cdots & b_{s,s-1} & \\ \hline & c_1 & \cdots & c_{s-1} & c_s \end{array}$$

In this case, in contrast to a general implicit method, the system Eq. (1.29) can be solved by directly computing the coefficients k_i one by one using s evaluations of $\mathbf{f}$.

Remark 1.2.5 A Runge-Kutta method is consistent, if for every $i = 1, \ldots, s$

$$\sum_{j=1}^{s} c_{i,j} = a_j. \tag{1.30}$$

The order of consistency of the Runge-Kutta method defined by a given set of parameters can be determined by comparing coefficients of a Taylor expansion of Φ around t. Conversely, given desired order $p+1$, this yields a system of linear equations for a, c, and B. If a Runge-Kutta method is consistent of order $p+1$ and $\mathbf{f}$ is Lipschitz continuous, then the method is convergent of order p.

The Butcher tableaus for some common explicit Runge-Kutta methods are shown in Fig. 1.1.

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

(a) First order Euler method

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(b) Second order Heun method

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

(c) Classic fourth order Runge-Kutta method.

Figure 1.1: Explicit Runge-Kutta Methods of different orders

1.2.1 Sensitivity analysis

Often $\mathbf{f}$, t_0 and $\mathbf{y}_0$, and therefore the solution $\mathbf{y}(t)$, of an IVP depend on a vector of parameters $\mathbf{p}$. Shifting time, we can assume w.l.o.g. that t_0 is independent of $\mathbf{p}$. We will denote the solution to the problem

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t), \mathbf{p}) \tag{1.31a}$$

$$\mathbf{y}(t_0) = \mathbf{y}_0(\mathbf{p}) \tag{1.31b}$$

by $\mathbf{y}(t; t_0, \mathbf{y}_0, \mathbf{p})$. If $\mathbf{f}$ is continuously differentiable in $\mathbf{p}$, then so is $\mathbf{y}$.

Forward sensitivity

We will first describe the so-called *forward sensitivity analysis*. By differentiating Eq. (1.31) w.r.t. $\mathbf{p}$, we obtain the *variational differential equation*

$$\dot{\mathbf{y}}_{\mathbf{p}}(t) = \mathbf{f}_{\mathbf{y}}(t, \mathbf{y}(t), \mathbf{p})\mathbf{y}_{\mathbf{p}}(t) + \mathbf{f}_{\mathbf{p}}(t, \mathbf{y}(t), \mathbf{p}) \tag{1.32a}$$

$$\mathbf{y}_{\mathbf{p}}(t_0) = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{y}_0(\mathbf{p}), \tag{1.32b}$$

where $\mathbf{y_p} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{y}(t;t_0,\mathbf{y}_0,\mathbf{p})$ and $\mathbf{f_y}$ and $\mathbf{f_p}$ are the partial derivatives of $\mathbf{f}$ w.r.t. $\mathbf{y}$ and $\mathbf{p}$ respectively. For the special case $\mathbf{y}_0(\mathbf{p}) = \mathbf{p}$ we get

$$\dot{W}(t) = \mathbf{f_y}(t,\mathbf{y}(t),\mathbf{p})W(t) \tag{1.33a}$$

$$W(t_0) = \mathbb{I}_{n_y}, \tag{1.33b}$$

where $W = \frac{\mathrm{d}}{\mathrm{d}\mathbf{y}_0}\mathbf{y}(t;t_0,\mathbf{y}_0,\mathbf{p})$ and $\mathbb{I}_{n_y}$ is the n_y-dimensional identity matrix.

Note that, independent of the structure of the initial problem, Eqs. (1.32) and (1.33) are systems of linear differential equations. If the parameter $\mathbf{p}$ is a scalar, the system Eq. (1.32) has the same number of differential states as the original system. If it is a vector, one such system has to be solved for each component of $\mathbf{p}$. Since W is a $n_y \times n_y$-matrix, Eq. (1.33) is a differential equation of dimension ${n_y}^2$.

Numerical solutions to Eqs. (1.32) and (1.33) are usually computed simultaneously with the solution of Eq. (1.31), because values of $\mathbf{y}$ are required to evaluate the RHS of the variational equations. Therefore, computing derivatives of $\mathbf{y}$ using forward sensitivity analysis requires the solution of an IVP of dimension $n_y(n_p+1)$ or $n_y + {n_y}^2$ respectively.

Although the special structure of the variational equations can be exploited, the forward approach is still inefficient if the number of parameters n_p is very large. Also note that during the solution of the forward system the whole trajectory for derivatives of all differential states is computed. In practice however, particularly when solving optimal control problems, it is often sufficient to compute the gradient of a single or a small number of real valued functionals depending on $\mathbf{y}$, instead of evaluating the full sensitivity system of $\mathbf{y}$ w.r.t. multiple control variables.

The objective is usually given as

$$g(\mathbf{p}) = \phi\left[T,\mathbf{y}\left(T(\mathbf{p});t_0,\mathbf{y}_0(\mathbf{p}),\mathbf{p}\right),\mathbf{p}\right] \tag{1.34}$$

with the Mayer functional ϕ, or as

$$G(\mathbf{p}) = \int_{t_0}^{T(\mathbf{p})} L\left[t,\mathbf{y}\left(t;t_0,\mathbf{y}_0(\mathbf{p}),\mathbf{p}\right),\mathbf{p}\right]\,\mathrm{d}t \tag{1.35}$$

with the Lagrange functional L, or a sum of both. If T explicitly depends on $\mathbf{p}$, we can make T constant via a linear transformation of time. For simplicity's sake we can therefore assume in the following w.l.o.g. that T is a constant.

In order to simplify notation in the following we will omit the parameters of $\phi[t,\mathbf{y},\mathbf{p}]$ and $L[t,\mathbf{y},\mathbf{p}]$. If ϕ and L are sufficiently smooth, the derivatives of g and G can straightforwardly be computed from Eqs. (1.34) and (1.35) to be

$$\frac{\mathrm{d}g}{\mathrm{d}\mathbf{p}}(\mathbf{p}) = \phi_\mathbf{y}\mathbf{y_p}(T) + \phi_\mathbf{p} \tag{1.36a}$$

$$\frac{\mathrm{d}G}{\mathrm{d}\mathbf{p}}(\mathbf{p}) = \int_{t_0}^{T} L_\mathbf{y}\mathbf{y_p} + L_\mathbf{p}\,\mathrm{d}t \tag{1.36b}$$

and can therefore be evaluated using Eq. (1.32) at the cost of the solution of an IVP of dimension $n_y + n_y n_p$ or $n_y + n_y n_p + 1$ respectively. Here, the computation of the trajectory of $\mathbf{y_p}$, which is an intermediate value that is not needed for its own sake, induces most of the computational cost.

Adjoint sensitivity

One can get around computing $\mathbf{y_p}$ using so-called *reverse* or *adjoint sensitivity analysis.* This approach can be motivated in various different ways.

In this work we choose to present the more descriptive motivation via simple differential calculus. While not as general, the results are sufficient for objective functions like those described in Eqs. (1.34) and (1.35) and, first and foremost, for the application at hand. In addition, the central idea of a *Lagrangian* is very natural in the context of nonlinear optimization.

Lagrange term
We will first consider an objective functional of the form of Eq. (1.35). Writing the constraints for the differential states (Eq. (1.31)) as

$$g(t, \mathbf{y}, \dot{\mathbf{y}}, \mathbf{p}) = \dot{\mathbf{y}}(t) - \mathbf{f}(t, \mathbf{y}(t), \mathbf{p}) = 0 \tag{1.37}$$

we form an augmented objective function

$$H(\mathbf{p}) = G(\mathbf{p}) - \int_{t_0}^{T} \boldsymbol{\lambda}^\top \underbrace{\left[\dot{\mathbf{y}}(t) - \mathbf{f}(t, \mathbf{y}(t), \mathbf{p})\right]}_{g(t,\mathbf{y},\dot{\mathbf{y}},\mathbf{p})} \,\mathrm{d}t \tag{1.38}$$

with time-dependent Lagrange multipliers, sometimes called *co-states*, $\boldsymbol{\lambda}$. First, note that with partial integration we have

$$\int_{t_0}^{T} \boldsymbol{\lambda}^\top \dot{\mathbf{y}}_\mathbf{p} \,\mathrm{d}t = \boldsymbol{\lambda}^\top \mathbf{y}_\mathbf{p}\Big|_{t=t_0}^{t=T} - \int_{t_0}^{T} \dot{\boldsymbol{\lambda}}^\top \mathbf{y}_\mathbf{p} \,\mathrm{d}t. \tag{1.39}$$

Then, because $g \equiv 0$,

$$\begin{aligned}
\frac{\mathrm{d}G}{\mathrm{d}\mathbf{p}} &= \frac{\mathrm{d}H}{\mathrm{d}\mathbf{p}} && \text{(1.40a)}\\
&= \int_{t_0}^{T} L_\mathbf{y}\mathbf{y}_\mathbf{p} + L_\mathbf{p} \,\mathrm{d}t - \int_{t_0}^{T} \boldsymbol{\lambda}^\top \left[\dot{\mathbf{y}}_\mathbf{p} - \mathbf{f}_\mathbf{y}\mathbf{y}_\mathbf{p} - \mathbf{f}_\mathbf{p}\right] \,\mathrm{d}t && \text{(1.40b)}\\
&\overset{(1.39)}{=} \int_{t_0}^{T} L_\mathbf{p} + \boldsymbol{\lambda}^\top \mathbf{f}_\mathbf{p} \,\mathrm{d}t + \int_{t_0}^{T} \left[L_\mathbf{y} + \boldsymbol{\lambda}^\top \mathbf{f}_\mathbf{y} + \dot{\boldsymbol{\lambda}}^\top\right] \mathbf{y}_\mathbf{p} \,\mathrm{d}t - \boldsymbol{\lambda}^\top \mathbf{y}_\mathbf{p}\Big|_{t=t_0}^{t=T}. && \text{(1.40c)}
\end{aligned}$$

In order to avoid having to compute a trajectory of $\mathbf{y}_\mathbf{p}$, we now require the co-states to satisfy the linear ODE

$$\begin{aligned}
\dot{\boldsymbol{\lambda}} &= -\mathbf{f}_\mathbf{y}^\top \boldsymbol{\lambda} - L_\mathbf{y}^\top && \text{(1.41a)}\\
\boldsymbol{\lambda}(T) &= 0. && \text{(1.41b)}
\end{aligned}$$

With Eqs. (1.41) and (1.40c) we have

$$\frac{\mathrm{d}G}{\mathrm{d}\mathbf{p}} = \int_{t_0}^{T} L_\mathbf{p} + \boldsymbol{\lambda}^\top \mathbf{f}_\mathbf{p} \,\mathrm{d}t + \boldsymbol{\lambda}^\top(t_0)\mathbf{y}_\mathbf{p}(t_0). \tag{1.42}$$

Mayer term
Assuming that all functions are sufficiently smooth, we can use these results to derive a similar solution for Eq. (1.34). Letting $L \coloneqq \phi$,

$$\begin{aligned}
&& g &= \frac{\mathrm{d}G}{\mathrm{d}T} && \text{(1.43a)}\\
\Rightarrow && \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}} g &= \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}} \frac{\mathrm{d}G}{\mathrm{d}T} = \frac{\mathrm{d}}{\mathrm{d}T} \frac{\mathrm{d}G}{\mathrm{d}\mathbf{p}} && \text{(1.43b)}\\
&& &\overset{(1.42)}{=} \frac{\mathrm{d}}{\mathrm{d}T} \left(\int_{t_0}^{T} L_\mathbf{p} + \boldsymbol{\lambda}^\top \mathbf{f}_\mathbf{p} \,\mathrm{d}t + \boldsymbol{\lambda}^\top(t_0)\mathbf{y}_\mathbf{p}(t_0) \right) && \text{(1.43c)}
\end{aligned}$$

Note that $\mathbf{y}(t)$ is independent of T, while $\boldsymbol{\lambda}(t)$ does depend on T through the initial value condition Eq. (1.41b). Equation (1.43c) then becomes

$$\begin{aligned}
\frac{\mathrm{d}g}{\mathrm{d}\mathbf{p}} &= L_\mathbf{p}(T) + \underbrace{\boldsymbol{\lambda}^\top(T)}_{=0} \mathbf{f}_\mathbf{p}(T) - \int_{t_0}^{T} \boldsymbol{\lambda}_T^\top \mathbf{f}_\mathbf{p} \,\mathrm{d}t - \boldsymbol{\lambda}_T^\top(t_0)\mathbf{y}_\mathbf{p}(t_0) && \text{(1.44a)}\\
&= L_\mathbf{p}(T) + \boldsymbol{\mu}^\top(t_0)\mathbf{y}_\mathbf{p}(t_0) + \int_{t_0}^{T} \boldsymbol{\mu}^\top \mathbf{f}_\mathbf{p} \,\mathrm{d}t && \text{(1.44b)}
\end{aligned}$$

with $\boldsymbol{\mu} = -\frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}T}$. Differentiating Eq. (1.41) w.r.t. T we find that $\boldsymbol{\mu}$ is the solution to the IVP

$$\dot{\boldsymbol{\mu}} = -\frac{\mathrm{d}\dot{\boldsymbol{\lambda}}}{\mathrm{d}T} = \mathbf{f}_{\mathbf{y}}^{\mathsf{T}}\boldsymbol{\lambda}_T \qquad = -\mathbf{f}_{\mathbf{y}}^{\mathsf{T}}\boldsymbol{\mu} \tag{1.45a}$$

$$\boldsymbol{\mu}(T) = -\frac{\mathrm{d}}{\mathrm{d}T}\boldsymbol{\lambda}(T) = -\dot{\boldsymbol{\lambda}}(T) = \mathbf{f}_{\mathbf{y}}^{\mathsf{T}}(T)\underbrace{\boldsymbol{\lambda}(T)}_{=0} + L_{\mathbf{y}}^{\mathsf{T}}(T) \qquad = L_{\mathbf{y}}^{\mathsf{T}}(T). \tag{1.45b}$$

Computational cost
This method to compute sensitivities has several important properties. During computation of the co-states all parameters are considered constant. In other words, the co-states do not depend on any information about the directions w.r.t. which we want to compute sensitivities. Once the trajectory of $\boldsymbol{\lambda}$ (or $\boldsymbol{\mu}$) is known, sensitivities of G (or g) w.r.t. arbitrary parameters can be computed simply by evaluating the one-dimensional integral in (1.42) (or (1.43c)) for each required directional derivative.

Computation of the co-states themselves only requires the solution of a linear IVP of dimension n_y. Therefore, the total computational cost of evaluating $\frac{\mathrm{d}G}{\mathrm{d}\mathbf{p}}$ for n_p parameters using adjoint sensitivity is roughly that of solving an ODE of dimension $2n_y + n_p$ instead of $n_y + n_y n_p$.

However, these benefits entail some difficulties. When the systems Eqs. (1.41) and (1.31) are combined, the problem is a boundary-value problem, which is considerably harder to solve than an IVP. For this reason the systems are usually solved successively: First, the original ODE is solved in a so-called *forward sweep*. Then, the adjoint states are computed in the *backward sweep* by solving the terminal value problem Eq. (1.41). However, at every time t during the backward sweep the evaluation of the RHS of Eq. (1.41a) requires the current value of the states $\mathbf{y}(t)$. Therefore, the full trajectory of $\mathbf{y}$ has to be available during integration of the adjoint system.

If the numerical integration methods and step sizes are chosen such that the RHS of the adjoint systems is only evaluated at times t for which estimates of $\mathbf{y}(t)$ were computed during the forward run, it is sufficient to store these discrete estimates. Memory consumption is linear in the number of states and the number of integration steps, which can quickly become infeasible for a long time horizon or small step sizes. Most advanced numerical codes for adjoint sensitivity use a *check-pointing scheme*, which allows computation of state value estimates at arbitrary points at the cost of only a few additional forward sweeps, while keeping memory consumption very low. The idea was first proposed by Griewank [53] and the algorithm *revolve* was published as *Algorithm 799* in ACM TOMS [54]. Recent improvements primarily address strategies to choose optimal checkpoint locations [27, 114]

If values at intermediate points are needed, a continuous approximation of $\mathbf{y}$ is required. The problem of computing continuous solutions as opposed to estimates on a discrete time grid arises in numerous other applications as well, for example for the detection of implicit switching or stopping conditions (see Section 1.2.2), and different solution methods can be found in the literature. Gladwell proposed to use Hermite polynomials to interpolate the state estimates over multiple steps [50]. After Horn introduced interpolation schemes requiring very little overhead by using the special structure of Runge-Kutta methods of orders 3, 4 and 5 [60, 61], similar formulas have been developed for higher order methods.

1.2.2 Implicit switches

In many applications the dynamics cannot be described by a model that is everywhere continuously differentiable. Discontinuities in an ODE model are frequently caused by switches between a discrete set of different modes of operation, triggered by one of the following:

Change of discrete control. A switch of this kind is caused by an influence that is controlled automatically or manually, for example:

- Gear shift in a car model
- Discrete on/off switch for a pump or valve in a chemical reactor

Explicitly modeled event. Event that is planned to occur over the simulation/optimization horizon. If the models that apply during each part of the time horizon (including their order) are known, we usually speak of different *problem phases* or *problem stages*. Note that the switching time can still be defined by an implicit condition depending on the system states.

- Takeoff/landing of an airplane
- Start of new sub-task in robot's workflow

Implicitly defined event. Event whose occurrence depends on the system state.

- Starting chemical reaction once threshold energy is reached
- Gear shift depending on engine speed

In this section we will outline how to deal with so-called *implicit switches*, that are defined by the sign structure of a continuously differentiable *switching function*

$$\psi : I \times R^{n_y} \to R^{n_\psi} \tag{1.46a}$$
$$(t, \mathbf{y}) \mapsto \psi(t, \mathbf{y}), \tag{1.46b}$$

indicating which set of model equations are active at the current time and state. Let

$$\sigma : I \times R^{n_y} \to \{-1, 0, 1\}^{n_\psi} \tag{1.47a}$$
$$(t, \mathbf{y}) \mapsto \operatorname{sgn} \psi(t, \mathbf{y}) \tag{1.47b}$$

be the *sign function*, yielding the sign structure of ψ. A *switching time* or *switching point* is a time $t^* \in I$ such that at least one component of $\psi(t^*, \mathbf{y}(t^*))$ is zero.

Omitting the dependence on model parameters for simplicity's sake, the model itself is given by

$$\dot{\mathbf{y}}(t) = \mathbf{f}_{\sigma(t, \mathbf{y}(t))}(t, \mathbf{y}(t)) \tag{1.48}$$

whenever all components of the switching function are nonzero, in other words on the open interval between two switching points.

Now let $t^* \in I$ such that ψ_i has a zero crossing and all other ψ_j are nonzero. Because ψ is continuously differentiable, there is only one switch in a small neighborhood of t^*. Behavior of the model at the switching times is described by a family of *jump functions* Δ_σ of the form

$$\Delta : I \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_y} \tag{1.49a}$$
$$(t, \mathbf{y}) \mapsto \Delta(t, \mathbf{y}). \tag{1.49b}$$

via

$$\mathbf{y}^+(t^*) = \Delta_{\sigma(t^*, \mathbf{y}^-(t^*))}(t^*, \mathbf{y}^-(t^*)) \tag{1.50}$$

where

$$\mathbf{y}^-(t^*) = \lim_{\substack{t \to t^* \\ t > t^*}} \mathbf{y}(t) \qquad \mathbf{y}^+(t^*) = \lim_{\substack{t \to t^* \\ t < t^*}} \mathbf{y}(t) \tag{1.51}$$

are the left and right limits of $\mathbf{y}$ at t^* respectively.

In this general setting various difficulties can arise. Most importantly, the switching conditions can be inconsistent due to the potential jump of $\mathbf{y}$. In other words, reversing time and approaching a switch from the right leads to a different solution than the original one, or a solution to the reversed problem does not exist at all. The switches occurring in the application presented in this work arise from changing bang-bang controls of an Optimal Control Problem (OCP) and therefore have a special structure. In particular, the states are continuous at switching times and switches are guaranteed to be consistent. Therefore, we will skip the details on how to deal with inconsistent switches and refer the reader to the thorough treatment by Filippov [43] and Brandt-Pollmann [17].

Since on each interval between two switches the model is just an initial value problem, numerical solution techniques are very similar to those presented above. However, because convergence theory for integration methods require the RHS to be smooth over a whole time step, the integrator has to be stopped at each switching time in order to ensure convergence. The main difficulty here is to reliably detect switches during each integration step and to efficiently compute good estimates of the switching times.

A common approach is to find roots of ψ based on a continuous approximation of $\mathbf{y}$ over the whole time step (see Section 1.2.1).

Sensitivity analysis

As shown by Bock [12], derivatives of $\mathbf{y}$ w.r.t parameters still exist in the presence of switches, if all switches are consistent and if the sign structure of the switching function does not change in a small neighborhood of the solution. This is the case if only one component of the switching function ψ has a root at each switching point and no switch occurs at t_0 or T.

Let $t^{(1)}, \dots, t^{(n)}$ be the switching times and

$$I^{(i)} = (t^{(i)}, t^{(i+1)}) \tag{1.52}$$

the corresponding partitioning of the time horizon, where $t^{(0)} = t_0$ and $t^{(n+1)} = T$. Denote by $\sigma^{(i)}$ the sign structure of ψ on the interval $I^{(i)}$. Let

$$\begin{aligned} \mathbf{f}^{(i)} &= \mathbf{f}_{\sigma^{(i)}} \\ \Delta^{(i)} &= \Delta_{\sigma(t^{(i)}, \mathbf{y}^-(t^{(i)}))} \end{aligned}$$

be the corresponding active RHS and the jump function at switching time $t^{(i)}$ respectively. The solution $\mathbf{y}(t)$ of Eq. (1.48) is given by $\mathbf{y}(t) = \mathbf{y}^{(i)}(t)$ for $t \in I^{(i)}$, where $\mathbf{y}^{(i)}(t)$ is the solution to the initial value problem

$$\dot{\mathbf{y}}^{(i)}(t) = \mathbf{f}^{(i)}(t, \mathbf{y}^{(i)}(t)) \tag{1.53a}$$

$$\mathbf{y}_0^{(i)} := \mathbf{y}^{(i)}(t^{(i)}) = \Delta^{(i)}(t^{(i)}, \mathbf{y}^{(i-1)}(t^{(i)})) \qquad \forall i \geq 1 \tag{1.53b}$$

$$\mathbf{y}_0^{(0)} := \mathbf{y}^{(0)}(t^{(0)}) = \mathbf{y}_0. \tag{1.53c}$$

Forward sensitivity analysis on each interval $I^{(i)}$ remains the same as in the case without switches (see Section 1.2.1). The only difference to the standard IVP case is how perturbations are transferred across switches. In view of the fact that the application presented in this thesis is a much simpler case, we will not go into further details on the general forward and adjoint sensitivity problems in the presence of implicit switches and how to solve them, but instead refer the reader to the PhD thesis of Brandt-Pollmann [17] for further information. Instead, we will only review the special case where the solution is continuous, i.e. $\Delta^{(i)}(t, \mathbf{y}, \mathbf{p}) = \mathbf{y}$ for every i.

It is sufficient to consider a problem on the interval $[0, 1]$ with only one switch, which extends straightforwardly to the general case. Let ψ be the (scalar) switching function and $t^* \in (0, 1)$ the switching time satisfying

$$\psi(t^*, \mathbf{y}(t^*; \mathbf{p}), \mathbf{p}) = 0, \tag{1.54}$$

where $\mathbf{y}(t; \mathbf{p})$ is the solution of

$$\dot{\mathbf{y}}(t) = \begin{cases} \mathbf{f}^-(t, \mathbf{y}(t), \mathbf{p}) & \text{if } \psi(t, \mathbf{y}(t), \mathbf{p}) < 0 \\ \mathbf{f}^+(t, \mathbf{y}(t), \mathbf{p}) & \text{else} \end{cases} \tag{1.55a}$$

$$\mathbf{y}(t_0) = \mathbf{y}_0. \tag{1.55b}$$

Assume that the regularity condition $\frac{\partial \psi}{\partial t}(t^*, \mathbf{y}(t^*; \mathbf{p}), \mathbf{p}) \neq 0$ is satisfied. Writing Eq. (1.55) in integral form we can compute $\mathbf{y}(1; \mathbf{p})$ as

$$\mathbf{y}(t^*; \mathbf{p}) = \mathbf{y}_0 + \int_0^{t^*} \mathbf{f}^-(t, \mathbf{y}(t), \mathbf{p}) \, \mathrm{d}t \tag{1.56a}$$

$$\mathbf{y}(1; \mathbf{p}) = \mathbf{y}(t^*; \mathbf{p}) + \int_{t^*}^{1} \mathbf{f}^+(t, \mathbf{y}(t), \mathbf{p}) \, \mathrm{d}t. \tag{1.56b}$$

When solving the forward sensitivity equation (Eq. (1.32)), the sensitivity matrix $\mathbf{y_p}$ has to be updated at the discontinuity. Let

$$\mathbf{y}_\mathbf{p}^-(t^*;\mathbf{p}) = \lim_{\substack{t\to t^*\\ t>t^*}} \mathbf{y_p}(t) \qquad \mathbf{y}_\mathbf{p}^+(t^*;\mathbf{p}) = \lim_{\substack{t\to t^*\\ t<t^*}} \mathbf{y_p}(t) \tag{1.57a}$$

denote the left and right limit of $\mathbf{y_p}$ at t^* respectively.

Taking the derivative of Eq. (1.56) w.r.t $\mathbf{p}$ yields

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{y}(t^*;\mathbf{p}) = \underbrace{\mathbf{y}_{0\mathbf{p}} + \int_0^{t^*} \left[\mathbf{f}_\mathbf{y}^-\mathbf{y_p} + \mathbf{f}_\mathbf{p}^-\right] \mathrm{d}t}_{=\mathbf{y}_\mathbf{p}^-(t^*;\mathbf{p})} + \mathbf{f}^-(t^*,\mathbf{y}(t^*,\mathbf{p}),\mathbf{p})\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}t^* \tag{1.58a}$$

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{y}(1;\mathbf{p}) = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{y}(t^*;\mathbf{p}) + \int_{t^*}^{1} \left[\mathbf{f}_\mathbf{y}^+\mathbf{y_p} + \mathbf{f}_\mathbf{p}^+\right] \mathrm{d}t - \mathbf{f}^+(t^*,\mathbf{y}(t^*,\mathbf{p}),\mathbf{p})\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}t^* \tag{1.58b}$$

$$= \underbrace{\mathbf{y}_\mathbf{p}^-(t^*;\mathbf{p}) + \int_{t^*}^{1} \left[\mathbf{f}_\mathbf{y}^+\mathbf{y_p} + \mathbf{f}_\mathbf{p}^+\right]}_{=\mathbf{y}_\mathbf{p}^+(t^*;\mathbf{p})}$$

$$+ \left[\mathbf{f}^-(t^*,\mathbf{y}(t^*,\mathbf{p}),\mathbf{p}) - \mathbf{f}^+(t^*,\mathbf{y}(t^*,\mathbf{p}),\mathbf{p})\right]\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}t^* \,\mathrm{d}t. \tag{1.58c}$$

Therefore the update at each switch is given by

$$\mathbf{y}_\mathbf{p}^+(t^*;\mathbf{p}) = \mathbf{y}_\mathbf{p}^-(t^*;\mathbf{p}) + \left[\mathbf{f}^-(t^*,\mathbf{y}(t^*,\mathbf{p}),\mathbf{p}) - \mathbf{f}^+(t^*,\mathbf{y}(t^*,\mathbf{p}),\mathbf{p})\right]\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}t^* \tag{1.59}$$

where $\frac{\mathrm{d}t^*}{\mathrm{d}\mathbf{p}}$ can be computed using Eq. (1.54) and the implicit function theorem. In particular, if $\mathbf{f}$ is continuous across the switching time, i.e. $\mathbf{f}^- = \mathbf{f}^+$, then $\mathbf{y}_\mathbf{p}^+$ is continuous as well.

1.3 Optimal control

An Optimal Control Problem (OCP) is an optimization problem, in which a dynamical system with states $\mathbf{y}$, governed by a set of differential equations, is controlled via time-dependent controls $\mathbf{u}$ and constant parameters $\mathbf{p}$. In this thesis we will only be concerned with the special case where the system dynamics can be described by an explicitly stated IVPs. We will therefore not cover differential algebraic equations (DAEs) or multistage problems. Moreover, we will not deal with interior point constraints, as they are not required by our application.

Definition 1.3.1 (Continuous ODE-constrained OCP) A *continuous ODE-constrained OCP* is an optimization problem

$$\begin{array}{lll} \max\limits_{\mathbf{y},\mathbf{u},\mathbf{p}} & \Phi[\mathbf{y},\mathbf{u},\mathbf{p}] & \\ \text{s.t.} & \dot{\mathbf{y}}(t) = \mathbf{f}(t,\mathbf{y}(t),\mathbf{u}(t),\mathbf{p}) & \forall t \in [t_0,T] \\ & \mathbf{y}(t_0) = \mathbf{y}_0 & \\ & 0 \leq \mathbf{g}(t,\mathbf{y}(t),\mathbf{u}(t),\mathbf{p}) & \forall t \in [t_0,T], \end{array} \tag{\textbf{OCP}}$$

where $t \in [t_0,T]$ is the time, $\mathbf{y}\colon[t_0,T] \to \mathbb{R}^{n_y}$ are the state variables, $\mathbf{u}\colon[t_0,T] \to \mathbb{R}^{n_u}$ are the control functions, and $\mathbf{p} \in \mathbb{R}^{n_p}$ is a vector of time-independent parameters. The RHS $\mathbf{f}\colon\mathbb{R} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_y}$ is piecewise Lipschitz-continuous in $\mathbf{y}$, and the inequality constraint function $\mathbf{g}\colon\mathbb{R}\times\mathbb{R}^{n_y}\times\mathbb{R}^{n_u}\times\mathbb{R}^{n_p} \to \mathbb{R}^{n_g}$ is twice differentiable. The objective is a Bolza functional $\Phi[\mathbf{y},\mathbf{u},\mathbf{p}] = \phi(\mathbf{y}(T),\mathbf{p}) + \int_{t_0}^{T} L(t,\mathbf{y}(t),\mathbf{u}(t),\mathbf{p})\,\mathrm{d}t$, with a Mayer term ϕ and a Lagrange term $\int_{t_0}^{T} L\,\mathrm{d}t$, both of which are assumed to be twice differentiable.

A *solution* of the OCP is a triple $(\mathbf{y},\mathbf{u},\mathbf{p})$ with absolutely continuous $\mathbf{y}$ and measurable $\mathbf{u}$, that satisfies the constraints of (**OCP**). A control function $\mathbf{u}$ is feasible, if there exists a solution $(\mathbf{y},\mathbf{u},\mathbf{p})$.

A solution $(\mathbf{y}^*, \mathbf{u}^*, \mathbf{p}^*)$ is *globally optimal*, if

$$\Phi[\mathbf{y}^*, \mathbf{u}^*, \mathbf{p}^*] \geq \Phi[\mathbf{y}, \mathbf{u}, \mathbf{p}] \tag{1.60}$$

for every solution $(\mathbf{y}, \mathbf{u}, \mathbf{p})$.

A solution $(\mathbf{y}^*, \mathbf{u}^*, \mathbf{p}^*)$ is *locally optimal*, if there exists a $\delta > 0$, such that Eq. (1.60) holds for every solution $(\mathbf{y}, \mathbf{u}, \mathbf{p})$ that satisfies

$$\|\mathbf{u}^*(t) - \mathbf{u}(t)\| \leq \delta \qquad \forall t \in [t_0, T] \tag{1.61}$$

$$\|\mathbf{p}^* - \mathbf{p}\| \leq \delta. \tag{1.62}$$

1.3.1 Maximum principle

In order to simplify notation, we will state the optimality conditions for an optimal control problem of the form

$$\begin{array}{llll} \max\limits_{\mathbf{y},\mathbf{u}} & \Phi[\mathbf{y}, \mathbf{u}] & & \\ \text{s.t.} & \dot{\mathbf{y}}(t) & = & \mathbf{f}(\mathbf{y}(t), \mathbf{u}(t)) \quad \forall t \in [t_0, T] \\ & \mathbf{y}(t_0) & = & \mathbf{y}_0 \\ & 0 & \leq & \mathbf{g}(\mathbf{y}(t), \mathbf{u}(t)) \quad \forall t \in [t_0, T], \end{array} \tag{\textbf{OCP'}}$$

where the RHS of the ODE and the constraints do not depend explicitly on the time t or on constant parameters $\mathbf{p}$. This is w.l.o.g., because both can be included in the vector of differential states.

Similarly to the role of the Lagrangian in nonlinear programming, optimality conditions for optimal control problems use the idea of the Hamiltonian.

Definition 1.3.2 (Hamiltonian) The *Hamiltonian function* or simply *Hamiltonian* of the optimal control problem (**OCP'**) is the function

$$\mathcal{H}: \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_g} \to \mathbb{R} \tag{1.63a}$$

$$(\mathbf{y}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mapsto \mathcal{H}(\mathbf{y}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \tag{1.63b}$$

where

$$\mathcal{H}(\mathbf{y}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := L(\mathbf{y}, \mathbf{u}) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{y}, \mathbf{u}) + \boldsymbol{\mu}^\top \mathbf{g}(\mathbf{y}, \mathbf{u}). \tag{1.63c}$$

The following set of necessary conditions for an optimal solution of (**OCP'**), the so-called *maximum principle* or *minimum principle*, is mainly due to Pontryagin [100]. It is the central result in optimal control theory [58] and can be found in any standard textbook on the subject [49, 11].

Theorem 1.3.3
*Let $(\mathbf{y}^*, \mathbf{u}^*)$ be an optimal solution to (**OCP'**). Then there exist* Lagrange multipliers $\boldsymbol{\lambda}^*: \mathbb{R}^{n_y} \to \mathbb{R}$ *and $\boldsymbol{\mu}^*: \mathbb{R}^{n_g} \to \mathbb{R}$, such that for almost all $t \in [t_0, T]$ the quadruple $(\mathbf{y}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ satisfies*

- *the* stationarity *condition*

$$\mathbf{u}^*(t) = \arg\max_{\mathbf{u}} \mathcal{H}(\mathbf{y}^*(t), \mathbf{u}, \boldsymbol{\lambda}^*(t), \boldsymbol{\mu}^*(t)), \tag{1.64a}$$

- *the* primal feasibility *conditions*

$$\dot{\mathbf{y}}^*(t) = \mathcal{H}_{\boldsymbol{\lambda}}^\top(\mathbf{y}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{\mu}^*(t)) = \mathbf{f}(\mathbf{y}^*(t), \mathbf{u}^*(t)), \tag{1.64b}$$

$$\mathbf{y}^*(t_0) = \mathbf{y}_0, \tag{1.64c}$$

$$0 \leq \mathcal{H}_{\boldsymbol{\mu}}^\top(\mathbf{y}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{\mu}^*(t)) = \mathbf{g}(\mathbf{y}^*(t), \mathbf{u}^*(t)), \tag{1.64d}$$

- *the* dual feasibility *conditions*

$$\dot{\boldsymbol{\lambda}}^*(t) = -\mathcal{H}_{\mathbf{y}}^\top(\mathbf{y}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{\mu}^*(t)), \tag{1.64e}$$

$$\boldsymbol{\lambda}^*(T) = \phi_{\mathbf{y}}^\top(\mathbf{y}^*(T)), \tag{1.64f}$$

$$0 \leq \boldsymbol{\mu}^*(t), \tag{1.64g}$$

- *and the* complementary slackness *condition*

$$0 = \boldsymbol{\mu}^{*\top}(t)\mathbf{g}(\mathbf{y}^*(t), \mathbf{u}^*(t)). \tag{1.64h}$$

The point-wise maximum of the Hamiltonian in Eq. (1.64a) *is taken over all feasible controls, i.e. all* $\mathbf{u}$ *such that the inequality constraints Eq.* (1.64d) *are satisfied.*

Proof See [100] or [21]. □

Remark 1.3.4 Again, together with Eqs. (1.64d) and (1.64g) it is easy to see that the complementary slackness condition Eq. (1.64h) holds element-wise.

Applying the KKT-conditions to the point-wise maximization of the Hamiltonian, we obtain for almost every $t \in [t_0, T]$ the first order necessary conditions

$$\begin{aligned} 0 &= \mathcal{H}_{\mathbf{u}}(\mathbf{y}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{\mu}^*(t)) && (1.65\text{a}) \\ &= L_{\mathbf{u}}(\mathbf{y}^*(t), \mathbf{u}^*(t)) + \boldsymbol{\lambda}^{*\top}\mathbf{f}_{\mathbf{u}}(\mathbf{y}^*(t), \mathbf{u}^*(t)) + \boldsymbol{\mu}^{*\top}\mathbf{g}_{\mathbf{u}}(\mathbf{y}^*(t), \mathbf{u}^*(t)) && (1.65\text{b}) \end{aligned}$$

and the second order necessary Legendre-Clebsh condition, stating that the Hessian of the Hamiltonian

$$\frac{\partial^2 \mathcal{H}}{\partial \mathbf{u}^2}(\mathbf{y}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{\mu}^*(t)) \tag{1.66}$$

is positive-semidefinite on the kernel of the active constraints. Second order sufficient conditions for a local optimum can for example be found in Maurer and Osmolovskii [88].

There are several structurally different classes of local optima, which we will describe in the following. To this end, let

$$\psi(\mathbf{y}, \mathbf{u}, \boldsymbol{\lambda}) := L_{\mathbf{u}}(\mathbf{y}, \mathbf{u}) + \boldsymbol{\lambda}^\top \mathbf{f}_{\mathbf{u}}(\mathbf{y}, \mathbf{u}). \tag{1.67}$$

be the *switching function*.

Remark 1.3.5 (Solution structure) For every $1 \le j \le n_u$ and almost every time t, the j-th control $\mathbf{u}_j^*$ falls in one of two classes, depending on the behavior of the respective component of the switching function in a neighborhood of t:

Constraint seeking control: If $\psi_j(\mathbf{y}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*) \neq 0$, then with Eq. (1.65) $\boldsymbol{\mu}^{*\top}\mathbf{g}_{\mathbf{u}}(\mathbf{y}^*, \mathbf{u}^*) \neq 0$ as well. Therefore, there exists an inequality constraint $\mathbf{g}_k$, such that both $\frac{\partial \mathbf{g}_k}{\partial \mathbf{u}_j} \neq 0$ and $\boldsymbol{\mu}_k^* \neq 0$. The latter implies, together with Eq. (1.64h), that the constraint $\mathbf{g}_k \le 0$ is active. Locally, the optimal control $\mathbf{u}_j^*$ is then uniquely defined by the implicit condition $\mathbf{g}_k = 0$. Controls of this type are called *constraint seeking.*

Sensitivity seeking control: If $\psi_j = 0$ and ψ_j depends explicitly on $\mathbf{u}_j$, the control $\mathbf{u}_j^*$ is implicitly defined by $\psi_j = 0$ and is called *sensitivity seeking.*

If $\psi_j = 0$ and ψ_j does not depend explicitly on $\mathbf{u}_j$, we use the fact that $\mathcal{H}_{\mathbf{u}} \equiv 0$ on an open neighborhood of t and therefore all time derivatives $\frac{\mathrm{d}^i \mathcal{H}_{\mathbf{u}}}{\mathrm{d}t^i} = 0$ as well. Differentiating Eq. (1.65) often enough and using Eqs. (1.64b) and (1.64e), with a similar analysis as above one again ends up in one of the two cases: Either $\mathbf{u}_j^*$ is determined by the system dynamics (sensitivity-seeking) or by an active constraint (constraint-seeking).

For the problems occurring in this thesis, we are most interested in the following special case of a *control affine system* with only *simple bounds* as inequality constraints.

Remark 1.3.6 (Control affine system) Consider the OCP

$$\begin{array}{lllll} \max\limits_{\mathbf{y},\mathbf{u}} & \phi(\mathbf{y}(T)) + \int_{t_0}^T L^0(\mathbf{y}(t)) + L^u(\mathbf{y}(t))\,\mathbf{u}(t)\,\mathrm{d}t & & & \\ \text{s.t.} & \dot{\mathbf{y}}(t) & = & \mathbf{f}(\mathbf{y}(t), \mathbf{u}(t)) = \mathbf{f}^0(\mathbf{y}(t)) + \mathbf{f}^u(\mathbf{y}(t))\mathbf{u}(t) & \forall t \in [t_0, T] \\ & \mathbf{y}(t_0) & = & \mathbf{y}_0 & \\ & 0 & \le & \mathbf{u}(t) - \mathbf{u}^{\min} & \forall t \in [t_0, T], \\ & 0 & \le & \mathbf{u}^{\max} - \mathbf{u}(t) & \forall t \in [t_0, T], \end{array} \tag{1.68}$$

where the Lagrange term in the objective functional as well as the RHS are affine functions of $\mathbf{u}$ with state-dependent coefficients $L^0: \mathbb{R}^{n_y} \to \mathbb{R}$, $L^u: \mathbb{R}^{n_y} \to \mathbb{R}^{1\times n_u}$, $\mathbf{f}^0: \mathbb{R}^{n_y} \to \mathbb{R}^{n_y}$, $\mathbf{f}^u: \mathbb{R}^{n_y} \to \mathbb{R}^{n_y \times n_u}$. Moreover, the inequality constraints are simple bounds on the controls and, in particular, independent of the state vector $\mathbf{y}$.

The optimal control $\mathbf{u}^*(t)$ at time t is determined by locally maximizing the Hamiltonian subject to the inequality constraints, i.e.

$$\mathbf{u}^* = \arg\max_{\mathbf{u}} \left\{ \mathcal{H} = L^0 + L^u \mathbf{u} + \boldsymbol{\lambda}^{*\top} \left[\mathbf{f}^0 + \mathbf{f}^u \mathbf{u} \right] \;\middle|\; \mathbf{u}^{\min} \leq \mathbf{u} \leq \mathbf{u}^{\max} \right\} \tag{1.69a}$$

$$= \arg\max_{\mathbf{u}} \{ \underbrace{\left[L^u + \boldsymbol{\lambda}^{*\top} \mathbf{f}^u \right]}_{=\psi(\mathbf{y})} \mathbf{u} \;\mid\; \mathbf{u}^{\min} \leq \mathbf{u} \leq \mathbf{u}^{\max} \}. \tag{1.69b}$$

We again distinguish two cases:

Non-singular control ⇒ constraint-seeking control: If the j-th component of the switching function ψ is non-zero, the respective control variable $\mathbf{u}_j$ is called *non-singular*. The optimal choice for $\mathbf{u}_j$ is then uniquely determined by the sign of ψ_j: $\psi_j > 0 \Rightarrow \mathbf{u}_j^* = \mathbf{u}_j^{\max}$, and $\psi_j < 0 \Rightarrow \mathbf{u}_j^* = \mathbf{u}_j^{\min}$. Therefore, non-singular controls are always constraint seeking. Controls that are determined by active simple bound constraints are called *bang-bang controls*.

Singular control ⇒ sensitivity-seeking control: If $\psi_j \equiv 0$ on an interval, the condition $\mathcal{H}_{\mathbf{u}} = 0$ is never sufficient to determine $\mathbf{u}_j^*$, because ψ is independent of $\mathbf{u}$. Instead, we again differentiate $\mathcal{H}_{\mathbf{u}}$ w.r.t to time. Since the inequality constraints are state independent, the time derivatives of $\mathcal{H}_{\mathbf{u}}$ are independent of $\mathbf{g}$ and the optimal control is a always sensitivity-seeking one.

Remark 1.3.7 (Extensions to more general problems) The maximum principle as stated above can be extended in several ways. We will briefly state those relevant for the problems occurring in this thesis and refer the reader to the standard optimal control literature for everything else.

Dependence on parameters and explicit dependence on time. If the problem has the more general form of (**OCP**), i.e. the RHS $\mathbf{f}$ depends on a vector of parameters $\mathbf{p} \in \mathbb{R}^{n_p}$ and the time t, we can include both in the vector of differential states $\mathbf{y}$: Let $\mathbf{y}$ be a solution of the n_y-dimensional ODE

$$\mathbf{y}(t_0) = \mathbf{y}_0 \tag{1.70a}$$

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t), \mathbf{u}(t), \mathbf{p}). \tag{1.70b}$$

Then

$$\mathbf{y}' := \begin{pmatrix} t \\ \mathbf{p} \\ \mathbf{y} \end{pmatrix} \tag{1.70c}$$

is a solution to the $(1 + n_p + n_y)$-dimensional ODE

$$\mathbf{y}'(t_0) = \begin{pmatrix} t_0 \\ \mathbf{p} \\ \mathbf{y}_0 \end{pmatrix} \tag{1.71a}$$

$$\dot{\mathbf{y}}'(t) = \mathbf{f}'(\mathbf{y}'(t), \mathbf{u}(t)) := \begin{pmatrix} 1 \\ 0 \\ \mathbf{f}(t, \mathbf{y}(t), \mathbf{u}(t), \mathbf{p}) \end{pmatrix}. \tag{1.71b}$$

Applying the maximum principle to an OCP with a dynamical system of the form Eq. (1.71) we obtain a vector of Lagrange multipliers

$$\boldsymbol{\lambda}' = \begin{pmatrix} \boldsymbol{\lambda}^t \\ \boldsymbol{\lambda}^{\mathbf{p}} \\ \boldsymbol{\lambda} \end{pmatrix}. \tag{1.72a}$$

With the fact that

$$\frac{\partial \mathbf{f}'}{\partial \mathbf{y}'} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\partial \mathbf{f}}{\partial t} & \frac{\partial \mathbf{f}}{\partial \mathbf{p}} & \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \end{pmatrix}, \tag{1.73}$$

it is easy to see that the ODE governing the dynamics of the adjoint variables $\boldsymbol{\lambda}$ associated to the original states $\mathbf{y}$ remains unchanged, i.e. Eq. (1.64e) still holds. Furthermore, because

$$\frac{\partial \mathbf{f}'}{\partial \mathbf{u}} = \begin{pmatrix} 0 \\ 0 \\ \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \end{pmatrix}, \tag{1.74}$$

and therefore $\boldsymbol{\lambda}^\top \mathbf{f}'(\mathbf{y}'(t), \mathbf{u}(t)) = \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{y}(t), \mathbf{u}(t)) + C$, with C constant w.r.t. $\mathbf{u}(t)$, the optimality condition Eq. (1.64a) stays the same.

Terminal value constraints. If additional equality constraints of the form

$$\mathbf{r}\left(\mathbf{y}(T)\right) = 0 \tag{1.75}$$

are imposed on the states at the terminal time T with a twice-differentiable function $\mathbf{r}: \mathbb{R}^{n_y} \to \mathbb{R}^{n_r}$, we introduce additional (constant) Lagrange multipliers $\boldsymbol{\nu} \in \mathbb{R}^{n_r}$ associated to the above constraint and form the *end point Lagrangian*

$$\psi(\mathbf{y}, \boldsymbol{\nu}) := \phi(\mathbf{y}) + \boldsymbol{\nu}^\top \mathbf{r}(\mathbf{y}). \tag{1.76}$$

The optimality condition Eq. (1.64f) is then replaced by the condition

$$\boldsymbol{\lambda}^*(T) = \psi_{\mathbf{y}}^\top(\mathbf{y}^*(T), \boldsymbol{\nu}^*). \tag{1.77}$$

1.3.2 Sensitivity analysis

Under suitable smoothness and regularity assumptions, a sensitivity analysis similar to the results for parametric NLPs presented in Section 1.1.2 can be carried out for the parametric problem (**OCP**). In the following, we will briefly state some results that are of concern for the problems studied in this thesis. Proofs, generalizations and extensive lists of references can for example be found in the work of Maurer and Pesch [89], Malanowski et al. [82], and Buskens and Maurer [22]. Let again (**OCP**($\mathbf{p}$))denote the problem instance corresponding to a given parameter vector $\mathbf{p} \in \mathbb{R}^{n_p}$.

In the following we will assume that all solutions are non-degenerate in the sense that the time horizon $[t_0, T]$ can be partitioned into finitely many intervals, such that the solution structure remains constant on each one. More precisely:

Assumption 1 There exist $t_0 = t^{(0)} < t^{(1)} < \cdots < t^{(n)} = T$, such for each time interval $I^{(i)} = (t^{(i)}, t^{(i+1)})$

- the set of active constraints remains constant on $I^{(i)}$,
- the behavior of every control $\mathbf{u}_j^*$ is constant on $I^{(i)}$, i.e. it is either sensitivity seeking for every $t \in I^{(i)}$ or constraint seeking and determined by the same active constraint for every $t \in I^{(i)}$.

Theorem 1.3.8 (Solution differentiability of OCPs)
*Let $\mathbf{p}_0 \in \mathbb{R}^{n_p}$ be a fixed parameter vector and let $\mathbf{z}_0^* = (\mathbf{y}_0^*, \mathbf{u}_0^*, \boldsymbol{\lambda}_0^*, \boldsymbol{\mu}_0^*)$ be a locally optimal solution to (**OCP**($\mathbf{p}_0$)) satisfying the optimality conditions Eq. (1.64). Under certain regularity conditions, including Assumption 1, $\mathbf{z}_0^*$ can be embedded in a parametric family of functions*

$$\mathbf{z}^*: [t_0, T] \times U \to \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_g} \tag{1.78a}$$

$$(t, \mathbf{p}) \mapsto \mathbf{z}^*(t, \mathbf{p}) = (\mathbf{y}^*(t, \mathbf{p}), \mathbf{u}^*(t, \mathbf{p}), \boldsymbol{\lambda}^*(t, \mathbf{p}), \boldsymbol{\mu}^*(t, \mathbf{p})) \tag{1.78b}$$

defined on a neighborhood $U \subset \mathbb{R}^{n_p}$ of $\mathbf{p}_0$, such that *(a)* $\mathbf{z}^*(\cdot, \mathbf{p}_0) = \mathbf{z}_0^*$,

(b) $\mathbf{z}^*(\cdot, \mathbf{p})$ *satisfies the optimality conditions Eq.* (1.64).

(c) $\mathbf{z}^*(\cdot, \mathbf{p})$ *is a local minimum of (**OCP**($\mathbf{p}$))for every* $\mathbf{p} \in U$.

Different proofs have been given depending on the structure of the solution. Maurer and Pesch [89, 90] as well as Malanowski and Maurer [82] consider the case where the switching function depends explicitly on $\mathbf{u}$, while Maurer et al. [87] and Maurer and Vossen [91] deal with the case where controls enter linearly.

Generally, the proofs work similarly to the sensitivity analysis for NLPs (see Section 1.1.2): A set of necessary and sufficient conditions for optimality as well as regularity conditions are formulated, which, together with the original set of equality and active inequality constraints, form a boundary value problem (BVP). Then, a sensitivity analysis of this BVP is carried out for a nominal solution $\mathbf{z}_0^*$ for (**OCP**($\mathbf{p}_0$)). Derivatives of $\mathbf{y}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*$, and $\boldsymbol{\mu}^*$ at $\mathbf{p} = \mathbf{p}_0$ can be computed using the implicit function theorem.

We will omit the details, because for the applications in this thesis we will only need the following result.

Definition 1.3.9 (Locally optimal value function) Let $\mathbf{z}^*$ be a family of locally optimal solutions of (**OCP**($\mathbf{p}$)) that satisfies (a) to (c) in Theorem 1.3.8. We again call the function

$$\begin{aligned} F: U \subset \mathbb{R}^{n_p} &\to \mathbb{R} \\ \mathbf{p} &\mapsto F(\mathbf{p}) = \Phi[\mathbf{y}(\cdot, \mathbf{p}), \mathbf{u}(\cdot, \mathbf{p}), \mathbf{p}], \end{aligned}$$

mapping a value of $\mathbf{p}$ to the objective function value of the corresponding local optimum for (**OCP**($\mathbf{p}$)), the *optimal value function.*

In the following, a subscript again denotes a partial derivative. Furthermore, we will omit arguments whenever there is no ambiguity. Unless stated otherwise, all partial derivatives are evaluated at the nominal parameter $\mathbf{p} = \mathbf{p}_0$ and the nominal solution $\mathbf{y}_0^*, \mathbf{u}_0^*, \boldsymbol{\lambda}_0^*, \boldsymbol{\mu}_0^*$ respectively. For example, $\phi_{\mathbf{p}} = \frac{\partial \phi}{\partial \mathbf{p}}(\mathbf{y}_0^*(T), \mathbf{p}_0)$, etc.

Theorem 1.3.10 (Sensitivity analysis for OCPs)
Let $\mathbf{p}_0$*,* $\mathbf{z}_0^*$ *and* $\mathbf{z}^*$ *be as in Theorem 1.3.8 and let* F *be the optimal value function. Then, under suitable regularity and smoothness assumptions,* F *is continuously differentiable w.r.t.* $\mathbf{p}$ *almost everywhere. Where it exits, its gradient* $F_{\mathbf{p}} = \frac{\partial}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_0} F$ *is given by*

$$F_{\mathbf{p}} = \phi_{\mathbf{p}} + \int_{t_0}^{T} \mathcal{H}_{\mathbf{p}} \, \mathrm{d}t. \tag{1.79}$$

Proof In order to simplify notation, let

$$\begin{aligned} L^*(\mathbf{p}, t) &:= L\left(t, \mathbf{y}^*(\mathbf{p}, t), \mathbf{u}^*(\mathbf{p}, t), \mathbf{p}\right), && (1.80a) \\ \mathbf{f}^*(\mathbf{p}, t) &:= \mathbf{f}\left(t, \mathbf{y}^*(\mathbf{p}, t), \mathbf{u}^*(\mathbf{p}, t), \mathbf{p}\right), && (1.80b) \\ \mathbf{g}^*(\mathbf{p}, t) &:= \mathbf{g}\left(t, \mathbf{y}^*(\mathbf{p}, t), \mathbf{u}^*(\mathbf{p}, t), \mathbf{p}\right), && (1.80c) \\ \mathcal{H}^*(\mathbf{p}, t) &:= \mathcal{H}\left(t, \mathbf{y}^*(\mathbf{p}, t), \mathbf{u}^*(\mathbf{p}, t), \boldsymbol{\lambda}^*(\mathbf{p}, t), \boldsymbol{\mu}^*(\mathbf{p}, t), \mathbf{p}\right) && (1.80d) \end{aligned}$$

for every $\mathbf{p} \in U$ and every t. By virtue of the optimality conditions Eqs. (1.64b) and (1.64h) we have

$$\begin{aligned} F(\mathbf{p}) &= \phi(\mathbf{y}^*(\mathbf{p}, T), \mathbf{p}) + \int_{t_0}^{T} L^*(\mathbf{p}, t) \, \mathrm{d}t && (1.81a) \\ &= \phi + \int_{t_0}^{T} L^* + \boldsymbol{\lambda}^{*\top}(\mathbf{f}^* - \dot{\mathbf{y}}^*) + \boldsymbol{\mu}^{*\top}\mathbf{g}^* \, \mathrm{d}t && (1.81b) \\ &= \phi + \int_{t_0}^{T} \mathcal{H}^* - \boldsymbol{\lambda}^{*\top}\dot{\mathbf{y}}^* \, \mathrm{d}t. && (1.81c) \end{aligned}$$

Taking the total derivative w.r.t. $\mathbf{p}$, we obtain

$$
\begin{aligned}
F_{\mathbf{p}} =& \underbrace{\phi_{\mathbf{y}}(\mathbf{y}^*(\mathbf{p},T),\mathbf{p})}_{\boldsymbol{\lambda}^{*\top}(\mathbf{p},T)}\mathbf{y}^*_{\mathbf{p}}(\mathbf{p},T) + \phi_{\mathbf{p}}(\mathbf{y}^*(\mathbf{p},T),\mathbf{p}) \\
& + \int_{t_0}^{T} \mathcal{H}_{\mathbf{p}} + \underbrace{\mathcal{H}_{\mathbf{y}}}_{-\dot{\boldsymbol{\lambda}}^{*\top}}\mathbf{y}^*_{\mathbf{p}} + \underbrace{\mathcal{H}_{\mathbf{u}}}_{0}\mathbf{u}^*_{\mathbf{p}} + \underbrace{\mathcal{H}_{\boldsymbol{\lambda}}}_{\mathbf{f}^{*\top}}\boldsymbol{\lambda}^*_{\mathbf{p}} + \underbrace{\mathcal{H}_{\boldsymbol{\mu}}}_{\mathbf{g}^{*\top}}\boldsymbol{\mu}^*_{\mathbf{p}} \\
& - \dot{\mathbf{y}}^{*\top}\boldsymbol{\lambda}^*_{\mathbf{p}} - \boldsymbol{\lambda}^{*\top}\dot{\mathbf{y}}^*_{\mathbf{p}}\,\mathrm{d}t && (1.82a) \\
=& \boldsymbol{\lambda}^{*\top}(T)\mathbf{y}^*_{\mathbf{p}}(T) + \phi_{\mathbf{p}} + \int_{t_0}^{T}\mathcal{H}_{\mathbf{p}}\,\mathrm{d}t + \int_{t_0}^{T}\underbrace{\mathbf{g}^{*\top}\boldsymbol{\mu}^*_{\mathbf{p}}}_{0}\,\mathrm{d}t \\
& - \int_{t_0}^{T}\left[\dot{\boldsymbol{\lambda}}^{*\top}\mathbf{y}^*_{\mathbf{p}} + \boldsymbol{\lambda}^{*\top}\dot{\mathbf{y}}^*_{\mathbf{p}}\right]\mathrm{d}t + \int_{t_0}^{T}\underbrace{\left[\mathbf{f}^{*\top} - \dot{\mathbf{y}}^{*\top}\right]}_{0}\boldsymbol{\lambda}^*_{\mathbf{p}}\,\mathrm{d}t && (1.82b) \\
=& \phi_{\mathbf{p}} + \int_{t_0}^{T}\mathcal{H}_{\mathbf{p}}\,\mathrm{d}t + \boldsymbol{\lambda}^{*\top}(T)\mathbf{y}^*_{\mathbf{p}}(T) - \left[\boldsymbol{\lambda}^{*\top}\mathbf{y}^*_{\mathbf{p}}\right]_{t_0}^{T} && (1.82c) \\
=& \phi_{\mathbf{p}} + \int_{t_0}^{T}\mathcal{H}_{\mathbf{p}}\,\mathrm{d}t + \boldsymbol{\lambda}^{*\top}(t_0)\underbrace{\frac{\partial \mathbf{y}_0}{\partial \mathbf{p}}}_{0} && (1.82d) \\
=& \phi_{\mathbf{p}} + \int_{t_0}^{T}\mathcal{H}_{\mathbf{p}}\,\mathrm{d}t && (1.82e)
\end{aligned}
$$

In the above, the braces in the first equation hold almost everywhere due to Eqs. (1.64f), (1.64e), (1.65), (1.64b) and (1.64d) respectively.

To see that the integrand in the second integral in Eq. (1.82b) is zero almost everywhere, we use the fact that for almost every time t the set of active constraints will not change for a small perturbation of $\mathbf{p}$. Therefore, for every $1 \le k \le n_g$, either the k-th constraint is active at $(\mathbf{p},t)$, i.e. $\mathbf{g}^*_k = 0$, or it is inactive in a neighborhood of $(\mathbf{p},t)$, which implies that locally $\boldsymbol{\mu}^* \equiv 0$ (see Eq. (1.64h)) and therefore $\frac{\partial}{\partial \mathbf{p}}\boldsymbol{\mu}^*_k = 0$.

The proof is completed by applying partial integration on the third integral in Eq. (1.82b), simplifying, and using the fact that in our setup the initial value is independent of the parameter vector. □

Chapter 2

Customer choice

The classic forecasting techniques that have been used in RM since the 1970's attempt to directly estimate booking probabilities or booking rates for a given set of fixed products. This is mostly done using time–series methods like exponential smoothing or moving averages directly on the observed bookings. These simple models are not suited for our application, because they assume that the set of products is fixed. In particular, the dependence of the booking probability on the products' attributes is not explicitly included in such a model. As a result, no information about the demand response to a change of features can be obtained.

In the last decade some progress has been made, primarily focusing on the impact of prices on demand. However, the main goal was to be able to deal with so-called *fare families*, where all products contained in one *family* of products have the same product attributes and only differ in their price. Demand is often assumed to be independent between families. From the customer's point of view the individual booking-classes within one fare family are just a number of different price points for the corresponding product. Since again prices are fixed, the standard models do not focus on the demand response to a change in prices, but rather attempt to estimate the share of customers that are willing to buy the product at each of the fixed price points.

A rational customer will not simply choose which fare family they wish to purchase and then purchase the cheapest available product with the corresponding set of attributes. They will rather consider the whole set of available alternatives and make a decision which option to purchase or to purchase nothing at all, balancing the difference in product restrictions (e.g. booking flexibility, cancellation fees, free checked bag, etc.) as well as in price differences against each other.

In this chapter we deal with the problem of modeling such a booking decision made by an individual customer, depending on the set of relevant products and their respective product attributes. Our main tool is *discrete choice* theory, which is a general framework to model and predict choices made from a finite set of alternatives.

The first steps towards discrete choice analysis were made in the field of mathematical psychology, starting in 1927 [118]. For an overview of different applications of discrete choice theory, the reader is referred to Manski and McFadden [84] and the more recent review paper by McFadden [93]. In the airline context, discrete choice theory has been extensively used in demand modeling mainly for RM purposes [115, 116, 25], but some other applications as well, for example in network planning [51]. In our description we will use the definitions and terminology from Ben-Akiva and Lerman [7], which is the standard reference for discrete choice modeling.

In the first section of this chapter we review the fundamentals of discrete choice theory and introduce a generic class of customer choice models, which directly models the thought process that goes into an individual's decision. Because the model is based on choices of individual decision makers, it can can readily be used in a simulation environment by drawing samples of individuals and evaluating the optimal choice for each customer. From an optimization perspective however, one is often interested in an aggregate outcome, such as the probability that a random customer chooses a certain product. In Section 2.2 we describe the so-called aggregation problem, uses a model for individual customer choice to derive an aggregate demand model, yielding booking probabilities for each product depending on a set of alternatives and their respective attributes. Depending on the specific choice model, closed forms for aggregate demand can sometimes be derived analytically. However, in general one needs to employ analytical approximations or numerical tools in order to estimate booking probabilities. We conclude the chapter by reviewing some widely used customer choice models in Section 2.3.

2.1 Discrete choice modeling

The goal of discrete choice modeling is to describe the behavior of an individual or group when confronted with a discrete set of alternatives. The resulting choice model should enable the researcher to achieve the following goals:

Estimation In order to apply a model in practice, one has to be able to estimate the parameters that specify the model from historical data. The most common approaches are maximum likelihood estimation and least-squares-fitting of model predictions to observations. As the structure of the estimation problem strongly depends on the specific model that is used, we will not discuss estimation in the general setting.

Simulation Often different optimization methods, control mechanisms, or forecasting techniques are benchmarked against each other by means of simulation. In order to conduct such a computational experiment, we have to be able to simulate a series of decisions on the level of single customers. Simulation can usually be reduced to the problem of drawing random samples from the given joint probability distributions that are used to describe the population of decision makers. We will therefore not go into details about simulation.

Aggregation Almost all optimization methods used in practice require a forecast of aggregate demand. In general this is a model for the distribution of the possible outcomes, e.g. the distribution of the number of customers choosing a specific product in a given time frame. However, in many cases it is sufficient to know the expected value of the quantity in question, e.g. as the expected number of bookings for a certain product. The aggregation problem is one of the most complex problems and one of the limiting factors when dealing with intricate choice models. We will discuss general aggregation methods in Section 2.2.

A discrete choice model consists of three components, the *decision maker*, the discrete *set of alternatives* and the *decision rule*, which will be covered in detail in the following.

Customer behavior strongly depends on the market, culture, economical climate and many other factors. Therefore, a vast number of different choice models for various applications can be found in the literature. Because of the very general nature of the problem there are very few obvious restrictions one can impose on potential customer models in order to discriminate between them. However, there is one central property that we expect a sensible choice model to have in any case:

Definition 2.1.1 (Transitivity) A choice model is transitive, if and only if the following is satisfied: Let $k \in \mathcal{S}$ be the choice made by a fixed customer when offered the set of alternatives $\mathcal{S} = \{1, \dots, M\}$. Then this customer will still choose k when given a subset of alternatives $\mathcal{S}' \subseteq \mathcal{S}$ with $k \in \mathcal{S}'$.

Equivalently, given that a customer chooses k out of the choice set $\mathcal{S}$, then given an extended choice set $\mathcal{S}' \supseteq \mathcal{S}$ the customer will either still choose k or one of the added products $\mathcal{S}' \setminus \mathcal{S}$.

Remark 2.1.2 If the attributes describing the elements of the set of alternatives $\mathcal{S}$ are fixed, Definition 2.1.1 is equivalent to the following: Each customer has a personal order of preference $k_1 \succ k_2 \succ \dots \succ k_M$. Given the set of choices $\mathcal{S}' \subseteq \mathcal{S}$, the customer chooses $\max(\mathcal{S}')$ where the maximum is taken with respect to $\succ$.

The ordering can be constructed easily by setting

$$\mathcal{S}_1 = \mathcal{S} \tag{2.1a}$$

$$k = \max(\mathcal{S}_k) \qquad \forall k = 1, \dots, M \tag{2.1b}$$

$$\mathcal{S}_k = \mathcal{S} \setminus \{k-1\} \qquad \forall k = 2, \dots, M, \tag{2.1c}$$

where again the maximum is taken with respect to the ordering $\succ$.

On the other hand certain properties clearly indicate that a choice model is too simple and thus not well-suited to capture a sophisticated decision making process. The most prominent example is the IIA property, which greatly simplifies the parameter estimation and aggregation problems and hence is present in many wide-spread models. It is best described in terms of choice probabilities:

Property 1 (Independence of Irrelevant Alternatives (IIA)) Let $\mathcal{S}$ be a set of products and denote by $\mathrm{P}(k \mid \mathcal{S})$ the probability that a customer chooses element $k \in \mathcal{S}$ when given the set of choices $\mathcal{S}$. A choice model satisfies the Independence of Irrelevant Alternatives (IIA) property, if for every $k, k' \in \mathcal{S}' \subseteq \mathcal{S}$ such that $\mathrm{P}(k' \mid \mathcal{S}) \neq 0 \neq \mathrm{P}(k' \mid \mathcal{S}')$

$$\frac{\mathrm{P}(k \mid \mathcal{S}')}{\mathrm{P}(k' \mid \mathcal{S}')} = \frac{\mathrm{P}(k \mid \mathcal{S})}{\mathrm{P}(k' \mid \mathcal{S})}. \tag{2.2}$$

This implies that

$$\mathrm{P}(k \mid \mathcal{S}') = \frac{\mathrm{P}_k}{\sum_{k' \in \mathcal{S}'} \mathrm{P}_{k'}} \tag{2.3}$$

where $\mathrm{P}_k = \mathrm{P}(k \mid \mathcal{S})$ for every $k \in \mathcal{S}$.

Remark 2.1.3 The IIA property implies that relative booking probabilities are carried over to a subset of the initial choice set or, in other words, that demand is not correlated between products.

One immediately sees the limitations implied by the IIA property by considering the classic red bus/blue bus example [92]: Suppose a commuter has to decide their mode of transportation between going by car and taking a (red) bus and assume the choice probabilities are

$$\mathrm{P}^{\mathrm{red}}_{\mathrm{bus}} = \frac{1}{2} \qquad\qquad \mathrm{P}_{\mathrm{car}} = \frac{1}{2}. \tag{2.4}$$

Now, if we introduce a second bus, that is blue but otherwise completely identical to the first one, we can safely assume that $\mathrm{P}^{\mathrm{blue}}_{\mathrm{bus}} = \mathrm{P}^{\mathrm{red}}_{\mathrm{bus}}$. If IIA holds we can use Eq. (2.2) and see that now

$$\mathrm{P}^{\mathrm{red}}_{\mathrm{bus}} = \frac{1}{3} \qquad\qquad \mathrm{P}^{\mathrm{blue}}_{\mathrm{bus}} = \frac{1}{3} \qquad\qquad \mathrm{P}_{\mathrm{car}} = \frac{1}{3}. \tag{2.5}$$

Therefore, the introduction of a new alternative that is equivalent to an already existing one leads to a reduced choice probability for other options.

There are various attempts to find models that relax the IIA property in ways that avoid this kind of unreasonable effect but still profit from some of its analytical advantages[80, 81]. Many approaches use choice hierarchies or other clustering of similar alternatives in order to introduce a correlation between choice probabilities. With these methods, considerable manual effort is required in deriving a model tailored to the specific application, depending on the attributes of the alternatives. Since we ultimately intend to optimize on product attributes, we cannot assume that our products are constant, prohibiting the use of fixed clusterings. On the other hand, dynamic clustering of variable products leads to difficulties as well: When varying continuous product attributes, the choice probabilities will change discontinuously whenever the discrete clustering changes. Most wide-spread models satisfying only a relaxed IIA property are therefore still not directly applicable to our problem.

2.1.1 Decision maker

In most cases the entity making the decision is a person, but it can also be a group of people or a company. While in the further case the final decision is the result of some kind of thought process, in the latter case it is the result of complex interaction among the individuals the group consists of. In our application the decision maker is always a single customer and we will therefore often write *customer* instead of *decision maker*.

We account for the fact that different individuals might make different decisions in the exact same circumstances by describing an individual using a vector of attributes, sometimes called *taste* variables. This vector is assumed to be deterministic per individual, but is usually unobservable for the analyst and varies among individuals. Thus, it has to be treated as a random variable.

Definition 2.1.4 (Decision Maker) The *decision maker* or *customer* is a random variable $\mathbf{X}$ taking values in the *customer space* $\mathbf{C}$. In general $\mathbf{X}$ can have both continuous and discrete components. Moreover, attributes are not necessarily mutually independent.

Remark 2.1.5 Typically the distribution of $\mathbf{X}$ or, in other words, the joint probability distribution of the individual attributes is modeled using a parametrized family of distributions, such as a multivariate normal distribution.

In order to make a customer model analytically and computationally tractable it is often assumed that the individual attributes are independent. In this case the joint distribution is just the product distribution constructed from the distributions of the individual attributes, which are again modeled using (possibly different) parametrized families of well-studied probability distributions.

For aggregation purposes it is convenient to use a distribution that can be described using a probability density function. In case all components of $\mathbf{X}$ are continuous variables this is a classical probability density function. For discrete or mixed random vectors the generalized concept of a density function in the sense of measure theory has to be used.

Each individual can then be seen as a realization $\mathbf{x}$ of this random vector $\mathbf{X}$. In particular, we assume that each customer has perfect information regarding their own attributes.

In some cases it seems like the customer makes a decision depending on an unknown quantity, violating this assumption. However, if the value of an attribute is unknown to the decision maker, it cannot truly enter into the decision making process. Rather, the individual will estimate related deterministic quantities, such as the expected value or the variance of the random attribute, and base the decision on these. This way, the randomness can be removed from the individual by using these derived quantities as the customer's attribute instead of the variable. The random nature of the original attribute then enters into the decision rule through the way the customer deals with the uncertainty.

Example 2.1.1 All customers are attending different business events with uncertain end times and need to choose which flight connection to use in order to fly back after their event ends. The end times of all events are identically and independently distributed. Therefor the end time t differs among different customers and is a random property of the individual. At the same time, the exact value of t might still not be known to the decision maker at the time of booking.

The customer might choose to book a flight that leaves as early as possible after the expected end of the meeting $\mu = \mathbb{E}(t)$, while keeping the risk of missing the flight below a certain threshold. Assuming a normal distribution $t \sim \mathcal{N}\left(\mu, \sigma^2\right)$ with expected value μ and variance σ^2, the customer will book the first flight with departure time later than μ plus a certain time buffer that depends on σ. In this reasoning, later flights are preferred if the variance σ^2 is high.

Instead of the unknown time t, the quantities actually influencing the decision are then the variables μ and σ that the customer uses to describe the uncertainty of t. Thus, the customer can be described using the attributes μ and σ, which are known to the individual. Note that these quantities can still be observably random to the analyst.

2.1.2 Set of alternatives

By definition, the choice by the decision maker is made from a nonempty set of alternatives, where each alternative is described by a vector of attributes or *characteristics*.

Definition 2.1.6 (Set of Alternatives) The (possibly infinite) *universal set of alternatives* is the set $\mathbf{P}$ of all imaginable options.

An actual choice will then always be made from a finite subset $\mathcal{S} \subseteq \mathbf{P}$ of the universal set, which we call the *choice set*. It includes all options that are available to the customer at the time of the decision.

We assume that the characteristics of the alternatives are observable to the analyst. A customer's uncertainty about product attributes can again be modeled by incorporating the uncertainty in the customer attributes. Similar to the above, the decision will be based on the decision maker's expectation or prior distribution of the unknown quantity rather than the correct value. Assuming that, from the customer's point of view, the unknown attributes follow a parametrized probability distribution, the decision can be modeled by considering the distribution parameters as customer attributes and introducing a suitable decision rule. Therefore, we can assume w.l.o.g. that each alternative is described by a deterministic vector $\mathfrak{p}$.

Definition 2.1.7 In the following, let always $\mathcal{S} = \{1, \dots, M\}$ be a set of products, and $\mathfrak{p}_k \in \mathbf{P}$ the vector of attributes of product k.

In some cases it is more convenient to directly work with the vectors of product attributes. Slightly abusing notation, we then identify the set of products $\mathcal{S} = \{1, \dots, M\}$ with the corresponding subset $\{\mathfrak{p}_1, \dots, \mathfrak{p}_M\} \in \mathbf{P}$ of the product space and therefore say that $\mathcal{S} \subset \mathbf{P}$. For an attribute vector $\mathfrak{p} \in \mathbf{P}$ we then also write $\mathfrak{p} \in \mathcal{S}$ if and only if $\mathfrak{p} \in \{\mathfrak{p}_1, \dots, \mathfrak{p}_M\}$.

Alternatively, we identify $\mathcal{S}$ with the product matrix $(\mathfrak{p}_1 \,|\, \dots \,|\, \mathfrak{p}_M)$, the columns of which are the product vectors.

In practice, not all products are available to all customers. Therefore, the set of alternatives is not necessarily the same for all customers. Availability of an option to a specific customer can be determined by various factors, depending on the attributes of the alternatives as well as those of the decision maker, for example:

Availability The choice set for the customer is determined by external constraints such as limited stock or restrictions regarding transportation, but also constraints that may be actively controlled by some entity, such as an airlines revenue management system.

Information Clearly, a customer can only make a choice from the set of options they know about.

Other Constraints Depending on product and customer attributes, some alternatives are infeasible for the customer, for example because of a limited budget or time constraints. In order to make the set of alternatives independent from the individual customer, this kind of constraint is often treated as part of the decision rule (see below).

2.1.3 Decision rule

Given a single customer, represented by a deterministic vector of attributes $\mathbf{x}$, and a set of alternatives $\mathcal{S} = \{1, \dots, M\}$, there are numerous ways to model the decision process. The main intricacy is to describe how a customer weighs drawbacks and benefits of different options against each other. Depending on the way this is done, different customer attributes may be necessary. In our treatment of discrete choice modeling we will always assume *rational customer behavior*. Because there is no agreement in the literature what this actually means, we will shortly summarize some of the aspects of what we consider a rational choice.

- The customer makes a deliberate, logical decision according to their own objectives. In particular, the decision is not influenced by impulsiveness or spontaneity. This means that the choice itself is a mathematical optimization problem with a well-defined feasible set and objective function.
- The outcome of the decision making process does not depend on foreign influences such as interactions with third parties.
- Preferences are transitive.
- The decision is predictable, i.e. a fixed customer will always make the same decision given the same set of alternatives. We will show below that this is without loss of generality, because it can always be achieved via a model transformation.

- There are no psychological effects such as the fact that customers tend to purchase less when the set of alternatives grows too large, because they are overwhelmed by the number of choices [64, 105, 71, 98, 18]. This behavior would also contradict the first and third items in this list: If from the customer's point of view the choice problem is essentially an optimization problem, a larger feasible set can only improve the outcome of the overall decision. Also, when adding additional alternatives to the choice set, the customer should either choose one of the newly added options or stay with their original choice, but not switch to a different alternative or the *no purchase* option, which was already available beforehand.

Remark 2.1.8 Instead of the idealized framework of *perfect rationality* presented above, the concept of *bounded rationality* [110] is used in many applications. Instead of assuming an omniscient individual, this approach deals with the fact that humans can only grasp and utilize a certain amount of information for their decision making process. However, since an airline product is mainly defined by abstract attributes like cost and cancellation fees, the airline market is much more transparent for the customer than, for example, the retail industry or automotive market, where features like design or quality of materials are much more important, but cannot be quantified as easily. We feel that, for our application, the complexity arising from the treatment of bounded rationality vastly outweighs the benefits. We will therefore assume perfect rationality for the rest of this work and leave the inclusion of bounded rationality for further research.

Definition 2.1.9 (Deterministic Decision Rule) Let $\mathbf{X}$ denote the random variable describing the attributes of the decision maker, taking values in the customer space $\mathbf{C}$. Each product, described by a number of attributes, is an element of the product space $\mathbf{P}$. A *decision rule* is a function that maps a customer and an offer set to the customer's choice, which is an element of the offer set, and that satisfies the transitivity assumption (Definition 2.1.1).

In other words, it is a function

$$\mathfrak{p}^*: \mathbf{C} \times \wp_{<\infty}(\mathbf{P}) \to \mathbf{P} \tag{2.6a}$$

$$(\mathbf{x}, \mathcal{S}) \mapsto \mathfrak{p}^*(\mathbf{x}, \mathcal{S}) \in \mathcal{S} \tag{2.6b}$$

mapping a realization $\mathbf{x}$ of $\mathbf{X}$ and an offer set $\mathcal{S}$ to a product $\mathfrak{p} \in \mathcal{S}$. Here, $\wp_{<\infty}(\mathbf{P})$ is the set of finite subsets of $\mathbf{P}$.

Applying the decision rule (Definition 2.1.9) to the random customer variable $\mathbf{X}$ instead of a realization $\mathbf{x}$ we obtain the discrete random variable $\mathfrak{p}^*(\mathbf{X}, \mathcal{S})$ with values in $\mathcal{S}$.

Definition 2.1.10 (Booking Probability) Let $\mathcal{S} \subseteq \mathbf{P}$ be an offer set and $\mathfrak{p} \in \mathcal{S}$ a product. The *booking probability* for $\mathfrak{p} \in \mathcal{S}$ is defined as

$$d_{\mathfrak{p}}(\mathcal{S}) = d_{\mathfrak{p}}(\mathbf{X}, \mathcal{S}) := \mathrm{P}\left[\mathfrak{p}^*(\mathbf{X}, \mathcal{S}) = \mathfrak{p}\right]. \tag{2.7}$$

When the set of products $\mathcal{S} = \{1, \ldots, M\}$ and their attribute vectors $\mathfrak{p}_1, \ldots, \mathfrak{p}_M$ are fixed, we again identify product k with the attribute vector $\mathfrak{p}_k$ and denote its booking probability by

$$d_k(\mathcal{S}) = d_k(\mathbf{X}, \mathcal{S}) := \mathrm{P}\left[\mathfrak{p}^*(\mathbf{X}, \mathcal{S}) = \mathfrak{p}_k\right]. \tag{2.8}$$

The aggregation problem for discrete choice models is the problem of computing this booking probability for a known distribution of the customer variable $\mathbf{X}$ and a given offer set $\mathcal{S}$.

Definition 2.1.11 (Probabilistic Decision Rule) A *probabilistic* or *random decision rule* maps a realization $\mathbf{x}$ of $\mathbf{X}$ and an offer set $\mathcal{S}$ to a discrete random variable with values in the finite set $\mathcal{S}$. Because such a finite discrete random variable is uniquely identified by the respective probabilities of each element in the underlying set, the decision rule can be described as a function

$$d: \mathbf{C} \times \wp_{<\infty}(\mathbf{P}) \to \mathbb{R}^{\mathbf{P}} \tag{2.9}$$

$$(\mathbf{x}, \mathcal{S}) \mapsto \left(d_{\mathfrak{p}}(\mathbf{x}, \mathcal{S})\right)_{\mathfrak{p} \in \mathbf{P}} \tag{2.10}$$

Product	t	γ	f
A	10:00	No	400
B	10:00	Yes	600
C	16:00	No	300
D	16:00	Yes	500

(a) Products

Customer	T	α	B
1	11:00	0.25	1000
2	18:00	0.5	1000
3	12:00	0	400
4	16:00	0	400

(b) Customers

Table 2.1: Data for Example 2.1.2

mapping a realization $\mathbf{x}$ of $\mathbf{X}$ and an offer set $\mathcal{S}$ to a vector of booking probabilities indexed by the product space, satisfying

$$d_{\mathfrak{p}}(\mathbf{x}, \mathcal{S}) = 0 \qquad \forall \mathfrak{p} \notin \mathcal{S} \tag{2.11}$$

$$\sum_{\mathfrak{p} \in \mathcal{S}} d_{\mathfrak{p}}(\mathbf{x}, \mathcal{S}) = 1. \tag{2.12}$$

In this case, the transitivity property of the choice model (see Definition 2.1.1) is equivalent to

$$d_{\mathfrak{p}}(\mathbf{x}, \mathcal{S}') \geq d_{\mathfrak{p}}(\mathbf{x}, \mathcal{S}) \qquad \forall \mathfrak{p} \in \mathcal{S}' \subseteq \mathcal{S} \subset \mathcal{P}. \tag{2.13}$$

In practice it is very hard to directly model a random choice, described by the booking probabilities, without very restrictive assumptions like the IIA property. This is particularly true if product attributes are not fixed and a realistic model response to a change of product attributes or availability is required. However, one can derive a probabilistic model from a deterministic one by treating some of the customer attributes as unobservable. Conversely, many random models can be transformed into a deterministic rule through the use of additional customer attributes capturing the random effects.

We will at first only introduce deterministic decision rules and then present a method to use these to construct a probabilistic model Section 2.1.4. In the discussion of the different options, we will use an example in order to illustrate the ideas instead of going into too much technical detail.

Example 2.1.2 We consider airline customers choosing between four products distributed across two flights departing at different times, with the attributes for each product $\mathfrak{p} = (t, f, \gamma)$ being departure time t, price f and a binary variable γ indicating whether the product is refundable or not. There is one refundable and one non-refundable product for each flight. Product attributes are listed in Table 2.1a. The main customer attributes are the desired departure time T, a budget B and the probability α that the customer will have to cancel the booking. Attributes for the sample customers are listed in Table 2.1b. We will use the example data provided in Tables 2.1a and 2.1b throughout this section.

The goals for a customer are to spend as little money as possible and departing close to their desired departure time, while staying in their limited budget. The expected cost C for customer $\mathbf{x} = (T, B, \alpha)$ and product $\mathfrak{p}$ is given by

$$C(\mathbf{x}, \mathfrak{p}) = \begin{cases} (1-\alpha) f & \text{if product refundable,} \\ f & \text{else.} \end{cases} \tag{2.14}$$

Dominance

An alternative A is said to *dominate* another alternative B, if A is at least as good as B in all respects and, as a consequence, alternative B can be removed from the choice set without negative impact on the outcome for any customer. The concept of dominance can only be used if for every customer $\mathbf{x}$ and every product attribute $\mathfrak{p}_i$ there is an ordering on the possible values for

$\mathfrak{p}_i$ according to the customer's preference. In particular, this ordering may not depend on the values of other attributes. In other words, for two products that only differ in exactly one of their attributes it must be possible to uniquely deduce the customer's decision only from the values of this attribute. This assumption is often satisfied; for instance a customer will always choose the cheaper one of two otherwise identical products.

In Example 2.1.2, this assumption is satisfied for all three attributes: Clearly, a lower price is always preferable, as is a refundable product to a non-refundable one, while the preference in departure time now depends on the specific customer.

If a product $\mathfrak{p}_1$ is at least as good as $\mathfrak{p}_2$ with respect to each individual attribute, $\mathfrak{p}_1$ is said to dominate $\mathfrak{p}_2$. If, for a specific customer, one alternative dominates all other feasible options, it is obviously their optimal choice in any sensible decision logic. If more than one attribute is considered, the existence of a dominant alternative is not guaranteed and even depends on the decision maker's attributes. For this reason, in practice the concept of dominance is rarely used as the primary decision rule. However, it is often applied to ensure uniqueness of a customer's decision by using a dominance criterion as a tie-breaker.

Example (Continued from Example 2.1.2) Coming back to the above example, we see that customers 3 and 4 can only afford products A and C. For cust. 4, product C dominates A. For cust. 3 there is no dominant alternative, because A departs closer to the preferred departure time, but C is cheaper.

As a generalization, dominance can also be applied to criteria that are derived from product characteristics and customer attributes.

Example (Continued from Example 2.1.2) The expected cost that a customer tries to minimize in our example is a good illustration. Using Eq. (2.14), we see that expected costs for customers 1 and 2 are as follows:

Customer \ Product	A	B	C	D
1	400	450	300	375
2	400	300	300	250

Thus, product D is dominant for customer 2, because deviation from the desired departure time as well as expected cost are minimal. Note that product D is not a dominant choice if the individual product attributes are considered separately, because D is more expensive than C.

For Customer 1 there is still no dominant alternative, again emphasizing the fact that the existence of a dominant choice depends on the specific customer.

Satisfaction

A customer might decide to only accept alternatives that exceed a certain threshold, called *satisfaction level*, for all or a subset of the attributes or other criteria that are derived from the attributes of both the product and the customer. Again, this rule alone does not necessarily lead to a decision, because it is not guaranteed that exactly one option will be satisfactory. However, it can be used in conjunction with other decision rules such as dominance. In particular, if a satisfaction level is chosen for all but one criterion, the concept of dominance can be used on the remaining one and, excluding ties, a unique decision is guaranteed.

Satisfaction levels can be identical for all decision makers or differ between customers. In the latter case, individual thresholds are included in the vector of customer attributes.

Example (Continued from Example 2.1.2) The budget constraint in Example 2.1.2 is a classic example of this kind of decision rule. Combining satisfaction levels with dominance, a decision rule could be: *Choose the product that minimizes expected cost (Eq.* (2.14)*) among all alternatives that satisfy the budget constraint and depart within four hours of the desired departure time.*

Lexicographic rules

By ordering the attributes or other decision criteria with respect to their importance to the decision maker, a decision can be obtained by maximizing the most important criterion first and consecutively using the other criteria as tie breakers in decreasing order of importance. This kind of decision rule is problematic, because in most cases only one criterion is relevant, while all others can be arbitrarily unfavorable without having an influence on the decision. As a consequence, the trade-off between upsides and downsides of different options cannot be reasonably modeled.

Utility

All of the rules presented above suffer from the non-comparability of different product attributes or derived decision criteria between each other. Yet, in practice a customer will generally try to choose the alternative that has maximum expected overall use for them. Therefore a natural decision rule is to choose the product that maximizes their personal *utility*, which is a *scalar* measure for the all-around usefulness of a product depending on the particular customer.

Definition 2.1.12 (Utility) A *utility function* is a scalar function

$$u\colon \mathbf{C} \times \mathbf{P} \to \mathbb{R} \tag{2.15a}$$

$$(\mathbf{x}, \mathfrak{p}) \mapsto u(\mathbf{x}, \mathfrak{p}) \tag{2.15b}$$

mapping a customer $\mathbf{x}$ and an alternative $\mathfrak{p}$ to a measure of utility $u(\mathbf{x}, \mathfrak{p})$. In general we do not make any assumptions regarding continuity or smoothness of u.

A customer $\mathbf{x}$ will then choose the product that maximizes their personal utility $u(\mathbf{x}, \cdot)$. In order to ensure a unique optimal decision, an additional criterion has to be used as a tie breaker in case multiple products maximize $u(\mathbf{x}, \cdot)$. Common choices are an arbitrary fixed ordering on the set of products or a lexicographic decision rule (see above).

In some models, the utility function can simply be derived from a descriptive criterion such as minimal expected cost or travel time. Here, a value of the utility function usually has a unit and a direct interpretation and hence carries a good deal of intrinsic information. A utility function of this type is called a *cardinal utility.* The model can then even be used to compare products that are fundamentally different in the attributes that define them.

Example The criterion of minimal expected cost (Eq. (2.14)) introduced above can be used to compare the products defined in Example 2.1.2 with different modes of travel such as renting a car or going by train, assuming that expected cost can be computed for all relevant options.

On the other hand, if the utility is only used to capture the trade-off between attributes, often the value of a utility measure does not convey any information on its own. It only provides a way to compare two or more products using the exact same utility function and is called an *ordinal utility.* However, the parameters involved are often very descriptive and carry a lot of information about customer behavior.

Example Instead of the satisfaction constraint on the departure time, we can model the compromise between price and favorite departure time by using the utility function

$$u\colon \mathbf{C} \times \mathbf{P} \to \mathbb{R} \tag{2.16a}$$

$$(\mathbf{x}, \mathfrak{p}) \mapsto C(\mathbf{x}, \mathfrak{p}) + \delta\,|T - t| \tag{2.16b}$$

where again $C(\mathbf{x}, \mathfrak{p})$ denotes the customer's expected cost as in Eq. (2.14) and the parameter δ is one of the following:

- A global parameter
- An additional customer attribute (possibly correlated to the other attributes)
- A scalar function depending on the customer attributes, for example modeling the fact that a customer with a higher budget will be willing to pay a higher fare difference than someone with a lower budget in order to get a more convenient flight time:

$$\delta(\mathbf{x}) = \hat{\delta}B$$

  with a global parameter $\hat{\delta}$ and budget B.

In any of these cases δ quantifies the virtual cost a customer associates with the absolute deviation from their expected departure time.

In order to directly estimate the abstract quantity of utility from actual data and to give it a meaning, it has been proposed to assume that utility is proportional to the willingness-to-pay of a customer with respect to the given product:

> Utility is taken to be correlative to Desire or Want. It has been already argued that desires cannot be measured directly, but only indirectly, by the outward phenomena to which they give rise: and that in those cases with which economics is chiefly concerned the measure is found in the price which a person is willing to pay for the fulfillment or satisfaction of his desire. (A. Marshall, 1895 [85])

Of course in this interpretation it does not make sense to consider price to be a product attribute. Through experiments involving different prices and only one product at a time, the economist can then theoretically estimate utility directly. In practice, however, the existence of multiple competing alternatives will usually not allow the analyst to reliably estimate willingness-to-pay.

In many applications a utility based model, often combined with one more of the other criteria presented above, is the most plausible way to describe customer choice. The main advantage from the analyst's point of view is that the problem is reduced to finding a good utility model. Instead of having to deal with correlation of demand between different products, one only has to consider one customer and one product at a time. Given a utility model, choice probabilities for an unknown customer can be described in terms of the parameters of the known distribution of customer attributes (Eq. (2.7)). This way, the high-level problem of modeling aggregate demand depending on the attributes of all alternatives, including substitution effects, can be transformed to a lower-level problem. Depending on the customer model, the explicit computation of booking probabilities through this representation has to be performed numerically (see Section 2.2.

Having defined a utility model, practical application requires estimates for the parameters that define it. If choice probabilities can be derived analytically from the utility function and the distribution of customer attributes, one can obtain a joint estimate of the vector of parameters from historical data using standard statistical methods for parameter estimation. Whenever some or all of the parameters defining the utility function have a direct interpretation, individual parameters can also be measured directly, for example by carrying out surveys.

Remark 2.1.13 (Constant Utility) In our formulation as presented above we move all random components of the model to the decision maker and interpret them as effects arising from random customer attributes, following the assumption that the customer will make a perfect decision based on their attributes and that the market is fully transparent.

The *constant utility model* is an alternative way of dealing with randomness and assumes that the variables that affect utility, most importantly attributes of customers, are deterministic. Randomness of choice arises from a modified decision rule, where the customer does not necessarily choose the alternative with the highest utility, but the customer's choice follows a discrete random distribution on the set of alternatives, where the individual choice probabilities for each option depends on the utilities. The obvious problem with this approach is that one has to model the distribution and its dependence on the utilities explicitly. This is particularly complex because one such distribution has to be provided for every subset of the total set of products, and all of these distributions have to be compatible with each other in terms of the transitivity property (Definition 2.1.1).

The easiest and most common constant utility model is derived from the *choice axiom* introduced by Luce [79], which states that, if some elements of a choice set are removed, the relative probabilities of the remaining alternatives do not change. The property is closely related to IIA (see property 1), although in this case we are trying to model the choice probabilities for a single known customer while a general choice model tries to model booking probabilities for an set of unknown customers. Since the effects of the choice axiom are largely the same as those of IIA, models of this type cannot be applied in our case.

Most approaches at alleviating these problems again use hierarchies and are similar to the attempts to deal with IIA. Thus, none of the generalizations are very useful for the application at hand.

Random decisions

In the literature, a common scenario is that all alternatives have fixed attributes and all products and decision makers can be uniquely identified. Under these assumptions it is possible to satisfy the predictability condition stated above while still allowing two customers with the exact same attributes to make different decisions. This can for example be modeled using a random error term in the utility function (see below) for each customer/product combination.

Since we intend to optimize on the products' attributes, we need our model to yield consistent results with respect to changes in attributes. In addition, we expect that after a permutation of the set of alternatives the customer will still choose the same product. Thus we cannot allow the choice to depend on an arbitrary label assigned to a product, such as a product name. Instead, the choice has to depend only on attributes of the alternatives and the customer. Any random influences, such as error terms in the utility function, must be modeled as dependent on the customer and the anonymous product as defined by its attributes. One way to construct a random demand model that satisfies these requirements is the following:

2.1.4 Derived customer models

In some cases not all customer attributes are observable to the analyst, and as a result the decision made by a customer may seem like a random decision instead of a deterministic one. In order to be able to use the data we do have, we derive a probabilistic customer model from a deterministic one as follows:

Let $\mathbf{P}$, $\mathbf{X}$ and $\mathbf{C}$ be as above. Let $\mathfrak{p}^*$ be a deterministic decision rule. We introduce a random vector $\mathbf{Y}$ with values in $\mathbf{D}$ that only contains some of the information contained in $\mathbf{X}$. We can then derive a probabilistic model for $\mathbf{Y}$ from the deterministic decision rule for $\mathbf{X}$. Let

$$\pi\colon \mathbf{C} \to \mathbf{D} \tag{2.17a}$$

$$\mathbf{x} \mapsto \pi(\mathbf{x}) \tag{2.17b}$$

be a measurable function mapping a customer instance $\mathbf{x}$ to the observable information vector $y = \pi(\mathbf{x})$. Although we do not impose any restrictions on π it will often be the projection to a subspace of $\mathbf{C}$, simply discarding some of the attributes.

A random customer is described by the random variable $\mathbf{Y} = \pi(\mathbf{X})$ with values in the customer space $\mathbf{D}$ following the probability distribution induced by the distribution of $\mathbf{X}$.

The probability of customer y purchasing product $\mathfrak{p}$ among the alternatives $\mathcal{S}$ is the conditional choice probability of a random customer $\mathbf{X}$ given that $\pi(\mathbf{X}) = y$:

$$d_{\mathfrak{p}}(y, \mathcal{S}) = d_{\mathfrak{p}}(\mathbf{X} \mid \mathbf{Y} = y, \mathcal{S}) = \mathrm{P}\left[\mathfrak{p}(\mathbf{X} \mid \mathbf{Y} = y, \mathcal{S}) = \mathfrak{p}\right]. \tag{2.18}$$

Now, by definition of d and the probability distribution of $\mathbf{Y}$, we see that the overall choice probabilities satisfy

$$d_{\mathfrak{p}}(\mathbf{Y}, \mathcal{S}) = d_{\mathfrak{p}}(\mathbf{X}, \mathcal{S}). \tag{2.19}$$

2.2 Aggregation

The application of a discrete choice model for optimization purposes requires the possibility to compute measures of aggregate demand, such as an expected value for the number of customers or the arrival rate of a Poisson process. Most customer models assume that customers arrive one by one and make *independent* booking decisions depending on the set of alternatives they are offered. For most practical applications it is reasonable to assume that the offer only changes finitely often over the selling horizon. Let $I = [t^{\mathrm{s}}, t^{\mathrm{e}}] \subseteq [0, T]$ be a subset of the booking horizon with a constant offer set $\mathcal{S} \subseteq \mathcal{P}$. Depending on the type of demand model that is used, we then distinguish the following two cases:

Stochastic Arrival Process Let $\mathbf{N}$ be a stochastic arrival process on I that describes the arrival of individual customers. The booking process $\mathbf{N}_{\mathfrak{p}}$ on I for each product $\mathfrak{p} \in \mathcal{S}$ is then a filtration of $\mathbf{N}$.

> **Example 2.2.1 (Poisson arrival process)** Let λ denote the (time-dependent) rate of the Poisson arrival process $\mathbf{N}$ on I. Because the individual booking decisions are independent between each other and independent of $\mathbf{N}$, the filtered process is again a Poisson process with rate
>
> $$\lambda_{\mathfrak{p}}(t) = \lambda(t) d_{\mathfrak{p}}(t, \mathcal{S}), \tag{2.20}$$
>
> where $d_{\mathfrak{p}}(t, \mathcal{S})$ is the probability that a random customer arriving at time t will book product $\mathfrak{p}$ given the set of alternatives $\mathcal{S}$.

Number of Arrivals Let $\mathbf{N}$ be a random variable describing the number of customer arrivals during the interval I and assume that decisions are independent between each other and independent of $\mathbf{N}$. The expected number of bookings for product $\mathfrak{p}$ can be computed as

$$\mathbb{E}[\mathbf{N}_{\mathfrak{p}}] = \mathbb{E}[\mathbf{N}]\, d_{\mathfrak{p}}(I, \mathcal{S}) \tag{2.21}$$

where the booking probabilities

$$d_{\mathfrak{p}}(I, \mathcal{S}) = d_{\mathfrak{p}}(t, \mathcal{S}) \qquad \forall t \in I \tag{2.22}$$

are constant on I.

In other words, aggregate demand for each product given an offer set can easily be derived from the booking probabilities d. The remainder of this section will therefore be concerned with the computation of $d(t, \mathcal{S})$ for a fixed t. We will omit the explicit dependence of d on the request time t for the sake of simplicity.

By definition (see Eq. (2.7)) the booking probability is given by

$$d_{\mathfrak{p}}(\mathcal{S}) = \int d_{\mathfrak{p}}(\mathbf{X}, \mathcal{S})\, \mathrm{d}\mathbf{X}, \tag{2.23}$$

where $d_{\mathfrak{p}}(\mathbf{x}, \mathcal{S})$ denotes the booking probability for a fixed customer $\mathbf{x}$. In the following we will always assume that the distribution of $\mathbf{X}$ is described by a (generalized) density function

$$f \colon \mathbf{C} \to \mathbb{R}. \tag{2.24}$$

Equation (2.23) then becomes

$$d_{\mathfrak{p}}(\mathcal{S}) = \int_{\mathbf{C}} f(\mathbf{x}) d_{\mathfrak{p}}(\mathbf{x}, \mathcal{S})\, \mathrm{d}\mathbf{x}. \tag{2.25}$$

For the actual computation we have to distinguish between two cases:

Deterministic Decision Rule In case the decision rule is deterministic we can partition the set of customers as follows.

Definition 2.2.1 (Customer Set) Let $\mathcal{S} \subseteq \mathbf{P}$ be a finite offer set and $\mathfrak{p} \in \mathcal{S}$. The *customer set* $\mathcal{X}_{\mathfrak{p}}(\mathcal{S})$ is the set of customers who decide to book product $\mathfrak{p}$ given the set of alternatives $\mathcal{S}$. It is given by

$$\mathcal{X}_{\mathfrak{p}}(\mathcal{S}) = \{\mathbf{x} \in \mathbf{C} \mid \mathfrak{p}^*(\mathbf{x}, \mathcal{S}) = \mathfrak{p}\}. \tag{2.26}$$

Now we see that, by definition,

$$d_{\mathfrak{p}}(\mathcal{S}) = \mathrm{P}\left(\mathbf{X} \in \mathcal{X}_{\mathfrak{p}}(\mathcal{S})\right) \tag{2.27}$$

and, since in this case $d_{\mathfrak{p}}(\cdot, \mathcal{S})$ is just an indicator function for $\mathcal{X}_{\mathfrak{p}}(\mathcal{S})$, Eq. (2.25) becomes

$$d_{\mathfrak{p}}(\mathcal{S}) = \int_{\mathcal{X}_{\mathfrak{p}}(\mathcal{S})} f(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{2.28}$$

With Eq. (2.28) we can decompose the problem into two sub-problems:

(1) Identifying the customer set $\mathcal{X}_{\mathfrak{p}}(\mathcal{S})$. The computational complexity of this step heavily depends on the decision rule. The set $\mathcal{X}_{\mathfrak{p}}(\mathcal{S})$ is independent of the distribution of $\mathbf{X}$, but only depends on the decision rule and the product characteristics of all alternatives.

(2) Computing the probability that $\mathbf{X} \in \mathcal{X}_{\mathfrak{p}}(\mathcal{S})$. Here the decision rule that was used and the attributes of the alternatives only enter implicitly trough $\mathcal{X}_{\mathfrak{p}}(\mathcal{S})$ and the result only directly depends on the distribution of $\mathbf{X}$.

Probabilistic Decision Rule In this case the evaluation of d amounts to a straight-forward computation of the integral in the right-hand side of Eq. (2.25), if for every customer $\mathbf{x}$, choice set $\mathcal{S}$ and product $\mathfrak{p} \in \mathcal{S}$ the booking probabilities $d_{\mathfrak{p}}(\mathbf{x}, \mathcal{S})$ are continuous and almost everywhere continuously differentiable in both $\mathbf{x}$ and the attributes of the alternatives contained in $\mathcal{S}$.

2.2.1 Common methods

In the following we will summarize a number of exact and heuristic methods that are commonly used to solve the aggregation problem in order to be able to discuss their respective shortcomings with respect to our application and thus justify the introduction of a new approach.

We will again use an example to illustrate the ideas.

Example 2.2.2 Consider a customer model where the only customer attribute is a budget. Assume the budget is non-negative and its distribution has the density function

$$\begin{aligned} f\colon \mathbb{R}_+ &\to \mathbb{R} \\ \mathbf{x} &\mapsto f(\mathbf{x}). \end{aligned}$$

The offer set contains the products $1, \ldots, M$ with strictly monotonous prices $f_1 < \cdots < f_M$. Assuming that price is an indicator for quality, the customer will purchase the most expensive product they can afford. Then the booking probabilities can be computed as

$$d_k = \int_{f_k}^{f_{k+1}} f(\mathbf{x})\,\mathrm{d}x \qquad \forall k = 1, \ldots, M-1 \tag{2.30a}$$

$$d_M = \int_{f_M}^{\infty} f(\mathbf{x})\,\mathrm{d}x \tag{2.30b}$$

Analytical integration

If the decision rule as well as the distribution of $\mathbf{X}$ are simple enough, it is sometimes possible to compute the value of $d_{\mathfrak{p}}(\mathcal{S})$ analytically as a function of the attributes of $\mathfrak{p}$ and the alternatives contained in $\mathcal{S}$.

> **Example** If the density function f has an anti-derivative, a closed functional form for d can be computed by solving the integrals in Eq. (2.30) analytically.

The most prominent cases are the multinomial logit and multinomial probit models presented below. The general idea is to use a utility based model with a (possibly random) utility function

$$u(\mathbf{x}, \mathfrak{p}) = \theta^\top \mathbf{x} + \eta^\top \mathfrak{p} + \epsilon_{\mathbf{x},\mathfrak{p}} \tag{2.31}$$

that is linear in all attributes. The systematic components of the utility are modeled via the coefficient vectors θ and η, which describe the linear dependence of the utility on customer and product attributes respectively. The error term ϵ adds a random component, which is independent of customer or product attributes. Assuming a certain joint distribution for the error term ϵ, one can analytically compute aggregate demand as a function of $\mathbf{x}$ and $\mathfrak{p}$.

Nested numerical integration

If the distribution of $\mathbf{X}$ is described by a density function that can be easily evaluated and the shape of the set of customers we are integrating over is sufficiently well-behaved, we can directly apply numerical integration techniques to Eq. (2.25).

> **Example** This is particularly easy if there is only one customer attribute with a known distribution. For example going back to Example 2.2.2, which uses a deterministic decision rule, we see that, if the density function f can be evaluated efficiently, d can be computed numerically by applying standard quadrature methods to the integrals in Eq. (2.30).

In higher dimensions the model has to be chosen in a way such that one can easily determine the customer set $\mathcal{X}_\mathfrak{p}(\mathcal{S})$ that the respective integral runs over. For a probabilistic decision rule this is the is the whole customer space, while for a deterministic decision rule it is a subset.

If possible, the naive approach is to describe $\mathcal{X}_\mathfrak{p}(\mathcal{S})$ by lower and upper bounds $\mathbf{x}_j^{\mathrm{l}}(\mathbf{x}_1, \ldots, \mathbf{x}_{j-1})$ and $\mathbf{x}_j^{\mathrm{u}}(\mathbf{x}_1, \ldots, \mathbf{x}_{j-1})$ for the individual n attributes, each depending on the values of all preceding attributes (w.r.t. an arbitrary ordering). In this case the right-hand-side of Eq. (2.25) can be written as the nested integral

$$d_\mathfrak{p}(\mathcal{S}) = \int_{\mathbf{x}_1^{\mathrm{l}}}^{\mathbf{x}_1^{\mathrm{u}}} \int_{\mathbf{x}_2^{\mathrm{l}}(\mathbf{x}_1)}^{\mathbf{x}_2^{\mathrm{u}}(\mathbf{x}_1)} \cdots \int_{\mathbf{x}_n^{\mathrm{l}}(\mathbf{x}_1,\ldots,\mathbf{x}_{n-1})}^{\mathbf{x}_n^{\mathrm{u}}(\mathbf{x}_1,\ldots\mathbf{x}_{n-1})} f(\mathbf{x})\,\mathrm{d}\mathbf{x}_n \ldots \mathrm{d}\mathbf{x}_2\,\mathrm{d}\mathbf{x}_1. \tag{2.32}$$

The booking probabilities can then be computed using nested calls to a one-dimensional numerical integration routine. This approach can be implemented easily, but does not scale well with growing dimension n. On top of performance issues, error control of adaptive methods will fail, because they usually assume that the integrand can be evaluated accurately. Since the integrand for the outer integrals is computed by an inexact quadrature rule as well, this assumption is violated.

Instead of the nested formulation we propose to use higher dimensional adaptive quadrature rules if the shape of $\mathcal{X}_\mathfrak{p}(\mathcal{S})$ permits it. For more detail see Chapter 7.

Monte Carlo methods

The class of *Monte Carlo* integration methods is a set of algorithms for the computation of (usually higher dimensional) integrals that use repeated random sampling. Consider the integral

$$I = \int_X f(x)\,\mathrm{d}x \tag{2.33}$$

where $X \subset \mathbb{R}^N$ and

$$f\colon X \to \mathbb{R} \tag{2.34}$$

is square-integrable.

The plain Monte Carlo method for Eq. (2.33) works as follows:

(1) Choose a superset $Y \supseteq X$ such that one can easily draw IID random samples from a uniform distribution on Y.

(2) Extend f to a function on Y:

$$\bar{f}: Y \to \mathbb{R} \tag{2.35}$$

$$y \mapsto \begin{cases} f(y) & \text{if } y \in X, \\ 0 & \text{else} \end{cases} \tag{2.36}$$

(3) Draw n such samples $y_1, y_2, \ldots, y_n \in Y$.

(4) Compute an estimate of I as

$$I \approx \frac{V(Y)}{n} \sum_{i=1}^{n} \bar{f}(y_i) \tag{2.37}$$

where

$$V(Y) = \int_Y 1 \, \mathrm{d}y \tag{2.38}$$

is the volume of Y.

It is a direct consequence of the Central Limit Theorem [122, Chapter 18], that the RHS of Eq. (2.37) converges to the true value of I as $n \to \infty$ with an expected error proportional to $\frac{1}{\sqrt{n}}$.

Remark 2.2.2 Due to the fact that samples that are not elements of X do not contribute any information to the integral, the method becomes more accurate when the number of these points is reduced. More specifically, the expected error of the estimate is proportional to $\frac{V(Y)}{V(X)}$ [56]. Thus it is favorable to choose Y as small as possible, preferably even $Y = X$ if samples can be drawn easily from X.

More accurate results can be obtained with techniques that aim at reducing the variance of the estimate using a fixed number of sample points. Two well-known algorithms are the VEGAS-algorithm [75, 76], implementing *importance sampling* [57, 86], and the MISER-algorithm [101], implementing *adaptive stratified sampling* [45].

Average individual, classification and statistical differentials

The basic idea behind the *average individual* method is to generate a customer $\bar{x}$ that will represent the whole population of the customer set $\mathcal{X}_{\mathfrak{p}}(\mathcal{S})$, with the natural choice for $\bar{x}$ being expected value

$$\bar{x} = \mathbb{E}[\mathbf{X}]. \tag{2.39}$$

The behavior of a random individual $\mathbf{X}$ is then assumed to be identical to the behavior of the average individual.

While for a probabilistic decision rule this approach might provide viable results, it will yield an aggregate demand of zero for all but one product for deterministic rules, which makes it utterly useless in this case.

Classification is an extension of the average individual method where the set of potential customers is clustered into disjoint subsets according to their attributes and the average individual method is applied to each customer group. This accounts for different behavior depending on customer attributes and thus yields much better results.

The method of *statistical differentials* is a different extension of the average individual method, which only applies to probabilistic decision rules. The idea is to expand the integrand as a Taylor-series around the average individual and to integrate the (truncated) series over the same integration region. This way, the result contains approximate information about variation of the booking probability with respect to variation in customer attributes.

2.3 Widespread models

In this section we will quickly review some models that have successfully been used for demand analysis in various applications in the past. These models differ from the general setting introduced above in that customer attributes are not always used explicitly but rather enter implicitly through generic error terms. We will first introduce some basic notation that we will use as a foundation and then give a short overview over assumptions, benefits, and drawbacks for each of the individual models.

We denote by $\mathcal{P}$ the set of all potential choices for the whole population and by $\mathcal{S}_n \subseteq \mathcal{P}$ the set of potential choices for decision maker n with finite cardinality $|\mathcal{S}_n| = M_n$. We split the utility $u_{k,n}$ for decision maker n and product $k \in \mathcal{S}_n$ into a deterministic and a random component:

$$u_{k,n} = V_{k,n} + \epsilon_{k,n}. \tag{2.40}$$

Here, the vector $V_{\cdot,n}$ measures the systematic component of utility while $\epsilon_{\cdot,n}$ is a random disturbance. We assume that the distribution of ϵ is given by the joint density function $f(\epsilon_{1,n}, \dots, \epsilon_{M_n,n})$.

Customer n will choose the alternative that maximizes utility, in other words they will choose $k \in \mathcal{S}_n$ if and only if

$$\begin{aligned}
& u_{k,n} \geq u_{k',n} && \forall k' \in \mathcal{S}_n, k' \neq k && (2.41a)\\
\Leftrightarrow\quad & V_{k,n} + \epsilon_{k,n} \geq V_{k',n} + \epsilon_{k',n} && && (2.41b)\\
\Leftrightarrow\quad & V_{k,n} - V_{k',n} \geq \epsilon_{k',n} - \epsilon_{k,n}. && && (2.41c)
\end{aligned}$$

The probability of customer n choosing alternative 1 is then given by

$$\begin{aligned}
\mathrm{P}_n(1) &= \mathrm{P}(u_{1,n} \geq \max_{k \in \mathcal{S}_n} u_{k,n}) && (2.42a)\\
&= \mathrm{P}(\forall k \in \mathcal{S}_n \setminus \{1\} : u_{1,n} \geq u_{k,n}) && (2.42b)\\
&= \mathrm{P}(\forall k \in \mathcal{S}_n \setminus \{1\} : V_{1,n} - V_{k,n} \geq \epsilon_{k,n} - \epsilon_{1,n}) && (2.42c)\\
&= \int_{\epsilon_{1,n}=-\infty}^{\infty} \int_{\epsilon_{2,n}=-\infty}^{V_{1,n}-V_{2,n}+\epsilon_{1,n}} \cdots \\
&\quad \int_{\epsilon_{M_n,n}=-\infty}^{V_{1,n}-V_{M_n,n}+\epsilon_{1,n}} f(\epsilon_{1,n}, \dots, \epsilon_{M_n,n}) \,\mathrm{d}\epsilon_{M_n,n} \dots \mathrm{d}\epsilon_{2,n} \,\mathrm{d}\epsilon_{1,n} && (2.42d)
\end{aligned}$$

and analogously for the other alternatives.

2.3.1 Multinomial Probit

We will only give a very rough outline; a thorough treatment of multinomial probit (MNP) can be found in the book of Greene [52].

The MNP model assumes that the error term $\epsilon_{\cdot,n}$ for each decision maker n follows a multivariate normal distribution with mean 0 and variance-covariance matrix Σ. Because the cumulative distribution function (CDF) of the normal distribution cannot be expressed analytically, computation of the booking probabilities requires a numerical solution of the higher dimensional integral in the RHS of Eq. (2.42d).

The main advantages of MNP over other models is the fact that variation in customer taste variables can be incorporated naturally under certain conditions. Assume that utility is linear in both product and customer attributes:

$$\begin{aligned}
u_{k,n} &= \mathbf{x}_n^\top A q_k + \epsilon_{k,n} && (2.43a)\\
&= \bar{\mathbf{x}}_n^\top A q_k + \underbrace{(\hat{\mathbf{x}}_n^\top A q_k + \epsilon_{k,n})}_{\epsilon^*_{k,n}}. && (2.43b)
\end{aligned}$$

Here, the taste vector $\mathbf{x}_n = \bar{\mathbf{x}}_n + \hat{\mathbf{x}}_n$ of decision maker n is a random variable that can be decomposed into its expected value $\mathbb{E}[\mathbf{x}_n] = \bar{\mathbf{x}}_n$ and the deviation $\hat{\mathbf{x}}_n$ from this average. The vector of attributes associated with the k-th alternative is denoted by q_k, the matrix A describes the relationship between product attributes and customer preferences, and $\epsilon_{k,n}$ is again a random error term. If we assume that $\mathbf{x}_n$ (and thus $\hat{\mathbf{x}}_{k,n}$) and $\epsilon_{k,n}$ are normally distributed, then the combined error term $\epsilon^*_{k,n}$, which is a linear combination of these, is normally distributed as well. By construction it has mean zero. The variance-covariance matrix Σ_{ϵ^*} of $\epsilon^*_{\cdot,n}$ is given by

$$\Sigma_{\epsilon^*} = \Sigma_\epsilon + Q^\top A^\top \Sigma_{\mathbf{x}} A Q \tag{2.44}$$

where Σ_ϵ and $\Sigma_{\mathbf{x}}$ are the covariance matrices of $\epsilon_{\cdot,n}$ and $\hat{\mathbf{x}}_{\cdot,n}$ respectively and Q is the matrix of product attributes, consisting of the columns $q_1, \dots, q_{M_n}$.

Although some progress has been made, the high computational cost for the estimation of booking probabilities usually still outweighs the benefits. Therefore, MNP is rarely used in practice.

2.3.2 Multinomial Logit

In contrast to MNP, the multinomial logit (MNL) model can be handled very well analytically, which makes it much more suitable for large scale applications. On the other hand, MNL depends on much stronger restrictions and has the undesirable IIA property. Proofs for the following statements and a detailed description of the MNL model can be found in the book of Greene [52]. For an application to the travel sector see Domencich and McFadden [34].

Assume that for every decision maker n the error terms $\epsilon_{k,n}$ in Eq. (2.40) are

(1) independently distributed,

(2) identically distributed,

(3) Gumbel-distributed with a common location parameter η and scale parameter μ.

Here, (1) and (2), stating that the random error terms are IID, are very strong assumptions. Since we can always move constants into the systematic component $V_{k,n}$, the fact that for fixed n all $\epsilon_{k,n}$ share the same η does not pose any problems. However, not only are the error terms independent, the fact that they also share the same μ implies that all error terms will have the same variance.

Using these assumptions together with some basic properties of the Gumbel distribution and Eq. (2.42a) it is fairly straightforward to show that for every decision maker n and every alternative $k \in \mathcal{S}_n$:

$$\mathrm{P}_n(k) = \frac{e^{\mu V_{k,n}}}{\sum_{k' \in \mathcal{S}_n} e^{\mu V_{k',n}}}. \tag{2.45}$$

As a direct result one sees—by removing an arbitrary element from $\mathcal{S}_n$—that MNL satisfies the IIA property (see property 1).

When MNL is extended to a model with random taste parameters, the integral in Eq. (2.42d) cannot be solved analytically and thus the choice probabilities do not have a closed functional form.

Chapter 3

Capacity control

Arguably the most famous quote on the goal of Revenue Management is the following, taken from the 1987 American Airlines Annual Report [111]:

> "The objective of yield management is to maximize passenger revenue by selling the right seats to the right customers at the right prices."

Here, "right" of course means "right from the airline's perspective". In other words, the objective is to use the resources available as efficiently as possible by maximizing profit selling the limited stock that is available.

Revenue Management techniques can be applied in any situation where a finite, fixed inventory has to be sold out as profitable as possible. In the airline context, one tries to generate optimal revenue using all seats provided on all flights included in the schedule, while in the hotel or car rental businesses the limited good is the supply of rooms or cars respectively. Some other examples are department stores during clearance sales, theaters, or cloud computing providers selling hardware capacity. Because the application we are looking at in this work depends on some of the unique properties of the airline world, we will only cover the airline case. The interested reader can find detailed information about other applications of RM in the book of Talluri and van Ryzin [117, chapter 10].

In this chapter we give a summary of the different problem formulations and solution methods for the capacity control problem, which is the main focus of classic Revenue Management (RM). First, we will clarify the setting in which capacity control takes place and give an overview over the basic input data that is common to all approaches one can take to tackle the problem. We will then review the history of RM and list some of the most influential publications on the subject.

In this work we will solely focus on the standard approach to RM, where the problem is split into two major sub-problems: *demand forecasting* and *optimization*. A demand model estimated from past observations is used to create a forecast for future demand. This demand forecast is then used as input to mathematical optimization algorithms in order to determine revenue-optimal control parameters. We will cover both aspects in Sections 3.3 and 3.4 and will describe in detail some of the methods that are most commonly used in practice at the moment.

3.1 Problem definition

The capacity control problem is characterized by the following data:

- The *schedule*, consisting of a list of flights including the respective flight times, origin and destination. Each flight in the network is a *resource* with limited stock (capacity). We will denote the set of resources by $R = \{1, \dots, m\}$ and a single flight by $r \in R$.
- *Capacities* for each compartment on each flight. We denote the capacity of flight $r \in R$ by C_r.
- A set of booking classes or *products* $\mathcal{P} = \{1, \dots, M\}$, where each product $k \in \mathcal{P}$ is has fixed product characteristics and a fixed price.

- The *resource consumption matrix* $A \in \{0,1\}^{m \times M}$ that describes which products use which flights in the network. The entry $a_{r,k} = 1$ if and only if product k uses resource r and zero otherwise.
- The estimated yield for each booking class, measuring how much expected revenue a booking in the respective class will yield for the airline. This yield can differ from the price of the product from the customer's perspective, because not all that the customer pays is income for the airline. The yield is the product's price minus taxes, fees (e.g. airport fees, credit card fees) and marginal costs (e.g. distribution costs, fuel), and possibly increased in order to reflect expected revenue generated by selling additional services like extra baggage or lounge access. We denote the yield of product k by $\mathbf{y}_k$.
- The *booking horizon* or *selling period* $[0, T]$. Selling starts at time $t = 0$ and ends at the time of departure $t = T$, where remaining inventory is spoiled and becomes worthless to the seller.

In addition, the capacity control problem requires a model for customer demand. Different ways to model demand lead to slightly different version of the capacity control problems and therefore different solution algorithms, which are presented in the following sections.

The airline's goal is to maximize overall expected revenue over the course of the booking horizon. In order to maximize revenue, the airline must avoid spoilage (i.e. empty seats at the time of departure) as well as spill (high-value demand that is turned away because of limited remaining capacity) as much as possible, while exploiting the customers' willingness-to-pay at the same time. That is, one attempts to reserve enough capacity for high-value customers arriving at the end of the booking horizon, but at the same time, because an empty seat becomes worthless at departure, to sell out as many seats as possible.

The classic capacity control problem in RM is sometimes also called the *availability control* problem. It assumes that the airline can exercise control over the booking process by varying booking class *availability*, i.e. deciding which subset of products can be purchased at every time during the booking horizon. As bookings occur they are observed by the seller, who therefore gains information about the realization of the random demand process and can update controls accordingly.

An alternative to this classic approach is *dynamic pricing*, which uses varying prices instead of availability as a means of control over the booking process. In this case price—and therefore also the yields $\mathbf{y}_k$—of each product k is not fixed, but becomes a time-dependent control variable. While in theory dynamic pricing gives the airline finer control over the booking process, availability control is still the method used by almost all large network carriers today, mostly due to limitations inherent to the booking mechanisms implemented in the major Global Distribution Systems (GDS) that are currently in use (see Chapter 4).

In practice, many parameters of the capacity control problem can vary over time or can be uncertain. Among other things, capacities change due to aircraft changes, routes and flight times change due to schedule changes, and demand varies depending on competition or other external factors. In addition, cancellations and no-shows are frequent in practice, forcing airlines to regularly overbook their flights in order to avoid spoilage. As is common in RM literature unless one wants to address one of these topics specifically with the methods presented, we will omit all of these effects from our analysis and instead assume that all input data is given and fixed. Our analysis of the dynamic program in the following sections can be extendet to methods from the RM literature that incorporate cancellations and overbooking in the dynamic program, e.g. as presented by Sierag et al. [109].

3.2 History and literature

RM had its origins in the US, where between 1937 and 1978 air traffic was regarded as a public utility and highly regulated by the Civil Aeronautics Board (CAB), including the selection of routes and flight schedules and, in particular, fixed fares for each route.

Therefore, early RM methods were only concerned with the problem of efficiently managing capacity using overbooking. With the Airline Deregulation Act of 1978 control over fares was gradually handed back to the airlines. In the developing competitive market it quickly became clear that profitability was strongly linked to the ability to leverage the newly gained freedom in pricing decisions. However, the main focus remained on inventory control, now with availability control mechanisms added to the RM toolbox. The last decade of the twentieth century was characterized by a strong focus on O&D control. The pricing side of RM, dealing with customer behavior and—in particular— willingness-to-pay, was not incorporated into RM tools until the mid 2000s.

Inventory was initially controlled by means of static protection levels on flight level for each booking class. Protection levels were computed based on simple demand volume forecasts for each individual booking class. The first mathematical optimization method for the airline seat allocation problem was introduced by Littlewood in 1972 [78]. In 1987 Belobaba introduced the heuristic EMSR, which extends Littlewood's rule to the case of more than two booking classes [3]. EMSR and its numerous variants quickly became the industry standard and are still used today.

In contrast to these static control policies, dynamic policies take into account both the remaining time to departure and remaining capacity and can therefore react dynamically to the inherent volatility of the demand process. The corresponding dynamic optimization problem very naturally lends itself to a dynamic programming formulation, which first appeared in the RM literature in the early 1990s [74, 46] and has quickly gained popularity in industry practice.

In the late 1990s and early 2000s most large network carriers implemented O&D availability control schemes as replacements for the existing leg based systems in order to better handle network effects. Even with leg based forecast and optimization, O&D steering—e.g. bid price control—helped to account for the price differences between tickets sold on different markets and POSs. The transition to O&D forecasts allows to deal more accurately with external effects such as schedule changes or large peaks in demand during special events (e.g. holidays, sports events, congresses). Lastly, different heuristics for the computationally intractable stochastic network optimization problem were developed and quickly implemented in industry practice [117, sec. 3.3].

Through all of these developments, methods still worked under the assumption that demand is independent between booking classes, i.e. that fare restrictions like booking flexibility effectively separate the market into disjoint customer segments where each customer segment is only interested in exactly one booking class. As low-cost carriers entered the market with very simple fare structures and low prices, they forced the traditional network carriers to drop fare restrictions in order to stay competitive. In this scenario the independent demand model was not only theoretically unsuitable but has also been proven to have systematic deficiencies like the so–called *spiral down* effect in demand forecasting.

In the first decade of the 20th century RM research focused on the development of demand models that can capture customer choice behavior. Early models, like the *Q-forecast* model of Belobaba et al [5, 6], assumed a so-called fence-less fare structure without fare restrictions and thus were not directly applicable for network carriers, who still worked with partially restricted fares. The Q-forecast was extended to the *hybrid* forecast model, which aims to deal with price sensitivity as well as product awareness by splitting demand into so–called *priceable demand* and *yieldable demand*.

Even though demand models and forecasting techniques became much more complex, it turned out that the corresponding choice-based optimizations problem can be transformed to equivalent independent-demand problems using the *fare transformation* of Isler and Fiig [63, 42].

3.3 Demand modeling and forecasting

From the customers' point of view, booking is a two step process. The potential customer first requests availability of all relevant products from the provider and then—based on the set of alternatives they are offered—makes a decision which product to purchase or not to purchase anything at all. Therefore, one has to model customer arrival as well as customer choice. After discussing customer arrival first, in the rest of this section we will introduce the most prominent choice models in revenue management Estimation of model parameters based on observed bookings is generally done via pickup methods, such as exponential smoothing, or filter methods, for example a moving average.

3.3.1 Customer arrivals

Customer arrival is modeled through stochastic processes, most frequently using inhomogeneous Poisson processes. This strong assumption is justified if one assumes that all systematic effects on demand are known and randomness only arises from a large number of potential customers making independent choices. Under weaker assumptions, e.g. when there is additional unknown variance in the underlying population or if choices between individuals are correlated, more complex models such as semi-Markov processes have to be used.

For technical reasons the booking horizon is often discretized using a fixed grid of so–called Data Collection Points (DCPs). A simpler model that uses a finite number of random variables to describe the number of customer arrivals in each time interval can be derived from the stochastic process. The problem of estimating arrival rates on a continuous time scale is then reduced to the problem of estimating a finite number of distribution parameters.

The formulation as a time-continuous stochastic process seems much more natural, for various reasons. Each customer does indeed arrive at a specific point in time, and the order in which customers arrive potentially makes a difference in the result of their request. This is immediately reflected in an arrival process, while one has to model order of arrival explicitly if a fixed time interval with potentially multiple arrivals is considered. The approach also allows for a much larger degree of freedom and flexibility, for example by using a batch process in order to model group bookings.

On the other hand, unless the process is assumed to be homogeneous, it depends on time-dependent parameters. Without any assumptions on structure of these parameters as a function of time, this leads to an infinite dimensional space of potential models and makes the estimation problem practically infeasible. Therefore, given a fixed discretization of the booking horizon, the arrival rate is often assumed to be piecewise constant, piecewise linear or otherwise described by a parametric curve. Even then, for a generic stochastic process such as a Markov Arrival Process estimation is very complex and often depends on a-priori knowledge of some of the model parameters.

As a result, in practice one usually divides the booking horizon $[0, T]$ into a finite number of time slices and assumes that the number of customers arriving in each follows the same parametrized distribution. More precisely, given a partition of $[0, T]$

$$0 = t_1 < t_2 < \dots < t_{n+1} = T \tag{3.1a}$$

$$I_i = [t_i, t_{i+1}], \qquad \forall i = 1, \dots n \tag{3.1b}$$

one assumes that the number of customers $\mathbf{N}_i$ arriving in each interval i follows the same random distribution, parametrized by the vector α_i. The estimation problem is then reduced to the problem of estimating the α_i which, depending on the selected distribution, can be very simple.

> **Example 3.3.1** The most common model in practice uses an inhomogeneous Poisson process with rate
>
> $$\begin{aligned} \lambda \colon [0, T] &\to \mathbb{R} \\ t &\mapsto \lambda(t) \end{aligned}$$
>
> for customer arrival. Using the notation of Eq. (3.1), for each i the number of customers arriving in interval I_i is a Poisson distributed random variable $\mathbf{N}_i$ with expected value $\Lambda_i = \int_{t_i}^{t_{i+1}} \lambda(t)\,\mathrm{d}t$. Once the time discretization $t_1, \dots, t_{n+1}$ is fixed, only the finite length vector of parameters $(\Lambda_i)_{1 \le i \le n}$ needs to be estimated.
>
> However, the memorylessness property of the exponential distribution governing the time between arrivals leads to a fixed variance for $\mathbf{N}_i$, prohibiting the use of a Poisson process whenever variance is of particular interest.

Example 3.3.2 Although strictly illegitimate because its values are neither discrete nor non-negative, a normal distribution is often assumed for the $\mathbf{N}_i$, mainly because parameter estimation is then very simple and computationally efficient. Despite its drawbacks, the use of a normal distribution can be justified: The normal distribution with mean μ and standard deviation $\sqrt{\mu}$ is a good approximation of the Poisson distribution with mean μ, if μ is large, i.e. $\mu \gg 1$. In case the variance is high enough that the probability of negative arrivals becomes significant, a truncated normal distribution can be used instead to ensure non-negativity.

In the following we denote by $D(t)$ the *aggregate demand* or simply *demand* at time t, which—depending on whether we are considering a stochastic process or a fixed interval formulation—refers to the arrival rate at time t or the expected number of arrivals in the respective interval I_i.

Let $\mathcal{P} = \{1, \dots, M\}$ be the fixed set of products. Whenever a customer arrives, the airline can choose which of these products are made available for sale at that point in time. Therefore, each customer chooses from a subset of the full set of products, which, following the notation of Chapter 2, we denote by $\mathcal{S} \subseteq \mathcal{P}$. We assume that customers' choices are independent between each other. Then, given an availability $\mathcal{S} \subseteq \mathcal{P}$, aggregate demand $D_k(t, \mathcal{S})$ at time t for product $k \in \mathcal{P}$ is given by

$$D_k(t, \mathcal{S}) = D(t) d_k(t, \mathcal{S}), \tag{3.2}$$

where $d_k(t, \mathcal{S})$ is the probability of a random customer at time t booking product k given the set of choices $\mathcal{S}$. The booking probabilities obviously satisfy

$$\forall t \in [0, T], \forall k \in \mathcal{P} : k \notin \mathcal{S} \Rightarrow d_k(t, \mathcal{S}) = 0. \tag{3.3}$$

Using the above formulation, booking rates for each product depending on an offer set are decomposed into an offer-independent arrival rate and offer-dependent booking probabilities for each customer. Once the choice model is known, arrival rates can generally be estimated fairly easily using statistical methods for univariate time series.

In the following sections we will give a short overview over the most commonly used choice models in current airline RM practice. In order to simplify notation we will omit the potential dependence of parameters on the time t or the interval I_i.

3.3.2 Independent demand

Before the rise of low cost carriers and the related change in consumer choice behavior it was common practice to assume that the restrictions imposed on fares were strong enough to separate the market perfectly, causing demand to be independent between booking classes. In other words, in the independent demand model each customer requests availability of exactly one specific product $\mathcal{P}$, books the product if it is available and does not book at all otherwise.

In the corresponding choice model the customer $\mathbf{X}$ is a one dimensional discrete random variable with values corresponding to the customer's preferred product. Thus the booking probability $d_{\mathcal{P}}$ is independent from the availability of other products. The model is uniquely determined by the parameters $\mu_{\mathcal{P}}$, quantifying the probability that a random customer is interested in purchasing each product $\mathcal{P}$. Booking probabilities are given by

$$d_{\mathcal{P}}^{\text{indep}}(\mathcal{S}) = \begin{cases} \mu_{\mathcal{P}} & \text{if } \mathcal{P} \in \mathcal{S}, \\ 0 & \text{else.} \end{cases} \tag{3.4}$$

3.3.3 Q-Forecast

The Q-Forecast, introduced by Belobaba and Hopperstad [5, 6], is a forecasting model designed to fit the business model of low-cost carriers and assumes that all products are identical except for their price. Thus, given availability $\mathcal{S} \subseteq \mathcal{P}$, demand will only be nonzero for the product with the lowest price among the elements of $\mathcal{S}$, which we will denote by $\min(\mathcal{S})$.

The idea is to estimate demand for the cheapest booking class contained in $\mathcal{P}$, which at the time was called the Q-class in the considered product structure, and sell up potential from this class to the more expensive products. We can assume w.l.o.g. that every customer would purchase the Q-class if it is available, in other words $d_{\mathrm{Q}}(\mathcal{S}) = 1$ if $\mathrm{Q} \in \mathcal{S}$. The model assumes an exponentially distributed willingness-to-pay as the only customer attribute, which leads to the relation

$$d_{\mathcal{P}}^{\mathrm{Q}}(\mathcal{S}) = \begin{cases} e^{\lambda\left(1-\frac{f(k)}{f(\mathrm{Q})}\right)} & \text{if } k = \min(\mathcal{S}), \\ 0 & \text{else} \end{cases} \tag{3.5}$$

with the following parameters:

- Price elasticity of demand λ given by

$$\lambda = \frac{\ln(2)}{\mathrm{frat}_5 - 1}$$

  where frat_5 is the (time-dependent) fare-ratio at which 50% of the passengers are expected to sell up.

- Prices $f(\mathrm{Q})$ and $f(k)$ of classes Q and k respectively.

Initially the elasticity parameter λ had to be provided manually by the analyst. In recent years there have been efforts to obtain estimates from historical data [55].

3.3.4 Hybrid forecast

The fare structures that are widespread among network carriers today fit neither the assumptions of the independent demand model nor those for the Q-forecast: Fare restrictions are used to differentiate products, but cannot be expected to be strong enough to fully segment the market. The hybrid forecast model, introduced in 2004 by Boyd and Kallesen [14], aims to deal with these *semi-restricted* fare structures. It covers both price sensitivity as well as product awareness by splitting demand into two categories.

The share of customers who decide purely by price is called *priceable demand* and is expected to behave according to the Q-forecast model. The remaining customers are assumed to be purely product sensitive, and this *yieldable demand* is modeled according to the independent demand model.

Aggregate demand is then

$$D_k^{\mathrm{hybrid}} = D_k^{\mathrm{indep}} + D_k^{\mathrm{Q}} \tag{3.6}$$

and depends on all parameters present in both models.

3.4 Optimization

Depending on the customer model, different optimization methods can be used in order to compute an optimal or close to optimal control policy. A control policy is a set of rules that determine the set of alternatives each customer is offered at the time of their request. In the most general formulation the airline's choice may depend on the complete selling history so far. However, if the demand process is memory-less—for example if it is a Poisson process–the optimal controls only depend on remaining inventory.

3.4.1 Notation

Let $[0, T]$ be the booking horizon, where $t = 0$ is the start of the booking period and $t = T$ is time of departure. Let m denote the number of different resources (i.e. flight legs) and $C \in \mathbb{N}^m$ the vector of initial capacities. Let $\mathcal{P} = \{1, \ldots, M\}$ be the set of products. Each product $k \in \mathcal{P}$ is defined by its yield $\mathbf{y}_k$ and the incidence vector A^k, which is the k-th column of the *resource consumption matrix*:

$$A = (a_{r,k})_{\substack{r=1,\ldots,m \\ k=1,\ldots,M}} \tag{3.7}$$

Here $a_{r,k}$ denotes the number of units of resource r that are consumed by one unit of product k. In the airline case the entries of A are binary variables, indicating if the travel path, or ODI, associated to a product includes the respective flight leg or not. Reordering the products, we can assume w.l.o.g. that the products are ordered decreasingly by their yield so that

$$\mathbf{y}_1 \geq \mathbf{y}_2 \geq \cdots \geq \mathbf{y}_M.$$

In this section we will always assume that demand is independent between products. This assumption is justified, because a general discrete choice model can be transformed into an—for optimization purposes—equivalent independent demand model on a set of virtual products. This transformation is described in detail in Section 3.5. The transformed model is equivalent to the original one in the sense that an optimal control policy for the original problem can be generated in a straightforward way from an optimal solution for the transformed problem.

Customer arrival can therefore be modeled using a multivariate stochastic arrival process

$$\mathbf{N}(t) = (\mathbf{N}_k(t))_{k\in\mathcal{P}} \tag{3.8}$$

with *mutually independent* components $\mathbf{N}_k(\cdot)$, i.e. demand for each product $k \in \mathcal{P}$ is modeled via the stochastic process $\mathbf{N}_k(\cdot)$, whose realization is independent of the realization of demand for the other products. Here, $\mathbf{N}_k(t)$ denotes the number of customers requesting (virtual) product k up to and including time t, i.e. in the time interval $(0, t]$. In this work we do not consider external competition or other potential substitute products offered by the airline itself. We therefore call $\mathbf{N}$ the *demand process*. Due to the way airline tickets are sold and the nature of current reservation systems, this process has to be considered as unobservable for the analyst.

The airline exercises control over the booking process by deciding on a set of available (virtual) products at every point in time. The airline's varying offer can be represented by a multivariate continuous time stochastic process $\mathbf{S}(t)$, called the *offer process*, taking values in the set $\{0,1\}^M$, where

$$\mathbf{S}_k(t) = \begin{cases} 1, & \text{if product } k \text{ is available at time } t \\ 0, & \text{else.} \end{cases} \tag{3.9}$$

In the following we identify a value $\mathcal{S} \in \{0,1\}^M$ with the set of available products $\{k \in \mathcal{P} \mid \mathcal{S}_k = 1\}$. Because the airline's offer set usually depends on the booking history, the processes $\mathbf{S}$ and $\mathbf{N}$ are not independent. Likewise, the components of $\mathbf{S}$ are not mutually independent. On the contrary, we will show later that, under the independent demand assumption, an optimal policy is nested in the sense that, if a certain virtual product is available at time t, then all higher valued virtual products are available as well.

Combining the demand and offer processes we can form the *booking process* $\mathbf{B}$, where $\mathbf{B}_k(t)$ denotes the number customers who purchased product k up to and including time t. An arrival in the demand process leads to an arrival in the booking process if and only if the respective product is available at that time. The booking process

$$\mathrm{d}\mathbf{B}_k(t) = \begin{cases} \mathrm{d}\mathbf{N}_k(t), & \text{if product } k \text{ is available at time } t, \\ 0, & \text{else} \end{cases} \tag{3.10a}$$

$$\Leftrightarrow \mathbf{B}_k(t) = \int_0^t \mathbf{S}_k(t)\,\mathrm{d}\mathbf{N}_k(t) \tag{3.10b}$$

for every product $k \in \mathcal{P}$ therefore again a Poisson process. With the slight abuse of notation

$$\mathbf{B}(t) = \int_0^t \mathbf{S}\,\mathrm{d}\mathbf{N} \tag{3.11}$$

we can formulate the classic airline revenue management optimization problem as the stochastic optimal control problem

$$\max_{\mathbf{S}} \quad \mathbb{E}\left[\mathbf{y}^\top \mathbf{B}(T)\right] \tag{3.12a}$$

$$\text{subject to} \qquad A\mathbf{B}(T) \leq C \qquad \text{a.s.} \tag{3.12b}$$

where A is the resource consumption matrix and C is the vector of initial capacities.

In theory, it is sufficient to just solve Eq. (3.12) once at the beginning of the booking horizon and then apply the optimal control process over the course of the booking period. In practice, a re-optimization is performed at certain times during the booking horizon in order to update the solution depending on additional information that might have become available in the meantime. All of these problems are structurally equivalent to the one above. If demand is assumed to be memoryless, i.e. if the observed booking history does not give us any information about the future of the demand process $\mathbf{N}$, then—if there is no other external source of information—the assumptions about future demand have not changed. Therefore the optimal solution will be the same as the restriction of the original optimal solution to the remaining part of the booking horizon. Note that this is not necessarily the case for heuristic solutions.

3.4.2 Control mechanisms

There are a number of different ways to implement control strategies in practice, each depending on different assumptions about demand, different data, and specially tailored optimization algorithms. *Control schemes* or *control mechanisms* are rules that specify which products should be made available based on the history of the booking process. Of course, ideally one would prefer to use the history of the demand process. But because demand cannot be observed directly, one has to resort to working with the booking process, which is completely observable for the airline. For practical reasons, the airline's decision will often not take into account the full history of the booking process but rather a set of derived figures describing certain features of the process history.

In the following definition we will use the notion of a *system state*, which contains information about the situation the airline is in at a specific point in time during the booking horizon. The state depends on the initial conditions, for example initial capacities, and events that occur during the booking horizon. The most frequent events are

bookings, which change the situation by consuming inventory,

cancellations, which free up previously reserved inventory, and

changes of reservation, which can be regarded as a combination of both.

Other events occur less often but have a stronger impact, such as a change of aircraft or the cancellation of a flight. Because these strong events are rare and therefore hard to forecast, in practice they are often ignored in short term revenue management and, when they do occur, dealt with by re-optimizing the whole system under the newly changed conditions. We will therefore ignore these types of events in the following.

The state of the system is denoted by $x \in X$, where the state space X is the set of all feasible states and usually contains information about one of the following:

Full booking history The control decisions are made based on the booking process that was observed so far, i.e. the number of bookings for all booking classes and, potentially, the chronology of these bookings.

Set of existing bookings Demand in RM is often modeled as being memoryless. Under this assumption, the history of the booking process does not contain any information about the demand-to-come. However, the set of existing bookings may carry information about the probability distribution of future cancellations, which are inherently dependent on the booking history. In this kind of model, the state of the system at time t is described solely by the number of existing bookings for each product. The process that led to this state is irrelevant.

Remaining inventory If cancellations and no-shows are ignored or handled separately and demand is assumed to be memoryless, the history of the demand process only influences the future through remaining capacity. It is therefore sufficient to base the availability decision on the vector of remaining capacities on each flight leg together with the current time to departure.

Definition 3.4.1 (Feasible actions, availability control scheme)
Let $\mathcal{P} = \{\mathfrak{p}_1, \dots, \mathfrak{p}_M\} \subseteq \mathbf{P}$ be a set of products. A *feasible action* is a subset of $\mathcal{P}$, which the airline could choose to offer for sale at any given time. Let $\mathbb{S} \subseteq \wp(\mathcal{P})$ be the set of feasible actions. An *availability control scheme* or *control policy* $\mathcal{S}$ is a function

$$\mathcal{S}: [0,T] \times X \to \mathbb{S} \tag{3.13a}$$

$$(t,x) \mapsto \mathcal{S}(t,x), \tag{3.13b}$$

mapping a pair of time t and system state x to a feasible action.

A *static* availability control scheme is one in which the offer set does not depend on the time t but only on the state of the system x

$$\mathcal{S}: X \to \mathbb{S} \tag{3.14a}$$

$$x \mapsto \mathcal{S}(x). \tag{3.14b}$$

Remark 3.4.2 In contrast to dynamic control schemes, static policies suffer from the fact that they cannot react dynamically to variance in the demand process. In order to compensate for this it is common practice to re-optimize frequently during the booking horizon, thus updating the control policy to one that is suitable for the changed situation. This way, static control schemes and static optimization can be used to achieve a semi-dynamic control policy.

Remark 3.4.3 In practice availability may also depend on customer– or request–specific information such as frequent traveler status or POS. For optimization purposes this kind of discrimination can be modeled using additional virtual products. For example, if the airline wishes to grant different availability for a certain product to a selected POS, one would introduce a separate product that can only be purchased at this specific POS, and therefore has reduced demand, but is otherwise equivalent to the original product.

Control mechanisms for the single-leg problem
We will first describe a number of control mechanisms for the single-leg problem that are popular in the RM literature and industry practice:

Partitioned booking limits are *static* controls that allocate a number of seats L_k to each booking class k. The system state is described by the number of existing bookings x per booking class k and availability is given to a product as long as the number of existing bookings is below the booking limit:

$$\mathcal{S}(x) = \{k \in \mathcal{P} \mid L_k > x_k\}. \tag{3.15}$$

Nested booking limits work similarly, but avoid one of the major problems of partitioned booking limits. Assume that, because of high demand, the reserved inventory for a certain high value product has been sold out, and that physical capacity is still available but reserved for a different (lower valued) product. In this situation it is sensible to make this capacity available to all products that will generate more revenue while consuming the same resources.

Based on the assumption that products are ordered with decreasing yields, we assign booking limits $L_1 \geq L_2 \geq \cdots \geq L_M$, where L_k states how many bookings we wish to accept for classes $k, \dots, M$. As long as there is remaining capacity, it is never correct to reject a booking request for the most expensive class 1. The corresponding booking is therefore always equal to the initial capacity: $L_1 = C$.

In the simplest case, the system state x again counts the number of existing bookings in each class. The booking process is controlled so that for each booking class $k \in \mathcal{P}$ the constraint

$$\hat{x}_k := \sum_{k'=k}^{M} x_{k'} \leq L_k \tag{3.16}$$

is satisfied throughout the whole booking period. In other words, a request for product k is accepted as long as none of the constraints that contain x_k are satisfied with equality:

$$\mathcal{S}(x) = \{k \in \mathcal{P} \mid \forall k' = 1, \dots, k : L_{k'} > \hat{x}_{k'}\}. \tag{3.17}$$

In some cases, depending on the demand model, it is easier to keep track of the remaining booking limit instead of the number of bookings together with slightly more complicated update rule when a booking occurs [117, sec. 2.1.1].

Protection levels are a different view on the idea of booking limits. Instead of specifying how many bookings we wish to accept for each class, we decide how many seats should be reserved for higher-value products. If $\tilde{L}_k$ denotes the number of seats that are to be reserved for classes $1, \dots, k$, the (nested) protection levels $\tilde{L}_1, \dots, \tilde{L}_{M-1}$ can be computed from nested booking limits $L_2, \dots L_M$ as

$$\tilde{L}_k = C - L_{k+1} \tag{3.18}$$

for every $k = 1, \dots, M-1$.

Bid prices are estimates of the marginal value that one unit of a resource has to the airline. Bid price controls are usually used under the assumption that demand is memoryless. In this case, the bid price for the next seat to be sold depends on the time to departure and on the remaining capacity. For any time $t \in [0, T]$ and remaining capacity $c \in \mathbb{N}$ consider the single-leg version

$$\max_{\mathbf{S}} \quad \mathbb{E}\left[\mathbf{y}^{\top} \int_t^T \mathbf{S} \, \mathrm{d}\mathbf{N}\right] \tag{3.19a}$$

$$\text{subject to} \quad \sum_{k \in \mathcal{P}} \int_t^T \mathbf{S}_k \, \mathrm{d}\mathbf{N}_k \leq c \qquad \text{a.s.} \tag{3.19b}$$

of the availability control problem Eq. (3.12) . Let

$$V\,[0, T] \times \mathbf{C} \to \mathbb{R} \tag{3.20a}$$

$$(t, c) \mapsto V(t, c) \tag{3.20b}$$

be the real valued function mapping a time t and remaining capacity c to the expected future revenue $V(t, c)$, which is equal to the optimal objective function value of Eq. (3.19).

The marginal value of one unit of capacity depending on the time and current inventory is then given by the function

$$\pi \colon [0, T] \times \{1, \dots, C\} \to \mathbb{R} \tag{3.21a}$$

$$(t, c) \mapsto \pi(t, c) := V(t, c) - V(t, c-1), \tag{3.21b}$$

and can be interpreted as the opportunity cost associated with losing one seat.

If the current bid price π—or a good estimate—is available at the time of a booking request, it is intuitive to offer a product k to the customer only if the expected yield $\mathbf{y}_k$ associated with the product exceeds the estimated opportunity cost, i.e. if

$$\mathbf{y}_k \geq \pi(t, c). \tag{3.22}$$

If demand is independent between (virtual) products, the expected revenue from making a product available is pure incremental revenue and therefore the necessary condition is also sufficient in an optimal control policy. In other words, under an independent demand model a request for product k is accepted if and only if Eq. (3.22) holds.

Simple optimization methods generate constant bid prices π, which lead to constant booking class availability, or constant bid price vectors $(\pi_c)_{1 \leq c \leq C}$, which lead to static control schemes that are equivalent to nested booking limits.

Most advanced optimization methods compute time-dependent bid prices which are then used during the booking horizon to apply the control strategy described above:

$$\mathcal{S}(t, c) = \{k \in \mathcal{P} \mid \mathbf{y}_k \geq \pi(t, c)\}\,. \tag{3.23}$$

Remark 3.4.4 From an optimal control standpoint, it is more intuitive to write the value function as a vector valued function

$$V: [0,T] \to \mathbb{R}^m \tag{3.24a}$$
$$t \mapsto (V_1(t), \dots, V_m(t))^\top \tag{3.24b}$$

and treat capacity, which is an index to the discrete set of possible states of the system, as a subscript of the vector $V(t)$ rather than as an argument to V, and likewise for the bid price function π.

In order to be consistent with the classic RM literature we will use the notation from Eq. (3.20a) throughout the remainder of this overview chapter. For the detailed analysis of the dynamic program in Chapter 5 we will change to the more convenient notation of Eq. (3.24).

Control mechanisms for the network problem

The most common control strategies for the network problem are generalizations of the single-leg methods:

Booking limits: In a network with hundreds of resources and thousands of products on a large number of travel paths, partitioned booking limits for all booking classes are impractical, because the capacity of one flight leg might have to be divided between hundreds of products that require a seat on this leg. As the number of products increases, many products will have booking limits of zero and can therefore never be booked. At the same time, due to the randomness of demand, many products with positive booking limits will not be booked, leading to empty seats.

Nested limits, on the other hand, are not easy to implement either, mainly because the right nesting order of products is not always clear. A cheap product using only one flight leg might have a higher net profit than a more expensive product that consumes additional resources.

It is common practice to use so called *prorating* schemes, which split up the fare of each product between all corresponding flight legs according to certain rules, for example proportionally to the length of each flight leg. In order to reduce the number of booking classes, products are clustered based on their prorated fares and mapped to *virtual booking classes*, which are then used in the optimization and for availability control.

Bid prices: In the network case, the opportunity cost of selling a product depends on remaining inventory as well as the set of resources consumed by the product in question. It is again derived from the expected achievable future revenue. Let

$$V: [0,T] \times \mathbf{C} \to \mathbb{R} \tag{3.25a}$$
$$(t,c) \mapsto V(t,c) \tag{3.25b}$$

be the value function, where $V(t,c)$ is the optimal objective function value of Eq. (3.12). Here, the state space $\mathbf{C} = \times_{r=1}^{m} \{0, \dots, C_r\}$ consists of all possible combinations of remaining inventory on each leg.

In order to simplify notation we set the expected revenue to $-\infty$ for infeasible states, i.e. whenever at least one component of c is negative:

$$V(t,c) := -\infty \qquad \forall c \in \mathbb{N}^m : \exists 1 \le r \le m : c_r < 0. \tag{3.26}$$

Analogously to the single-leg case, the opportunity cost for selling one unit of a certain product at any given time is the difference in expected revenue due to the change of remaining inventory that is caused by the sale. Therefore, the bid price for product k at time $t \in [0,T]$ given remaining capacity c is computed as

$$\pi(t,c,A^k) := V(t,c) - V\left(t, c - A^k\right), \tag{3.27}$$

where $A^k = (a_{1,k}, \dots, a_{m,k})^\top$ is the k-th column of the resource consumption matrix, indicating the number of units of each resource consumed by one unit of product k (see Eq. (3.7)).

The exact bid price control scheme for the network problem is the straightforward extension of the single-leg case:

$$\mathcal{S}(t,c) = \left\{k \in \mathcal{P} \mid \mathbf{y}_k \geq \pi(t,c,A^k)\right\}. \tag{3.28}$$

Note that, due to Eq. (3.26), the bid price for a product k will be infinite whenever there is insufficient remaining capacity to accommodate a customer purchasing product k, i.e. if $A^k \geq c$ does not hold component-wise.

In this formulation, the bid price that is used to determine a product's availability theoretically depends on the remaining capacity of flight legs that are—if at all—only remotely related to the corresponding flight path. In order to reduce complexity, simplified heuristic control schemes are often used in practice. The most common bid price control scheme is derived in two steps.

First, in order to avoid having to compute an individual bid price for each flight path, the value function is replaced by a linear approximation around the current point in the state space. Viewing the bid price as a kind of discrete derivative of the value function w.r.t. a change in capacity, this is equivalent to the assumption that the opportunity cost of one seat on a given flight is constant in a neighborhood of the current remaining capacity c. With A^k as above and e_r denoting the r-th unit vector, we use the approximation

$$\pi\left(t,c,A^k\right) \approx V(t,c) - V\left(t,c-A^k\right) \tag{3.29}$$

$$= \nabla V\left(t,c\right) A^k, \tag{3.30}$$

where

$$\nabla V(t,c) := (\pi(t,c,e_1),\ldots,\pi(t,c,e_m)) \tag{3.31}$$

is the discrete gradient of the value function w.r.t. capacity.

With this linearization, approximate bid prices for each product can be easily computed from the vector valued bid price function

$$\pi\colon [0,T] \times \mathbf{C} \to \mathbb{R}^m \tag{3.32a}$$

$$(t,c) \mapsto \pi(t,c) := \left(\nabla V(t,c)\right)^\top, \tag{3.32b}$$

where the bid price $\pi(t,c)$ is a vector of length m containing the marginal value of one unit of capacity for each leg. The corresponding control scheme is given by

$$\mathcal{S}(t,c) = \left\{k \in \mathcal{P} \mid \mathbf{y}_k \geq \pi^\top(t,c)A^k\right\}. \tag{3.33}$$

Because the state space $\mathbf{C}$ grows exponentially with the number of flight legs m, the problem is decomposed further in a second heuristic approximation. By assuming that for each leg r the current bid price $\pi_r(t,c)$ only depends on c_r but not on the remaining capacity on all other legs, the vector-valued bid price function Eq. (3.32a) is replaced by a collection of maps

$$\pi_r\colon [0,T] \times \{1,\ldots,C_r\} \to \mathbb{R} \tag{3.34a}$$

$$(t,c_r) \mapsto \pi_r(t,c_r), \tag{3.34b}$$

for every $r = 1,\ldots,m$, where each map is defined on a one-dimensional state space.

Note that each such bid price function π_r is of the same form as the bid price function for the single-leg problem (see Eq. (3.21)). Therefore, approximate bid prices of this kind are usually computed by heuristically decomposing the network problem into suitably transformed single-leg problems that can then be solved using exact methods (see Section 3.4.4).

Given such a set of bid price functions, product availability is determined by the control rule

$$\mathcal{S}(t,c) = \left\{k \in \mathcal{P} \mid \mathbf{y}_k \geq \sum_{r=1}^{m} a_{r,k}\pi_r(t,c_r)\right\}. \tag{3.35}$$

3.4.3 Single-leg optimization

In this section we will introduce the most commonly used optimization algorithms for the single-leg problem, which assumes that there is only one flight leg with fixed capacity $C \in \mathbb{N}_{\geq 0}$ and each product requires exactly one unit of capacity.

Littlewood's rule

The fist optimization algorithm for the RM inventory control problem was introduced by Littlewood in 1972 [78]. Assume that there are two products $k = 1, 2$ with yields $\mathbf{y}_1 > \mathbf{y}_2$. Demand is described by independent discrete random variables $\mathbf{N}_1, \mathbf{N}_2 \geq 0$ that represent the number of requests for each class. The model assumes that demand for class 2 arrives strictly before demand for class 1. The problem is now to determine the optimal protection level for class 1, or equivalently, the optimal booking limits for class 2.

The solution can be derived easily by comparing the expected revenues associated with the *accept* and *reject* decisions for each class 2 request. Let $c \in \{1, \ldots, C\}$ be the remaining capacity.

- If the airline **accepts** a request for class 2, it will gain $\mathbf{y}_2$ and lose one unit of capacity.
- If the airline **rejects** the request and instead reserves the seat for class 1, it will sell the seat later and collect revenues of $\mathbf{y}_1$ if demand for class 1 is sufficiently high to use up all of the remaining capacity, otherwise the seat is spoiled. Thus, rejecting the request leads to an expected revenue of
$$\pi(c) = \mathrm{P}(\mathbf{N}_1 \geq c)\mathbf{y}_1, \tag{3.36}$$
which is the so-called *expected marginal seat revenue.*

Therefore, the decision to accept the request is favorable if and only if

$$\mathbf{y}_2 \geq \pi(c). \tag{3.37}$$

This control scheme, called *Littlewood's Rule*, is a static control policy that can be interpreted in two different ways.

Bid Price. The value $\pi(c)$ is the marginal opportunity cost of the c-th unit of capacity or, in other words, a (static) bid-price. The decision rule Eq. (3.37) is just Eq. (3.23) together with the fact that product 1 is always available as long as there is remaining capacity.

Booking Limit Clearly the right-hand-side of Eq. (3.36) is monotonically decreasing in c and $\lim_{c \to \infty} \pi(c) = 0$. Therefore the protection level
$$\tilde{L} = \max\{c \mid \pi(c)\mathbf{y}_1 > \mathbf{y}_2\} \tag{3.38}$$
is well-defined and Littlewood's Rule is a policy reserving $\tilde{L}$ seats for product 1 or, equivalently, setting a booking limit of $L := C - \tilde{L}$ for product 2.

EMSR

The first optimization methods for the single-leg problem with more than two products that were widely used in the airline industry were from the family of EMSR methods. EMSR stands for *expected marginal seat revenue*, which is essentially a different—and more precise—name for what is now widely known as a bid price. All EMSR methods are heuristics that share the idea of using Littlewood's rule to compute an optimal solution for one re multiple derived two-class problems and computing protection levels from these solutions.

EMSR-a

EMSR-a was introduced by Belobaba in his 1987 PhD thesis [3] and is an extension of Littlewood's Rule to the case of $M > 2$ booking classes. Ordering the classes by their expected yield such that $\mathbf{y}_1 \geq \mathbf{y}_2 \geq \cdots \geq \mathbf{y}_M$, we again assume that demand arrives in strict low-before-high order and represent the number of requests for each class by independent discrete random variables

$\mathbf{N}_1, \ldots, \mathbf{N}_M$. The goal is now to compute protection levels $\tilde{L}_k$ for each class $k = 1, \ldots, M-1$. The idea behind EMSR-a is to apply Littlewood's Rule to all pairs of booking classes and construct protection levels from the results.

Consider a fixed class $k+1$. We compute the protection level $\tilde{L}_k$ that determines how many seats to reserve for the higher-valued classes $1, \ldots, k$, as follows: For each $k' = 1, \ldots, k$ we use Littlewood's Rule to obtain a booking limit $\tilde{L}_{k'}^{k+1}$, which is the number of seats that are reserved for the higher class k' in the two-class problem that only consists of $k+1$ and k'. EMSR-a protection levels for the original problem are then computed as

$$\tilde{L}_k = \sum_{k'=1}^{k} \tilde{L}_{k'}^{k+1}. \tag{3.39}$$

It is fairly easy to prove that these protection levels are not optimal. This is mostly due to the fact that the opportunity cost of capacity occurring in Littlewood's Rule heavily depend on the variance that is assumed for the demand of the higher-valued classes. The true bid-price includes a statistical averaging effect that arises from aggregating demand over multiple booking classes. When considering all these classes separately, one ignores this effect, which leads to very restrictive controls. The reader is referred to the book of Talluri and van Ryzin [117, sec. 2.2.4.1] for an example and a more detailed discussion of the matter.

EMSR-b

EMSR-b is another heuristic method using the concept of expected marginal seat revenues. It depends on the same assumptions as EMSR-a and the idea is again to derive protection levels using Littlewood's Rule. However, in order to avoid the pitfalls of EMSR-a, instead of aggregating the protection levels, one aggregates demand. Again, we consider a fixed booking class $k+1$ and wish to compute a protection level $\tilde{L}_k$ for the classes $1, \ldots, k$. We represent the collection of classes $1, \ldots, k$ by a virtual booking class with demand

$$\tilde{\mathbf{N}}_k = \sum_{k'=1}^{k} \mathbf{N}_{k'}.$$

The yield for the virtual class is estimated as the average of the individual yields weighted with expected demand:

$$\tilde{\mathbf{y}}_k = \frac{\sum_{k'=1}^{k} \mathbf{y}_{k'} \mathbb{E}\left[\mathbf{N}_{k'}\right]}{\sum_{k'=1}^{k} \mathbb{E}\left[\mathbf{N}_{k'}\right]}.$$

The booking limits are then computed by applying Littlewood's Rule to this virtual product and product $k+1$. With Eq. (3.38) and Eq. (3.36), we obtain the protection levels

$$\tilde{L}_k = \max\left\{c \mid \mathrm{P}(\tilde{\mathbf{N}}_k \geq c)\tilde{\mathbf{y}}_k > \mathbf{y}_{k+1}\right\}. \tag{3.40}$$

There are several inaccuracies in the EMSR methods: Littlewood's Rule for the two-class problem only yields an optimal policy if the very strong *low before high*-assumption is satisfied. In practice, although customers tend to approximately act this way, the assumption is never fully correct. In addition, aggregation of protection levels (EMSR-a) or demand and yields (EMSR-b) is a very simple approximation of the complex stochastic interactions that occur in reality. However, studies have shown that both methods perform reasonably well in practice, with EMSR-b often losing less that 0.5% of revenue compared to an optimal solution [4, 99].

Dynamic programming

The static policies and optimization methods presented so far are only optimal if the demand model satisfies the very strong assumption that demand arrives in strict low-before-high order. The dynamic programming formulation of the inventory control problem does not depend on this assumption. Dynamic programming in RM was first introduced in 1994 by Talluri and van Ryzin [46], who analyze a single resource continuous-time dynamic pricing problem and, as a special case, a problem with a finite number of fixed price points.

Dynamic programming has since become the de facto standard in dynamic optimization. In this section we quickly review the standard formulation of the single resource problem. An extension to the network case is presented in Section 3.4.4.

The most common formulation of the single-leg dynamic program (DP) both in scientific publications and in industry practice uses a fixed time discretization, partitioning the booking horizon $[0,T]$ into intervals $I_1,\dots,I_n$ with

$$\begin{aligned} &0 = t_0 < t_1 < \dots < t_n = T && \\ &I_i = [t_{i-1}, t_i] && \forall i = 1,\dots n \\ &h_i = t_i - t_{i-1} && \forall i = 1,\dots n, \end{aligned}$$

where in each time step i the step size h_i is small enough such that the probability of multiple arrivals during the interval I_i is negligible. Let $\mathcal{P} = \{1,\dots,M\}$ be the set of products, and for each product k let $\mathbf{y}_k$ denote the expected yield. Expected demand for product k during interval I_i is given by booking probabilities $d_{k,i}$ with

$$\sum_{k\in\mathcal{P}} d_{k,i} \ll 1 \qquad \forall i = 1,\dots,n.$$

Again denote by $V(t,c)$ the expected value that a remaining capacity of c has to the airline in time step t. Remaining inventory is spoiled at the time of departure, and no future revenue can be generated once the capacity is full. Therefore, we have the boundary conditions

$$V(T,c) = 0 \qquad \forall c \in \{1,\dots,C\} \tag{3.41a}$$

$$V(t,0) = 0 \qquad \forall t \in [0,T]\,. \tag{3.41b}$$

Let $\mathcal{S}$ be a dynamic control scheme, defining the offer sets $\mathcal{S}(t,c)$ for every time step t and remaining capacity c. Assuming that at most one request will arrive in each time step, the value function satisfies the recursion

$$V(t_{i-1},c) = V(t_i,c) + \sum_{k\in\mathcal{S}(t_i,c)} d_{k,i}\left[\mathbf{y}_k + V(t_i,c-1) - V(t_i,c)\right] \tag{3.42}$$

for every capacity c and time-step i. Here, the second summand in the RHS of Eq. (3.42) is the gain in expected revenue with increasing time to departure and can be interpreted as follows: For each available product $k \in \mathcal{S}(t_i,c)$, the net expected revenue is the product of the respective booking probability and the product's current *margin*, which is the yield $\mathbf{y}_k$ reduced by the opportunity cost $V(t_i,c) - V(t_i,c-1)$ of losing one unit of capacity at the current state.

The dynamic availability control problem is concerned with optimizing expected revenue at the beginning of the booking horizon and initial capacity over the set of feasible dynamic availability control schemes:

$$\max_{\mathcal{S}} V(0,C) \tag{3.43}$$

This problem has an optimal substructure, and can be written as the dynamic program

$$V(t_{i-1},c) = \max_{\mathcal{S}(i,c)} \left\{ V(t_i,c) + \sum_{k\in\mathcal{S}(i,c)} d_{k,i}\left[\mathbf{y}_k + V(t_i,c-1) - V(t_i,c)\right] \right\} \tag{3.44}$$

with the boundary conditions Eq. (3.41).

Continuous time formulation

The DP can also be formulated in a continuous-time version, which describes the value function as the unique solution to an ODE. For this purpose we will now switch the notation for the value function V to the one in Eq. (3.24), where $V_c(t)$ denotes expected future revenue given time t and remaining capacity c.

Demand is modeled as an inhomogeneous multivariate Poisson process with demand rate

$$\lambda\colon [0,T] \to \mathbb{R}^M \tag{3.45}$$

$$t \mapsto \lambda(t), \tag{3.46}$$

where λ_k is the demand rate for product k. The value function is now the unique solution of the linear ODE

$$\dot{V}_c(t) = -\max_{\mathcal{S}\in\mathbb{S}} \sum_{k\in\mathcal{S}} \lambda_k(t)\,[\mathbf{y}_k - \pi_c(t)] \qquad \forall 1 \le c \le C \tag{3.47a}$$

$$\pi_c(t) = \begin{cases} V_1(t) & \text{if } c = 1, \\ V_c(t) - V_{c-1}(t) & \text{else} \end{cases} \tag{3.47b}$$

$$V_c(T) = 0 \qquad \forall 1 \le c \le C, \tag{3.47c}$$

where $\pi_c(t)$ is the bid price at time t and capacity c. A detailed description of the continuous-time DP is given in Chapter 5.

Remark 3.4.5 The discrete-time DP in Eq. (3.44) is equivalent to solving Eq. (3.47) numerically using an explicit Euler method with fixed step sizes $h_1, \ldots, h_n$, where the maximum in the RHS of Eq. (3.47a) is taken at the end of each interval I_i.

In the following we will only use the continuous time formulation.

Choice-based Dynamic Program

In a more general formulation one allows the demand rate for each product to depend on the availability of other products, in other words the demand rate is a function

$$\lambda\colon \mathbb{S} \times [0,T] \to \mathbb{R}^M \tag{3.48a}$$

$$(\mathcal{S}, t) \mapsto \lambda(\mathcal{S}, t) \tag{3.48b}$$

mapping a pair of an offer set $\mathcal{S}$ and a time t to a vector of demand rates that naturally satisfies

$$\lambda_k(\mathcal{S}, \cdot) \equiv 0 \qquad \forall k \notin \mathcal{S}, \tag{3.49}$$

meaning that demand is zero for products that are unavailable.

Instead of Eq. (3.47) one then obtains the *choice-based dynamic program*

$$\dot{V}_c(t) = -\max_{\mathcal{S}\in\mathbb{S}} \{\mathbf{R}(\mathcal{S}, t) - \mathbf{D}(\mathcal{S}, t)\pi_c(t)\} \qquad \forall 1 \le c \le C \tag{3.50a}$$

$$\pi_c(t) = \begin{cases} V_1(t) & \text{if } c = 1, \\ V_c(t) - V_{c-1}(t) & \text{else} \end{cases} \tag{3.50b}$$

$$V_c(T) = 0 \qquad \forall 1 \le c \le C, \tag{3.50c}$$

where

$$\mathbf{R}(\mathcal{S}, t) = \sum_{k\in\mathcal{S}} \lambda_k(\mathcal{S}, t)\mathbf{y}_k \tag{3.51a}$$

$$\mathbf{D}(\mathcal{S}, t) = \sum_{k\in\mathcal{S}} \lambda_k(\mathcal{S}, t) \tag{3.51b}$$

are the *total revenue rate* and *total demand rate* associated with the offer set $\mathcal{S}$ at time t respectively. With the *fare transformation* described in Section 3.5, Eq. (3.50) can be transformed back to an equivalent dynamic program with independent demand.

3.4.4 Network optimization

From a theoretical standpoint the stochastic network availability control problem Eq. (3.12) is not much different from the single-leg problem. Practically, however, the problem cannot be solved to optimality for more that a handful of resources, because it suffers from the so-called curse of dimensionality: the system's number of states grows exponentially in the number of resources m. In this section we first describe the theoretical optimal solution via dynamic programming and then review a number of heuristics that are widely used in practice.

Network DP

The dynamic programming formulation of the single-leg problem (Eq. (3.47)) can be extended to the network case in a straightforward fashion. One obtains the network dynamic program

$$\dot{V}_c(t) = \max_{\mathcal{S}\in\mathbb{S}} - \sum_{k\in\mathcal{S}} \lambda_k(t) \left[\mathbf{y}_k - \pi_c^{A_{\cdot,k}}(t)\right] \qquad \forall 0 \le c \le C \tag{3.52a}$$

$$\pi_c^a(t) = \begin{cases} V_c(t) - V_{c-a}(t) & \text{if } a \ge c \text{ component-wise,} \\ \infty & \text{else} \end{cases} \tag{3.52b}$$

$$V_c(T) = 0 \qquad \forall 0 \le c \le C, \tag{3.52c}$$

where $A_{\cdot,k}$ denotes the k-th column of the resource consumption matrix and $\pi_c^a(t)$ is the bid price at time t and remaining capacity c for a product that consumes the resources indicated by the vector a. In order to ensure that the capacity constraints are satisfied, we fix the convention that the expected revenue to come is $-\infty$ whenever any component of the capacity vector is negative. This leads to an infinite bid price in cases where the remaining capacity is too small to provide the resources needed for a certain product, thus automatically making the respective product unavailable at that state. This DP cannot be solved exactly for even medium sized instances, because its number of states equal to $\prod_{r=1}^{m} c_r$, which grows exponentially in the number of resources m.

Deterministic LP

One approach to the network problem is to treat demand as deterministic and solve a static allocation problem for the network. Let λ_k be the *deterministic* demand for product k over the booking horizon, for example equal to the expected number of requests $\lambda_k = \mathbb{E}\left[\mathbf{N}_k\right]$. Let again $\mathbf{y}_k$ be the expected yield for product k. Then we can write the network availability control problem as a Linear Program (LP), called the Deterministic Linear Program (DLP),

$$\max_u \quad \mathbf{y}^\top u \tag{3.53a}$$

$$\text{subject to} \quad Au \le C \tag{3.53b}$$

$$u \le \lambda \tag{3.53c}$$

$$u \ge 0 \tag{3.53d}$$

where A is the resource consumption matrix and C is the vector of initial capacities. Here, the decision variable $u \in \mathbb{R}^M$ is the number of requests to accept per booking class, Eq. (3.53c) ensures that the number of accepted requests is at most equal to demand, and Eq. (3.53b) enforces the capacity restrictions. Note that we do not pose any integrality conditions on u but allow partial bookings. Even for large airline networks this problem is still fairly small for LP standards and can be solved very quickly.

The primal solution of Eq. (3.53) has only very limited value in practice, because the fractional solution vector u needs to be rounded somehow in order to obtain sensible booking limits. Due to the fact that in large airline networks demand is often made up of a very large number of travel paths, each with only very little demand but relatively high variance, booking limit policies generally lead to very poor performance and are therefore rarely used in practice.

Alternatively, the dual variables associated with the capacity restrictions Eq. (3.53b) can be interpreted as the opportunity cost for one unit of the respective resource. Thus, it is natural to use these dual values as bid prices in a bid price control mechanism. This control policy does not depend on the fractional primal variables, but is still a static control policy that does not react to variance in the demand process. Therefore, in order to achieve a reasonable performance, frequent re-optimization during the booking horizon is necessary.

One interesting application of the DLP is the fact that the objective function value of an optimal solution to Eq. (3.53) is an upper bound on the optimal expected revenue for the stochastic problem Eq. (3.52) [31].

Functional approximation

Adelman and Zhang [1, 126] propose to apply the functional approximation approach, which is a well-known method in approximate dynamic programming [108, 33], to the network inventory control problem. The (discrete time) network DP is written as an equivalent LP with one constraint and one variable $V_{i,c}$ for every pair of time-step t_i and remaining capacity c, where $V_{i,c}$ is the expected revenue-to-come during the remainder of the booking horizon, given that the system is in state c at time t_i. Like the original DP, this LP suffers from the curse of dimensionality, because both the number of variables and the number of constraints grow exponentially in the number of resources in the network.

In the functional approximation approach, the value function V is now approximated using a linear combination of a fixed set of basis functions. Plugging this approximation into the LP one obtains a new LP where the optimization variables are now the coefficients in the approximation. Due to the structure of the original problem the new LP is still feasible, and for every feasible solution the approximate value function overestimates the true value function.

The LP now has significantly fewer variables, but it still has the same exponential number of constraints. Therefore Adelman and Zhang propose to use a column-generation algorithm on the dual problem. Numerical results suggest good quality of the value function approximation [1, 126]. However, the performance measurements indicate that on average the computational complexity still grows about quadratically in the number of resources in the network. While this is significantly better than the exponential complexity of the dynamic program, solving problems of realistic size—often containing more than 1000 flight legs for large airlines—still seems infeasible.

LP-DP decomposition

One of the most commonly used heuristics in the airline industry for the solution of the dynamic network problem uses the following idea:

Decompose the network into a set of single-leg problems.

Optimize each of the generated single-leg problems separately.

Control availability during the booking horizon using the solutions of the single-leg problem.

First, Eq. (3.52) is decomposed into single resource problems as follows: Let $\hat{\pi} = (\hat{\pi}_1, \dots, \hat{\pi}_m)$ be a vector of approximate bid prices for each resource, for example the dual variables associated to an optimal solution of the DLP. For every resource r and every product k that uses this resource—i.e. with $a_{r,k} \neq 0$—let k uses resource r

$$\bar{\mathbf{y}}_k^r = \mathbf{y}_k - \sum_{r' \neq r} a_{r',k} \hat{\pi}_{r'} \tag{3.54}$$

be the *displacement-adjusted yield* for product k on leg r, computed by reducing the expected yield of product k by the estimated opportunity cost for the capacity consumed on other legs. This dual decomposition is based on the Lagrangian relaxation: The sub-problem for the r-th resource is obtained from the network problem by applying a Lagrange relaxation to all other capacity constraints (for a detailed description see Section 6.2.1).

Using these adjusted yields as input, we then solve one single-leg DP for each resource and obtain a collection of value functions $V^1(\cdot), \dots, V^m(\cdot)$, which we can use to linearly approximate the value function

$$V_c(t) \approx \sum_{r=1}^{m} V_{c_r}^r(t) \tag{3.55a}$$

of the original network problem. Similarly, we have a linear approximation of the dynamic bid prices for the network problem

$$\pi_c^a(t) \approx \sum_{r=1}^{m} a_r \pi_{c_r}^r(t), \tag{3.55b}$$

where $\pi_{c_r}^r(t)$ is the bid price from the r-th single resource problem. Bid prices computed in this way are then used to implement a bid price control scheme as described in Eq. (3.28).

The classic LP-DP decomposition has shown good performance in studies and is widely used in practice. An extension to the case of choice-based RM is analyzed by Liu and Van Ryzin [119].

Zhang and Adelman [126] show that solutions to the decomposed problem yield better bounds on the optimal expected revenue for the dynamic network problem than the DLP. Zhang [125] also presents an improved decomposition approach that requires parallel evaluation of the single-leg problems, sharing information between problems during the optimization.

Value buckets and virtual nesting

Similar to the LP-DP decomposition, numerous other decomposition heuristics for the network availability control problem have been presented.

Prorated EMSR Using a vector of fixed weights $(\alpha_1, \ldots, \alpha_m)$ for each resource, one distributes the yield $\mathbf{y}_k$ for each product k over the set of resources consumed by k proportionally to their respective weights. An EMSR method is then applied to each resource using these *prorated* fares and the value function is again approximated linearly as in Eq. (3.55) in order to implement a bid price control [123]. Clearly the performance of prorated EMSR depends on the weight vector used to prorate the yields. However, because a correct opportunity cost depends on demand, prorating with static weights can never be optimal.

DAVN Like the LP-DP decomposition, Displacement adjusted virtual Nesting (DAVN) uses displacement-adjusted fares computed according to Eq. (3.54) using an initial bid price vector π. For each resource, products are then clustered into *virtual products* or *value buckets* w.r.t. their adjusted fares, a virtual yield and demand is computed for each cluster and a single-leg optimization is performed on this data.

A *primal* control policy can be derived by accepting a request for a certain product if all corresponding virtual products are available on their respective legs. Alternatively, one can again use a linear approximation of the value function and implement a bid-price control using the original fares for each product.

It has been shown that the performance of DAVN strongly depends on the clustering method used to construct the virtual products [120, 121].

3.5 Demand and fare transformation

If the demand forecast used in an optimization algorithm satisfies certain conditions, the demand model can be reduced to an equivalent independent demand model. This makes it possible to use complex discrete choice models in conjunction with optimization techniques developed under the assumption of independent demand without the need to incorporate customer choice into the optimization algorithms explicitly. In particular, airlines can modify their forecasting models and tools while continuing to use existing optimization systems.

The concept of *transformed demand* and *transformed revenue* was first presented by Fiig et al. [42] and Isler et al. [63] in 2005. In this summary we will closely follow the summary paper of Fiig et al. from 2010 [41]. We will first consider deterministic demand and describe how the method can be generalized to stochastic models later on.

Assume we have a fixed capacity C and a set of products $\mathcal{P} = \{1, \ldots, M\}$, each consuming one unit of capacity, and their respective yields $\mathbf{y}_1 \geq \mathbf{y}_2 \geq \cdots \geq \mathbf{y}_M$.

Control can be exercised by choosing a set of available products $\mathcal{S} \in \mathbb{S} \subseteq \wp(\mathcal{P})$ from the set of feasible actions $\mathbb{S}$ which is a subset of the power set of $\mathcal{P}$. In most cases $\mathbb{S} = \wp(\mathcal{P})$, meaning that in fact every offer set can be chosen. Exceptions are usually either the result of business restrictions or an attempt to reduce computational complexity, as described towards the end of this section.

Demand is given as a vector-valued function

$$D\colon \mathbb{S} \subseteq \wp(\mathcal{P}) \to \mathbb{R}^{\mathcal{P}} \tag{3.56}$$

$$\mathcal{S} \mapsto (D_k(\mathcal{S}))_{k\in\mathcal{P}}\,, \tag{3.57}$$

where $D_k(\mathcal{S})$ denotes the demand that is expected for product k, given availability $\mathcal{S}$. Here, *demand* can have slightly different meanings depending on the specific model: It is always a deterministic quantity, either indicating a deterministic number of bookings, the expected value of a random number of bookings, or an arrival rate of a stochastic process.

The fare transformation exploits the fact that most RM optimization methods do not require full knowledge about how customers choose between products. Instead, it is often sufficient to know the derived quantities

$$\mathbf{D}(\mathcal{S}) = \sum_{k=1}^{M} D_k(\mathcal{S}) \tag{3.58a}$$

$$\mathbf{R}(\mathcal{S}) = \sum_{k=1}^{M} \mathbf{y}_k D_k(\mathcal{S}), \tag{3.58b}$$

where $\mathbf{D}$ and $\mathbf{R}$ are *total demand* and *total revenue* associated with the offer set $\mathcal{S}$ respectively. Again, depending on the specific model, $\mathbf{D}$ and $\mathbf{R}$ can be expected values of random quantities or arrival rates of stochastic processes. The basic idea of the fare transformation mechanism is to design a set of virtual booking classes $\{1, \dots, K\}$, with virtual yields $\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_K$—called *transformed fares*—and independent virtual demands $\tilde{D}_1, \dots, \tilde{D}_K$—called *transformed demand*—in such a way that the original total revenue and total demand as defined in Eq. (3.58) can be reconstructed from these values.

3.5.1 Deterministic demand

We assume that demand is a deterministic vector $D_k(\mathcal{S})$ depending on the offer set $\mathcal{S}$ and indexed by the set of products $\mathcal{P}$. The generic revenue optimization problem can be formulated as

$$\max_{\mathcal{S} \in \mathbb{S}} \quad \mathbf{R}(\mathcal{S}) \tag{3.59a}$$

$$\text{subject to} \quad \mathbf{D}(\mathcal{S}) \leq C \tag{3.59b}$$

with total revenue $\mathbf{R}$ and total demand $\mathbf{D}$ as in Eq. (3.58).

As a relaxation we can allow *mixed control strategies*, which leads to the convexified program

$$\max_{u \in \mathbb{R}^M} \quad \sum_{\mathcal{S} \in \mathbb{S}} \mathbf{R}(\mathcal{S}) u_{\mathcal{S}} \tag{3.60a}$$

$$\text{subject to} \quad \sum_{\mathcal{S} \in \mathbb{S}} \mathbf{D}(\mathcal{S}) u_{\mathcal{S}} \leq C \tag{3.60b}$$

$$\sum_{\mathcal{S} \in \mathbb{S}} u_{\mathcal{S}} = 1 \tag{3.60c}$$

$$u_{\mathcal{S}} \geq 0 \qquad \forall \mathcal{S} \in \mathbb{S}. \tag{3.60d}$$

Assuming that demand is spread homogeneously across the booking horizon, a solution to this problem can be interpreted as offering each set of products for a fraction of the booking period according to the respective component of u.

Independent demand

In case demand is independent between products we can associate to each product k its demand D_k and compute

$$\mathbf{D}(\mathcal{S}) = \sum_{k \in \mathcal{S}} D_k \tag{3.61a}$$

$$\mathbf{R}(\mathcal{S}) = \sum_{k \in \mathcal{S}} D_k \mathbf{y}_k. \tag{3.61b}$$

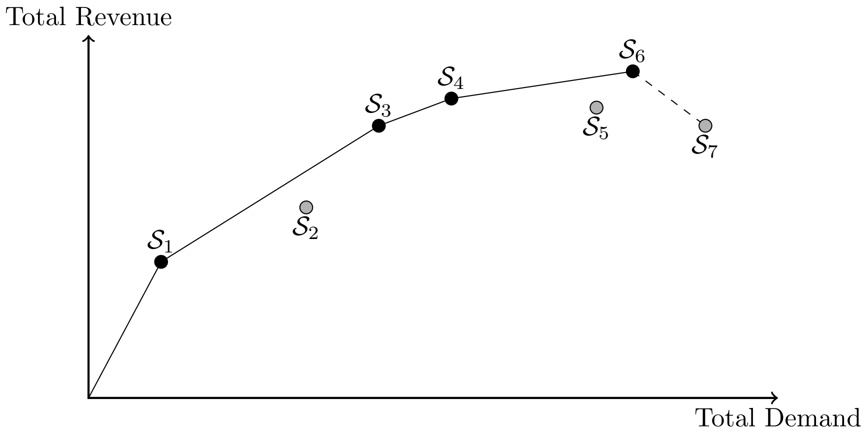


Figure 3.1: Scatter plot of feasible actions

If $\mathbb{S} = \wp(\mathcal{P})$, in other words if there are no restrictions on the possible offer sets, we can use the fact that $\mathbf{R}$ and $\mathbf{D}$ are separable and derive that (3.60) is equivalent to

$$\max_u \quad \sum_{k=1}^{M} D_k \mathbf{y}_k u_k \tag{3.62a}$$

$$\text{subject to} \quad \sum_{k=1}^{M} D_k u_k \le C \tag{3.62b}$$

$$u_k \in [0,1] \quad \forall k = 1, \ldots M \tag{3.62c}$$

where the components of the control variable u are indexed by the set of products $\mathcal{P}$ and determine the fraction of the booking horizon during which the respective products are available.

Remark 3.5.1 Problem (3.62) is a fractional version of the well-known knapsack problem and can easily be solved to optimality using a *greedy algorithm* that allocates capacity in decreasing order of revenue per unit of capacity, which matches the intuition to accept high value customers first before making capacity available for less profitable products. Thus, an optimal solution to (3.62) always has the form $x = (1, \ldots, 1, u_k, 0, \ldots, 0)^\intercal$, meaning that the first $k-1$ products are available, the next product is only available for the fraction of time needed to fill up the remaining capacity and all other products are unavailable. As a result, the offer sets used in optimal control strategies depending on capacity C are the *nested* sets $\{1\}, \{1,2\}, \ldots, \{1,2,\ldots,M\}$.

General discrete choice models

For a general discrete choice model, the solution is most easily described using a scatter plot of all feasible choice sets $\mathbb{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$, using total demand $\mathbf{D}(\mathcal{S}_k)$ as the x-coordinate and total revenue $\mathbf{R}(\mathcal{S}_k)$ as y-coordinate for the point $\mathcal{S}_k$ (see Fig. 3.1). Because in Eq. (3.60) we allow convex combinations of offer sets, the set of feasible combinations of total demand and total revenue is the convex hull of these points.

Definition 3.5.2 An offer set $\mathcal{S} \in \mathbb{S}$ is *inefficient* in $\mathbb{S}$ if and only if there exists a vector $u \in \mathbb{R}_+^K$ indexed by the elements of $\mathbb{S}$, satisfying

$$u_{\mathcal{S}} = 0 \tag{3.63a}$$

$$\sum_{\bar{\mathcal{S}} \in \mathbb{S}} u_{\bar{\mathcal{S}}} = 1 \tag{3.63b}$$

$$\sum_{\bar{\mathcal{S}} \in \mathbb{S}} \mathbf{D}(\bar{\mathcal{S}}) u_{\bar{\mathcal{S}}} \le \mathbf{D}(\mathcal{S}) \tag{3.63c}$$

$$\sum_{\bar{\mathcal{S}} \in \mathbb{S}} \mathbf{R}(\bar{\mathcal{S}}) u_{\bar{\mathcal{S}}} > \mathbf{R}(\mathcal{S}) \tag{3.63d}$$

An offer set is called *efficient* if it is not inefficient. In other words, an offer set is inefficient if there exists a mixed strategy that uses capacity more efficiently by generating a higher expected total revenue with the same or lower expected total demand.

Remark 3.5.3 One can easily see that equation (3.63a) is in fact unnecessary. However, it does make clear that an optimal mixed strategy can always be formed without using any inefficient offer sets.

Remark 3.5.4 The offer set $\mathcal{S} \in \mathbb{S}$ with maximum total revenue is always efficient and all products $\bar{\mathcal{S}}$ with $\mathbf{D}(\bar{\mathcal{S}}) > \mathbf{D}(\mathcal{S})$ are inefficient.

Furthermore it is clear that the empty offer set $\mathcal{S}_0 = \{\}$ is always efficient.

In addition, total revenues of efficient sets are monotonically increasing in their respective demands. In other words, if $\mathcal{S}, \bar{\mathcal{S}} \in \mathbb{S}$ are both efficient, then $\mathbf{D}(\mathcal{S}) > \mathbf{D}(\bar{\mathcal{S}}) \Rightarrow \mathbf{R}(\mathcal{S}) > \mathbf{R}(\bar{\mathcal{S}})$.

Remark 3.5.5 If overall customer choice behavior changes during the course of the booking horizon, for example through a varying mix of business and leisure customers, the set of efficient offer sets will generally change as well. In this work we assume that choice probabilities are piecewise constant over time. Therefore, the fare transformation needs to be applied for each time interval separately.

Let $\mathbb{S}^* = \{\mathcal{S}_0, \dots \mathcal{S}_K\} \subseteq \mathbb{S}$ be the set of efficient offer sets ordered by increasing total demand $0 = \mathbf{D}(\mathcal{S}_0) < \mathbf{D}(\mathcal{S}_1) < \cdots < \mathbf{D}(\mathcal{S}_K)$. Then for every capacity C there is an optimal mixed control strategy using at most two offer sets $\mathcal{S}_k$ and $\mathcal{S}_{k+1}$. We call the set of mixed strategies of this type the *efficient frontier* of $\mathbb{S}$.

Proposition 3.5.6
If $C < \mathbf{D}(\mathcal{S}_K)$ then an optimal solution can be found at the intersection of the efficient frontier and the line $\mathbf{D} = C$, otherwise $\mathcal{S}_K$ is optimal. Therefore, for every given capacity C there is an optimal solution that lies on the efficient frontier of $\mathbb{S}$.

Using this fact we can derive a structure of *virtual products* associated to the efficient sets, such that the optimal strategy described above is equivalent to the optimal strategy of the independent demand case applied to the virtual products.

Definition 3.5.7 (Marginal Revenue Transformation) Let again $\mathbb{S}^*$ denote the set of efficient offer sets ordered by total demand, including the empty set $\mathcal{S}_0 = \{\}$. For every $k = 1, \dots, K$, the *transformed demand* $\tilde{D}$ of the offer set $\mathcal{S}_k \in \mathbb{S}$ is the marginal demand

$$\tilde{D}_k = \mathbf{D}(\mathcal{S}_k) - \mathbf{D}(\mathcal{S}_{k-1}) \tag{3.64a}$$

and the *transformed fare* $\tilde{\mathbf{y}}$ is the additional revenue per unit of marginal demand

$$\tilde{\mathbf{y}}_k = \frac{\mathbf{R}(\mathcal{S}_k) - \mathbf{R}(\mathcal{S}_{k-1})}{\mathbf{D}(\mathcal{S}_k) - \mathbf{D}(\mathcal{S}_{k-1})}. \tag{3.64b}$$

Theorem 3.5.8
Let $\mathbb{S}^* \subseteq \mathbb{S}$ be the set of efficient offer sets, ordered by increasing total demand. Let $\{1, \dots, K\}$ be the set of virtual products corresponding to the elements of $\mathbb{S}^*$. For each such virtual product k, let demand $\tilde{D}_k$ and yield $\tilde{\mathbf{y}}_k$ be as in Eq. (3.64)*. Let $\tilde{u}$ be the optimal solution to the independent demand problem* (3.62)*. Then u given by*

$$u_0 = 1 - \tilde{u}_1 \tag{3.65a}$$

$$u_k = \tilde{u}_k - \tilde{u}_{k+1} \qquad \forall k = 1, \dots, K-1 \tag{3.65b}$$

$$u_K = \tilde{u}_K \tag{3.65c}$$

is an optimal solution to problem (3.60) *with total expected demand and revenue*

$$\sum_{k=0}^{K} \mathbf{D}(\mathcal{S}_k) u_k = \sum_{k=1}^{K} \tilde{D}_k \tilde{u}_k \tag{3.66a}$$

$$\sum_{k=0}^{K} \mathbf{R}(\mathcal{S}_k) u_k = \sum_{k=1}^{K} \tilde{D}_k \tilde{\mathbf{y}}_k \tilde{u}_k. \tag{3.66b}$$

Proof As described in Remark 3.5.1, $\tilde{u} = (1, \ldots, 1, \tilde{u}_k, 0, \ldots, 0)^\top$ with $0 \le \tilde{u}_k \le 1$. Therefore, by definition (Eq. (3.65)), one immediately sees that u satisfies (3.60d). Equation (3.60c) is satisfied because of the telescopic sum

$$\sum_{k=0}^{K} u_k = (1 - \tilde{u}_1) + \sum_{k=1}^{K} (\tilde{u}_k - \tilde{u}_{k+1}) + \tilde{u}_K = 1. \tag{3.67}$$

Total expected demand is given by

$$\begin{aligned}\sum_{k=0}^{K} \mathbf{D}(\mathcal{S}_k) u_k &= \mathbf{D}(\mathcal{S}_0)(1 - \tilde{u}_1) + \sum_{k=1}^{K-1} \mathbf{D}(\mathcal{S}_k)(\tilde{u}_k - \tilde{u}_{k+1}) + \mathbf{D}(\mathcal{S}_K)\tilde{u}_K \\ &= \underbrace{\mathbf{D}(\mathcal{S}_0)}_{=0} + \sum_{k=1}^{K} (\mathbf{D}(\mathcal{S}_k) - \mathbf{D}(\mathcal{S}_{k-1}))\tilde{u}_k \\ &= \sum_{k=1}^{K} \tilde{D}_k \tilde{u}_k\end{aligned}$$

and analogously expected revenue is

$$\begin{aligned}\sum_{k=0}^{K} \mathbf{R}(\mathcal{S}_k) u_k &= \mathbf{R}(\mathcal{S}_0)(1 - \tilde{u}_1) + \sum_{k=1}^{K-1} \mathbf{R}(\mathcal{S}_k)(\tilde{u}_k - \tilde{u}_{k+1}) + \mathbf{R}(\mathcal{S}_K)\tilde{u}_K \\ &= \underbrace{\mathbf{R}(\mathcal{S}_0)}_{=0} + \sum_{k=1}^{K} (\mathbf{R}(\mathcal{S}_k) - \mathbf{R}(\mathcal{S}_{k-1}))\tilde{u}_k \\ &= \sum_{k=1}^{K} \tilde{D}_k \tilde{\mathbf{y}}_k \tilde{u}_k,\end{aligned}$$

where the last equalities follow from Eqs. (3.64a) and (3.64b) respectively.

By construction, the solutions obtained in the process of opening the virtual booking classes one by one as capacity increases all lie on the efficient frontier of $\mathbb{S}$. If total demand is lower than capacity the solution of the independent demand problem is to open all booking classes, which leads to $u = (0, \ldots, 0, 1)^\top$, corresponding to the offer set $\mathcal{S}_K$. If $C < \mathbf{D}(\mathcal{S}_K)$, the demand inequality becomes active for both problems and the optimal solution is at the intersection of the efficient frontier with the line $C = \mathbf{D}$. Hence the solution is optimal for (3.60) in both cases following Proposition 3.5.6 □

3.5.2 Dynamic programming

In dynamic programming the local optimization problem to be solved for every time t and every capacity c is of the form

$$\max_{\mathcal{S} \in \mathbb{S}} \quad \mathbf{R}(\mathcal{S}) - \mathbf{D}(\mathcal{S})\pi \tag{3.69}$$

where $\mathbf{R}(\mathcal{S}) = \mathbf{R}(\mathcal{S}, t)$ and $\mathbf{D}(\mathcal{S}) = \mathbf{D}(\mathcal{S}, t)$ denote time-dependent total revenue and total demand for availability $\mathcal{S}$ respectively and $\pi = \pi_c(t)$ is the bid price at the current state and time.

Independent demand

If demand is independent and there are no restrictions on the possible offer sets, Eq. (3.69) is equivalent to

$$\max_{u} \quad \sum_{k=1}^{M} D_k (\mathbf{y}_k - \pi) u_k \tag{3.70a}$$

$$\text{subject to} \qquad u_k \in [0, 1] \quad \forall k = 1, \ldots M \tag{3.70b}$$

where the control variable u, the demand vector D and the vector of expected yields $\mathbf{y}$ are indexed by the set of products $\mathcal{P} = \{1, \ldots M\}$. The optimal solution is trivially given by

$$u_k = \begin{cases} 1 & \text{if } \mathbf{y}_k > \pi, \\ 0 & \text{else,} \end{cases} \tag{3.71}$$

because there are only simple bounds on the variables and the objective function is separable. Since the bid price measures the expected value of the seat to be sold, the optimal offer set contains exactly those products that will yield a net gain when sold at the current state.

Like in the deterministic model, the optimal offer sets are again nested, with products becoming unavailable one by one in increasing fare order as the bid price rises.

General discrete choice models

At a bid price of $\pi = 0$ obviously the optimal solution is just the offer set with the highest total revenue. As the bid price increases, the slope of the objective function is slanted and the optimal solution changes to an offer set with higher average yield per unit of capacity used. This again resembles the optimal policy described above for the independent demand model.

Proposition 3.5.9
Let $\mathcal{S} \in \mathbb{S}$ be an optimal solution to (3.69)*. Then $\mathcal{S}$ is efficient in $\mathbb{S}$.*

Let $\mathbb{S}^* \subseteq \mathbb{S}$ be the set of efficient offer sets, ordered by increasing total demand. If transformed demand and transformed fare for the virtual products $\{1, \ldots, K\}$ are defined as in (3.64a) *and* (3.64b)*, then the value function obtained in the solution to the corresponding independent demand DP is identical to the value function obtained from the solution of the choice-based DP.*

Proof Consider again the relaxed problem

$$\max_u \quad \sum_{\mathcal{S} \in \mathbb{S}} (\mathbf{R}(\mathcal{S}) - \mathbf{D}(\mathcal{S})\pi)\, u_{\mathcal{S}} \tag{3.72a}$$

$$\text{subject to} \quad \sum_{\mathcal{S} \in \mathbb{S}} u_{\mathcal{S}} = 1 \tag{3.72b}$$

$$u_{\mathcal{S}} \geq 0 \quad \forall \mathcal{S} \in \mathbb{S}. \tag{3.72c}$$

This problem is linear and there are no integrality conditions. Therefore, there exists an optimal solution at a corner of the feasible set. This solution $\mathcal{S}$ is an element of $\mathbb{S}$, because by construction we optimize over the convex hull of $\mathbb{S}$. Therefore, the solution is feasible and optimal for (3.69). Also, $\mathcal{S}$ is efficient in $\mathbb{S}$, because, by definition, an inefficient set cannot be an optimal solution to (3.72).

The second part follows directly from (3.66). □

Summary

Using the demand transformation introduced in this section, it is possible to apply algorithms that were developed under the assumption of independent demand to problems involving very general customer choice models.

The computational effort for solving the transformed independent demand problem is then the same as for the general problem when only applied to the set of efficient sets. However, in general the number of efficient sets can grow exponentially in the number of products. Therefore, the computational cost involved can still be much higher than in the independent demand case.

In practice it is therefore common to use discrete choice models that always lead to nested strategies. Alternatively one can restrict the set of feasible actions to a subset of $\wp(\mathcal{S})$ that is small enough that the resulting problem is computationally tractable.

However, in particular in a DP, even with exponentially many virtual products, the demand transformation is still a useful tool to reduce computational effort. If demand is (piecewise) constant over time, the set of efficient offer sets only has to be computed once at the beginning of the optimization, reducing the number of options that have to be evaluated for the solution of the local problem in each time step. Using the monotonicity properties of bid prices, optimal controls can the be computed very efficiently in each time step.

3.6 Dynamic pricing

Dynamic Pricing is an alternative control method in RM that, instead of controlling the availability of a number of products with fixed prices, uses dynamically varying prices for all products to steer the booking process. In other words, in this scenario price is not considered a fixed product characteristic but a control variable. Product restrictions are still fixed and described by the remaining attributes. Given a price for each product, choice behavior is the same as in the availability control case.

Dynamic pricing was introduced to airline RM by Gallego and van Ryzin, who gave a solution to the static problem and presented several heuristic solutions for the dynamic problem [46, 47]. Since then it has been a common topic in the RM literature and we will only list a few recent developments. Otero and Akhavan-Tabatabaei [96] present a novel approach for estimating estimate arrival rates and price elasticity and a corresponding optimization algorithm. Fiig et al. [40] propose a dynamic pricing mechanism that uses real-time competitor price information, which, if implemented would raise additional game-theoretic questions. For further variants of dynamic pricing considering additional effects such as demand uncertainty can be found in the survey article by Chen and Chen [28].

In this section we will describe a dynamic solution for the single-leg case with the same basic assumptions as for the inventory control problem presented in the previous sections. Extension to the network problem works analogously to the availability control case.

We again model demand as an inhomogeneous Poisson process on the booking horizon $[0,T]$ with time-dependent rate

$$\lambda\colon [0,T] \to \mathbb{R}_{\geq 0}$$

and assume that choices of individual customers are made independently. Let $\mathcal{P} = \{1,\ldots,M\}$ denote the set of products. Given a time-dependent vector $(f_k(t))_{k\in\mathcal{P}}$ of prices, the booking process for product $k \in \mathcal{P}$ is again a Poisson process with rate

$$\lambda_k(t,f) = \lambda(t) d_k(t,f) \tag{3.73}$$

where $d_k(t,f)$ denotes the probability that a random customer arriving at time t will book product k given the prices f. Note that in general the booking rate for product k does not only depend on its price f_k but on the full price vector f. Overall we can model the booking process as a multivariate Poisson process with time- and price-dependent rate vector or *demand function*

$$\lambda\colon [0,T] \times D \subseteq \hat{\mathbb{R}}^M \to \mathbb{R}^M \tag{3.74}$$

where $\hat{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ and D is the set of feasible price vectors. In order to ensure sensible solutions and computational feasibility for the optimization problem, we expect the demand function to meet certain requirements:

Definition 3.6.1 A demand function

$$\lambda\colon [0,T] \times D \subseteq \hat{\mathbb{R}}^M \to \mathbb{R}^M \tag{3.75}$$

$$(t,f) \mapsto \lambda(t,f) \tag{3.76}$$

is called *regular* if it satisfies the following conditions for every $t \in [0,T]$:

(1) $\lambda(t,f)$ is continuously differentiable w.r.t. f.

(2) For every product $k \in \mathcal{P}$, the booking rate $\lambda_k(t,f)$ is strictly monotonically decreasing in its own price f_k.

(3) The demand rate is bounded on D:

$$\exists C : \forall f \in D : 0 \leq \|\lambda(t,f)\| \leq C < \infty$$

(4) For every product $k \in \mathcal{P}$ there is a *null price* f_k^∞ such that

$$f_k \geq f_k^\infty \Rightarrow \lambda_k(t,f) = 0. \tag{3.77}$$

(5) The *revenue rate*

$$\mathbf{r}\colon [0,T] \times D \to \mathbb{R} \tag{3.78a}$$
$$(t,f) \mapsto \mathbf{r}(t,f) = f^{\top}\lambda(t,f) \tag{3.78b}$$

is bounded on D and has a finite maximizer in the interior of D.

Remark 3.6.2 In certain situations the airline might only want to offer a subset of the full set of products, and not offer other products at all, combining both availability control and dynamic pricing. However, we do not explicitly have to offer this additional degree of freedom in the control mechanism, because setting a price to the null price effectively makes a product unavailable to the customer, which already has the desired effect. With a null price being present for each product, the dynamic pricing problem is therefore a generalization of the availability control problem.

Remark 3.6.3 Instead of Eq. (3.78) one usually optimizes a revenue rate of the form

$$\mathbf{r}\colon [0,T] \times D \to \mathbb{R} \tag{3.79a}$$
$$(t,f) \mapsto \mathbf{r}(t,f) = \mathbf{y}(t,f)^{\top}\lambda(t,f), \tag{3.79b}$$

where $\mathbf{y}$ is a time-dependent yield function. If $\mathbf{y}$ is linear in f, Eq. (3.79) has a finite maximizer if and only if Eq. (3.78) does. In most cases the yield function is of the form

$$\mathbf{y}_k(t,f) = f_k - \pi_k(t) - c_k \tag{3.80}$$

where $\pi_k(t)$ is the time-dependent bid price that measures the value of the resources consumed by one unit of product k, and c_k is a constant which could for example represent the variable costs associated with a booking of product k.

Example 3.6.1 The most common demand model in the dynamic pricing literature assumes independent demand with exponentially distributed willingness-to-pay. Let $D = \hat{\mathbb{R}}^M$ be the set of feasible prices.

For a fixed product k, customer arrival is modeled as an inhomogeneous Poisson process $\mathbf{N}_k$ with arrival rate $\lambda^0_{\mathrm{p}}(t)$. Each customer will purchase the product if their willingness-to-pay exceeds the offered price, in other words if

$$f_k \leq \mathbf{X}, \tag{3.81}$$

where the random variable $\mathbf{X}$ is exponentially distributed with time-dependent scale parameter $\beta_k(t)$. Since customers' choices are independent, the demand function is given by

$$\lambda\colon [0,T] \times \hat{\mathbb{R}}^M \to \mathbb{R}^M \tag{3.82a}$$
$$(t,f) \mapsto \lambda(t,f) \tag{3.82b}$$
$$\lambda_k(t,f) = \lambda^0_{\mathrm{p}}(t)e^{-\frac{f_k}{\beta_k(t)}} \qquad \forall k \in \mathcal{P}. \tag{3.82c}$$

The dynamic pricing optimization problem can—analogously to the construction of the availability control dynamic program (Section 3.4.3)—be written as

$$\max_f V_C(0) \tag{3.83a}$$
$$\dot{V}_c(t) = -\sum_{k\in\mathcal{P}} \lambda_k(t, f(c,t))\left[\mathbf{y}_k(t, f(c,t)) - \pi_c(t)\right] \qquad \forall 1 \leq c \leq C \tag{3.83b}$$
$$\pi_c(t) = \begin{cases} V_1(t) & \text{if } c = 1, \\ V_c(t) - V_{c-1}(t) & \text{else} \end{cases} \tag{3.83c}$$
$$V_c(T) = 0 \qquad \forall 1 \leq c \leq C, \tag{3.83d}$$

where f is a *pricing scheme*

$$f\colon \{1,\dots,C\} \times [0,T] \to \hat{\mathbb{R}}^M \tag{3.84a}$$
$$(c,t) \mapsto f(c,t), \tag{3.84b}$$

mapping a pair of remaining capacity c and time t to a vector of prices $f(c,t)$. Using the memorylessness of the Poisson process and the fact that booking decisions are independent between each other, we can derive the dynamic pricing dynamic program

$$\dot{V}_c(t) = -\max_f \sum_{k\in\mathcal{P}} \lambda_k(t,f)\,[\mathbf{y}_k(t,f) - \pi_c(t)] \qquad \forall 1 \le c \le C \tag{3.85a}$$

$$\pi_c(t) = \begin{cases} V_1(t) & \text{if } c = 1, \\ V_c(t) - V_{c-1}(t) & \text{else} \end{cases} \tag{3.85b}$$

$$V_c(T) = 0 \qquad \forall 1 \le c \le C. \tag{3.85c}$$

Here, every evaluation of the right-hand-side requires the solution of an NLP of the form

$$\max_f \sum_{k\in\mathcal{P}} \lambda_k(t,f)\,[\mathbf{y}_k(t,f) - \pi]\,. \tag{3.86}$$

This is a static multi-product pricing problem with marginal cost π, which is well studied in the economics literature. Depending on the demand function λ, this problem can be non-convex. Therefore, the optimal prices f^* are not necessarily continuous in the parameter π. This makes it especially hard to solve the dynamic program Eq. (3.85a), because (at least in theory), a global optimum for the sub-problem has to be found in at each time step and for every capacity c.

Example Continuing with example Example 3.6.1, note that the demand rate λ_k of product k only depends on the price of k and is independent from other prices (Eq. (3.82c)). Thus the objective function in the NLP in Eq. (3.85a) is separable between the products. Then, assuming no variable costs, in other words $\mathbf{y}_k(f_k) = f_k$ for every product $k \in \mathcal{P}$, we have $\forall 1 \le c \le C$:

$$\dot{V}_c(t) = -\max_f \sum_{k\in\mathcal{P}} \lambda_k(t,f)\,[f - \pi_c(t)] \tag{3.87}$$
$$= -\sum_{k\in\mathcal{P}} \max_{f_k} \lambda_k(t,f_k)\,[f_k - \pi_c(t)]\,. \tag{3.88}$$

Now, for each product $k \in \mathcal{P}$ we have to solve a one-dimensional nonlinear sub-problem of the form

$$\max_f \mathbf{r}(f) = \lambda(t,f)\,[f - \pi] \tag{3.89}$$

where π is a constant. The first order optimality conditions for Eq. (3.89) are

$$0 = \frac{\mathrm{d}}{\mathrm{d}f}\left(\lambda(t,f)\,[f-\pi]\right) \tag{3.90}$$
$$= (f-\pi)\,\frac{\mathrm{d}}{\mathrm{d}f}\lambda(t,f) + \lambda(t,f) \tag{3.91}$$
$$= (f-\pi)\,\lambda^0(t)\frac{-1}{\beta(t)}e^{-\frac{f}{\beta(t)}} + \lambda^0(t)e^{-\frac{f}{\beta(t)}} \tag{3.92}$$
$$\Leftrightarrow \qquad 0 = -\frac{(f-\pi)}{\beta(t)} + 1 \tag{3.93}$$
$$\Leftrightarrow \qquad f = \pi + \beta(t), \tag{3.94}$$

where in Eq. (3.92) we assume that $\lambda^0(t) > 0$ and $\beta(t) > 0$. This is w.l.o.g. because

a) If $\lambda^0 = 0$, then the objective function value is $\mathbf{r}(f) \equiv 0$ regardless of f.

b) If $\beta < 0$ the problem is unbounded, and for $\beta = 0$ the demand rate is not well defined.

It is clear that $\mathbf{r}(f) \geq 0$ for all non-negative prices f, $\mathbf{r}(0) = 0$ and $\lim_{f\to\infty} \mathbf{r}(f) = 0$. The unique critical point Eq. (3.94) is therefore a global maximum. Using

$$\lambda(t, \pi + \beta(t)) = \lambda^0(t) e^{-\frac{\pi}{\beta(t)} - 1} \tag{3.95a}$$

$$\mathbf{r}(\pi + \beta(t)) = \lambda(t, \pi + \beta(t))) \left[\pi + \beta(t) - \pi\right] = \lambda^0(t) e^{-\frac{\pi}{\beta(t)} - 1} \beta(t) \tag{3.95b}$$

we see that the dynamic program Eq. (3.85) has the solution

$$\dot{V}_c(t) = -\sum_{k\in\mathcal{P}} \lambda^0_{\mathfrak{p}}(t) \beta_k(t) e^{-\frac{\pi_c(t)}{\beta_k(t)} - 1} \qquad \forall 1 \leq c \leq C \tag{3.96a}$$

$$\pi_c(t) = \begin{cases} V_1(t) & \text{if } c = 1, \\ V_c(t) - V_{c-1}(t) & \text{else} \end{cases} \tag{3.96b}$$

$$V_c(T) = 0 \qquad \forall 1 \leq c \leq C. \tag{3.96c}$$

Equation (3.96a) is an ODE that can be solved using standard numerical methods.

Chapter 4

The airline pricing problem

When traveling with one of the large network carriers, customers never just purchase a seat on a flight. Instead, they purchase a trip from A to B along a certain flight path—often including multiple flights—together with certain conditions that describe how, when and at what price the travel plans can be changed and what additional services are included in the ticket. Therefore, in the airline industry the term *pricing* not only refers to the act of setting a price for a certain product, but includes part of the product definition as well. It means the process of defining and publishing *fares*, which are composed of a fixed price, rules that determine on which O&Ds, itineraries and airlines the fare can be used, booking flexibility conditions, and a list of additional services such as a free checked bag or lounge access. In addition airlines heavily use conditions on where and when a fare can be applied as a means for customer segmentation, for by example selling discounted fares only to customers staying at their destination for a long period of time, which indicates that their reason for travel is leisure rather than business. These fares are the products that are sold by the airline. Analysis of price segmentation by airlines was conducted by Zhang and Bell [127].

The goal of airline pricing is to define a set of products that is optimal with respect to the total expected revenue on the network. Clearly, we cannot freely determine the product a customer experiences in its entirety, because tangible characteristics like customer service or quality of equipment cannot be simply changed. We can however change the more abstract properties of a product, most notably the flexibility we give the customer concerning choice and changes of their exact travel plans, and the segmentation criteria tied to the fares such as minimum stay or advanced purchase requirements.

In order to avoid confusion we want to point out the difference between *pricing* and *dynamic pricing*. The former is the problem of defining an optimal set of products, while the latter is a dynamic control mechanism in RM that can be used instead of availability control (see Section 3.6 for details).

Compared to capacity control, where a large body of scientific research and practical tools are available, the pricing side of RM is still mostly done manually, backed up with business intelligence tools that provide statistical information such as data collected from historical sales or competitor prices and schedules. We want to explore ways to use mathematical optimization techniques on the pricing problem in order to provide additional decision support tools for pricing analysts. In the last few years interest in pricing optimization has increased significantly both in airline RM and in other research areas with a number of publications that are relevant to this thesis. Kocabıyıkouglu et al. [68] analyze the value of coordinating pricing and inventory decisions in a single resource, two-class model with price-sensitive stochastic demand where inventory is controlled via booking limits. In simulation studies they show that methods which take into demand stochasticity significantly outperform deterministic methods, and that substantial revenue gains can be achieved by integrating pricing with inventory control. Cizaire and Belobaba [29] consider a heavily simplified version of the joint pricing and inventory control problem with two booking periods and two fare classes, where only prices (not fare restrictions) are the controls on the pricing side and booking limit controls are used to control availability. Côté et al. [32] consider a deterministic version of the problem that where neither overall demand nor individual customer choice are subject to random effects. He [59] analyzes the structural properties of a deterministic pricing problem for a specific type of two-hub network. So [112] formulates and solves a deterministic airline pricing optimization problem that focuses on choosing optimal prices for different customer segments (identified based on the time of purchase) under the very strong assumption that within each primary segment there are perfectly controlled secondary segmentation criteria and inventory control mechanisms (which are not part of the optimization model but assumed to exist implicitly) that force every customer to purchase the most expensive fare they can afford. Yu [124] analyzes the structural properties of a joint pricing and inventory control problem with two products sharing a single resource, where demand linearly depends on price and inventory is controlled via non-nested booking limits. Raza [104] formulates a joint pricing, customer segmentation and inventory control problem with two fare classes corresponding to two customer segments. Customers belong to the higher customer segment if their willingness-to-pay is above a certain threshold and to the lower one otherwise. Although it is unclear how this could be achieved in practice, in the model this threshold can be controlled by the seller (i.e. the seller can freely choose at which willingness-to-pay to split customers into the two segments) and is one of the control variables in the optimization problem. Kuyumcu and Garcia-Diaz [73] solve a joint pricing and seat allocation problem where the pricing part is formulated as the discrete problem of choosing a subset of fares with given cardinality from a larger, fixed menu of fares. However, because demand is assumed to be independent between fares, the approach does not really set prices for any customer at all but is rather an extended availability control problem with an additional constraint on the number of available fare classes. Gallego and Wang[48] consider price optimization under competition assuming a nested multinomial logit model without taking into account capacity constraints. Li et al. [77] present an efficient algorithm for assortment optimization and pricing that maximizes expected revenue from a single sale assuming a d-level nested logit model. The authors show that the assortment optimization algorithm can be applied within a choice-based dynamic inventory control problem determine optimal booking class availability assuming fixed prices, but do not cover pricing optimization in combination with inventory control.

So far there has been no analysis of the pricing problem in combination that allows for a general customer choice model with stochastic demand, considers both price and segmentation criteria in pricing optimization, and uses a dynamic inventory control mechanism. This thesis fills this gap.

In this chapter we will first give a definition of what a product actually is in the airline world and discuss the ideas behind some of the classical properties such a product can have. We will explain how restrictions are used to divide the market into different customer segments and how O&D pricing can influence customer behavior.

We will then introduce some of the sales mechanisms that are widely–used at the moment and the practical limitations they imply.

In the last section of this chapter we will formulate the pricing problem as a NLP. The formulation will be somewhat vague and abstract at first, because it is not clear right away what exactly the objective function for this problem is and how it can be computed. Nevertheless, we can establish a number of properties we expect the objective function to have based on the solution techniques we ultimately mean to use in order to solve the problem computationally. We will give a precise description of this objective function later, using the definitions and methods introduced in Chapter 3 about capacity control.

4.1 Products

In this section we explain what exactly an air travel product is, both from the airline's and from the customer's perspective. In addition, we will give a purely mathematical definition that we will use for the formulation of the optimization problem in Section 4.2 and in the definition of our customer model in Section 4.2.3. First and foremost an airline product is a service that allows the passenger to travel from A to B (and usually back to A). This does not always happen on a direct flight, but includes connections of multiple flights for each direction. Because of the diverse needs and expectations of different types of customers, products differ in a large number of ways. Some characteristics such as customer service or perceived security are directly associated with the airline, while others like departure and arrival times are features of the itinerary, and rebooking conditions and price are tied to the specific fare. Below is a non-exhaustive list of product characteristics as observed by customers:

Travel Basic attributes of the trip itself

- Origin, destination
- Departure time, arrival time, total travel time
- Routing (direct or transfer)

Price Airfare the passenger has to pay for the product.

Flexibility Restrictions that limit the flexibility of the customer w.r.t. a change of travel plans. Options range from non-flexible fares that can neither be re-booked nor refunded, to semi-restricted fares that can be changed for a fee, to fully flexible fares that can be re-booked and refunded free of charge at any time.

Quality of connection If the trip includes a transfer, there are a number of possible inconveniences for the customer:

- A large number of stops
- Very long or extremely short layovers
- Having to travel from one airport to another in order to get to the connecting flight.

Carrier's reputation The image of an airline is influenced, among other things, by:

- Friendliness, response time and competence of customer service
- Perceived security

Carrier's offer A large offer, both time-wise and geographically, is mostly important for business customers.

- A **large choice of connections** for the desired O&D is particularly important to business travelers who purchase flexible tickets, because it gives them the freedom to change their travel plans by a few hours, without having to wait a full day for the next flight.
- A **large network** is mostly relevant to long-term customers who choose the same carrier whenever they can (possibly complemented by a frequent flyer program), and companies who have a separate agreement with the airline that grants them special benefits.

Comfort Convenience of the trip as a whole, influenced by:

- Quality and characteristics of equipment, such as comfortable seats, low noise and good in-flight entertainment
- Simple and quick check-in and boarding

Availability and applicability This is not so much a feature that determines the value of a product to the customer but more a property that limits the customers' opportunities to buy or use the product, e.g. :

- Restrictions on the travel dates
- *Minimum stay* restrictions that force the customer to stay at their destination for a number of days or over a weekend before taking their return flight
- Restrictions limiting availability to certain passenger groups such as military personnel, students or senior citizens.

Product attributes are either

continuous quantities, such as price or potential fees,

integers, representing actual counts, such as a minimum stay restriction specifying the corresponding number of days, or

categorical variables, representing one of a number of choices, such as the availability to a certain group of customers or a flag indicating a *Saturday night stay*-restriction.

In order to describe the airline pricing problem mathematically, we will use the following definition:

Definition 4.1.1 A *product* is defined by the set of resources consumed on the network and its product characteristics. The former can be expressed as a vector $a = (a_r)_{1 \le r \le m}$ where a_r is the number of units of resource r consumed by the product, while the latter can be expressed as an $N_{\mathbf{P}}$-dimensional vector $\mathfrak{p} \in \mathbf{P}$ where each component of $\mathfrak{p}$ encodes information about product characteristics as described above.

The *product space* $\mathbf{P} \subseteq \mathbb{R}^{N_{\mathbf{P}}}$ is the set from which possible products are chosen. In general $\mathbf{P}$ need not be connected, for example due to the presence of integer variables.

Remark 4.1.2 Usually the discrete values for categorical variables are coded by integers. For optimization purposes these variables—and possibly other integer variables—need to be transformed, using outer convexification, into a set of binary variables, that are indicators for each state.

Remark 4.1.3 Consider a fixed market, defined by an O&D or a routing, and a POS. Many of the properties described above can be directly or indirectly influenced by the airline. From a pricing standpoint however, many attributes must be assumed to be given and fixed. A fare for a certain itinerary on this market is composed of a price, attributes defining the flexibility of the product and rules specifying availability and applicability. None of these characteristics actually influence the customer's experience during the flight itself. They are all abstract attributes that can be changed arbitrarily without the need to make changes on an operational level.

Still, in order to accurately describe customer choice behavior, we cannot discard the fixed attributes, such as quality of connection, from our model.

4.2 The Pricing NLP

In this section we will formulate the network airline pricing problem as a nonlinear optimization problem of the form (**NLP**). Generically, the pricing problem is the problem of defining a set of products that maximize overall revenue given optimal availability control. In other words, it can be written as

$$\max_{\mathcal{P}} \mathbf{r}(\mathcal{P}), \tag{4.1}$$

where the objective function $\mathbf{r}(\cdot)$ maps a set of products $\mathcal{P}$ to the overall expected revenue across the whole network assuming optimal availability control during the booking horizon. In the following we will describe the feasible set that $\mathcal{P}$ is chosen from as well as the objective function $\mathbf{r}$ in more detail. First, we will fix the following notation:

Definition 4.2.1 We again model the *booking horizon* as an interval $[0, T]$, where $t = 0$ is the beginning of the booking horizon and departure is at $t = T$. Denote by $R = \{1, \dots, m\}$ the set of *resources* and by $C \in \mathbb{N}^m$ the vector of *initial capacities* for each resource.

Definition 4.2.2 An *itinerary* I is a path on the network that can be traveled by the customer. It is described by a vector $a^I = (a^I_r)_{1 \le r \le m}$ where a^I_r is the number of units of resource r consumed by the itinerary.

We denote by $\mathbf{I}$ the set of all itineraries that are to be considered in the pricing and revenue management processes. Due to the large number of possible connections in a large airline network, in practice this is often a subset of the whole set of itineraries that can be traveled on the network, usually including those with highest expected demand.

In order to obtain a well-defined optimization problem, we first need to fix the *universal set of products*, which is the set we can choose products from.

Definition 4.2.3 For a fixed itinerary I, a *product* is defined by its product characteristics. It can be expressed as a vector $\mathfrak{p} \in \mathbf{P}$ where each component of $\mathfrak{p}$ encodes information about product attributes as described in Section 4.1, most importantly price and rebooking and refund flexibility.

The *product space* $\mathbf{P} \subseteq \mathbb{R}^{N_\mathbf{P}}$ is the set from which possible products are chosen. In general $\mathbf{P}$ need not be connected. In particular, some of the product variables, such as indicator variables for restrictions, can be binary or integer variables.

A *network product* is a pair $(I, \mathfrak{p})$ of an itinerary and a vector of product attributes.

Remark 4.2.4 Let $\mathbf{P}$ be a product space. We will often make the following assumptions:

(1) The set of feasible values for each attribute $\mathfrak{p}_i$ is independent of the values of other attributes, in other words the product space is a cartesian product $\mathbf{P} = \times_{i=1}^{N_\mathbf{P}} \mathbf{P}_i$.

(2) The first $N_\mathbf{P}^{\text{cont}}$ attributes are continuous and the corresponding feasible sets are intervals

$$\mathbf{P}_i = \left[\mathfrak{p}_i^{\text{lb}}, \mathfrak{p}_i^{\text{ub}}\right] \qquad \forall 1 \le i \le N_\mathbf{P}^{\text{cont}}. \tag{4.2}$$

(3) The remaining $N_\mathbf{P}^{\text{int}} = N_\mathbf{P} - N_\mathbf{P}^{\text{cont}}$ ones have discrete values in the feasible sets

$$\mathbf{P}_i = \left\{\mathfrak{p}_{i,1}^{\text{d}}, \dots, \mathfrak{p}_{i,n_i}^{\text{d}}\right\} \subset \mathbb{R} \qquad \forall N_\mathbf{P}^{\text{cont}} < i \le N_\mathbf{P}. \tag{4.3}$$

Clearly, the assumption about the order of attributes is w.l.o.g.

Remark 4.2.5 Let $\mathcal{P} \subset \mathcal{P}' \subset \mathbf{I} \times \mathbf{P}$ be a finite sets of network products. Any control strategy for $\mathcal{P}$ is also a feasible strategy for $\mathcal{P}'$ with equal expected revenue and therefore $\mathbf{r}(\mathcal{P}') \ge \mathbf{r}(\mathcal{P})$. Moreover, except for very specific demand models one can always increase expected revenue by adding an additional product. In other words for a finite set of products $\mathcal{P} \subset \mathbf{P}$ there usually exists a superset $\mathcal{P} \subset \mathcal{P}' \subset \mathbf{P}$, such that $\mathbf{r}(\mathcal{P}') > \mathbf{r}(\mathcal{P})$. We therefore introduce a bound on the cardinality of the set of products: $|\mathcal{P}| \le M$. By adding irrelevant products to an optimal solution we can assume w.l.o.g that this constraint is satisfied with equality and therefore write the pricing problem as

$$\begin{aligned} \max_{\mathcal{P} \subset \mathbf{P}} \quad & \mathbf{r}(\mathcal{P}) \\ \text{s.t.} \quad & |\mathcal{P}| \;=\; M. \end{aligned} \tag{4.4}$$

4.2.1 Control variables

Definition 4.2.6 Let $\mathbf{P}$ be a product space of dimension $N_\mathbf{P}$, let M be a fixed number of products and $n = N_\mathbf{P} M$. We can trivially parametrize the set $\mathbf{P}^M$ by interpreting a real n-dimensional vector as a $N_\mathbf{P} \times M$ matrix. This gives us a map

$$\mathcal{P} : U \subseteq \mathbb{R}^n \xrightarrow{\sim} \mathbf{P}^M \tag{4.5a}$$

$$u \mapsto \mathcal{P}(u) = \begin{pmatrix} \mathfrak{p}_1^1(u) & \mathfrak{p}_1^2(u) & \cdots & \mathfrak{p}_1^M(u) \\ \mathfrak{p}_2^1(u) & \mathfrak{p}_2^2(u) & \cdots & \mathfrak{p}_2^M(u) \\ \vdots & \vdots & & \vdots \\ \mathfrak{p}_{N_\mathbf{P}}^1(u) & \mathfrak{p}_{N_\mathbf{P}}^2(u) & \cdots & \mathfrak{p}_{N_\mathbf{P}}^M(u) \end{pmatrix} \tag{4.5b}$$

$$\mathfrak{p}_i^k(u) = u_i^k := u_{N_\mathbf{P}(i-1)+k} \qquad \forall 1 \le k \le M, 1 \le i \le N_\mathbf{P} \tag{4.5c}$$

where each column vector of $\mathcal{P}(u)$ contains the attribute vector for one product. The first M components of u are mapped to the first attribute, the next M components the second one, etc.

With Definition 4.2.6 the nonlinear program Eq. (4.4) can be written as

$$\begin{aligned} \max_{u} \quad & \mathbf{r}(\mathcal{P}(u)) \\ \text{s.t.} \quad & \mathfrak{p}^k(u) \;\in\; \mathbf{P} \quad \forall 1 \le k \le M \end{aligned} \tag{4.6}$$

which, with the assumptions from Remark 4.2.4, becomes

$$\begin{array}{llllll} \max\limits_{u} & \mathbf{r}(\mathcal{P}(u)) & & & & \\ \text{s.t.} & \mathfrak{p}_i^{\text{lb}} \le & u_i^k & \le & \mathfrak{p}_i^{\text{ub}} & \forall 1 \le k \le M, 1 \le i \le N_{\mathbf{P}}^{\text{cont}} \\ & & u_i^k & \in & \mathbf{P}_i = \left\{\mathfrak{p}_{i,1}^{\text{d}}, \dots, \mathfrak{p}_{i,n_i}^{\text{d}}\right\} & \forall 1 \le k \le M, N_{\mathbf{P}}^{\text{cont}} < i \le N_{\mathbf{P}}. \end{array} \tag{4.7}$$

In order to obtain sensible continuous relaxations for the problem, we use a Special Ordered Set Type 1 (SOS1) formulation to model the discrete attributes. The parametrization of $\mathbf{P}^M$ is then

$$\mathcal{P} : U \subseteq \mathbb{R}^n \xrightarrow{\sim} \mathbf{P}^M \tag{4.8a}$$

$$u \mapsto \mathcal{P}(u) = \left(\mathfrak{p}_i^k(u)\right) \tag{4.8b}$$

$$\mathfrak{p}_i^k(u) = u_i^k \qquad \forall 1 \le k \le M, 1 \le i \le N_{\mathbf{P}}^{\text{cont}} \tag{4.8c}$$

$$\mathfrak{p}_i^k(u) = \sum_{j=1}^{n_i} u_{i,j}^k \mathfrak{p}_{i,j}^{\text{d}} \qquad \forall 1 \le k \le M, N_{\mathbf{P}}^{\text{cont}} < i \le N_{\mathbf{P}}, \tag{4.8d}$$

where $n = \left(N_{\mathbf{P}}^{\text{cont}} + \sum_{i=N_{\mathbf{P}}^{\text{cont}}+1}^{N_{\mathbf{P}}} n_i\right) M$. With Eq. (4.8) the NLP becomes

$$\begin{array}{llllll} \max\limits_{u} & \mathbf{r}(\mathcal{P}(u)) & & & & \\ \text{s.t.} & \mathfrak{p}_i^{\text{lb}} \le & u_i^k & \le & \mathfrak{p}_i^{\text{ub}} & \forall 1 \le k \le M, 1 \le i \le N_{\mathbf{P}}^{\text{cont}} \\ & & u_{i,j}^k & \in & \{0,1\} & \forall 1 \le k \le M, N_{\mathbf{P}}^{\text{cont}} < i \le N_{\mathbf{P}}, 1 \le j \le n_i \\ & & \sum\limits_{j=1}^{n_i} u_{i,j}^k & = & 1 & \forall 1 \le k \le M, N_{\mathbf{P}}^{\text{cont}} < i \le N_{\mathbf{P}} \end{array} \tag{4.9}$$

Product clusters

In some cases it is useful to restrict the feasible set of control vectors u such that the resulting set of products $\mathcal{P}(u)$ can be partitioned into a small number of product clusters.

Definition 4.2.7 A *product cluster* is a set of products $\left\{\mathfrak{p}^1, \dots, \mathfrak{p}^M\right\}$, such that the products only differ in their price. In other words, with $\mathfrak{p}_1$ again being price for every product $\mathfrak{p}$,

$$\mathfrak{p}_i^k = \mathfrak{p}_i^{k'} \qquad \forall 1 \le k < k' \le M, 2 \le i \le N_{\mathbf{P}}. \tag{4.10}$$

This way, each product is defined by its price and the attributes of the corresponding cluster.

For a fixed number of product clusters M^{cluster}, there are two possible ways to model the corresponding optimization problem.

We can allocate a fixed number of products M_l to each product cluster $l = 1, \dots, M^{\text{cluster}}$. Let $l(k)$ denote the product cluster that product k is assigned to. We can then write the NLP as

$$\begin{array}{llllll} \max\limits_{u,v} & \mathbf{r}(\mathcal{P}(u,v)) & & & & \\ \text{s.t.} & \mathfrak{p}_1^{\text{lb}} \le & u^k & \le & \mathfrak{p}_1^{\text{ub}} & \forall 1 \le k \le M \\ & \mathfrak{p}_i^{\text{lb}} \le & v_i^l & \le & \mathfrak{p}_i^{\text{ub}} & \forall 1 \le l \le M^{\text{cluster}}, 2 \le i \le N_{\mathbf{P}}^{\text{cont}} \\ & & v_i^l & \in & \mathbf{P}_i = \left\{\mathfrak{p}_{i,1}^{\text{d}}, \dots, \mathfrak{p}_{i,n_i}^{\text{d}}\right\} & \forall 1 \le l \le M^{\text{cluster}}, N_{\mathbf{P}}^{\text{cont}} < i \le N_{\mathbf{P}} \end{array} \tag{4.11}$$

where the map $\mathcal{P}$ is given by

$$\mathfrak{p}_1^k(u,v) = u^k \qquad \forall 1 \le k \le M \tag{4.12a}$$

$$\mathfrak{p}_i^k(u,v) = v_i^{l(k)} \qquad \forall 1 \le k \le M, 2 \le i \le N_{\mathbf{P}} \tag{4.12b}$$

Here, the number of control variables is $n = M + (N_{\mathbf{P}} - 1)M^{\text{cluster}}$, u^k is the price for product k and v^l is the vector of attributes for cluster l, where the index along the attribute dimension runs from

2 to $N_{\mathbf{P}}$ in order to keep the notation consistent. We can again transform each discrete variable v_i^l to binary variables $v_{i,1}^l, \dots, v_{i,n}^l$ analogously to Eqs. (4.8) and (4.9) to obtain

$$
\begin{array}{lllllll}
\max\limits_{u,v} & \mathbf{r}(\mathcal{P}(u,v)) & & & & & \\
\text{s.t.} & \mathfrak{p}_1^{\text{lb}} & \le & u^k & \le & \mathfrak{p}_1^{\text{ub}} & \forall 1 \le k \le M \\
& \mathfrak{p}_i^{\text{lb}} & \le & v_i^k & \le & \mathfrak{p}_i^{\text{ub}} & \forall 1 \le l \le M^{\text{cluster}}, 2 \le i \le N_{\mathbf{P}}^{\text{cont}} \\
& & & v_{i,j}^l & \in & \{0,1\} & \forall 1 \le l \le M^{\text{cluster}}, N_{\mathbf{P}}^{\text{cont}} < i \le N_{\mathbf{P}}, 1 \le j \le n_i \\
& & & \sum\limits_{j=1}^{n_i} v_{i,j}^l & = & 1 & \forall 1 \le l \le M^{\text{cluster}}, N_{\mathbf{P}}^{\text{cont}} < i \le N_{\mathbf{P}}
\end{array}
\tag{4.13}
$$

with

$$\mathfrak{p}_1^k(u,v) = u^k \qquad \forall 1 \le k \le M \tag{4.14a}$$

$$\mathfrak{p}_i^k(u,v) = v^l(k)_i \qquad \forall 1 \le l \le M, 2 \le i \le N_{\mathbf{P}}^{\text{cont}} \tag{4.14b}$$

$$\mathfrak{p}_i^k(u,v) = \sum_{j=1}^{n_i} v_i^{l(k)} \mathfrak{p}_{i,j}^{\text{d}} \qquad \forall 1 \le l \le M, N_{\mathbf{P}}^{\text{cont}} < i \le N_{\mathbf{P}}. \tag{4.14c}$$

Alternatively to a fixed allocation of products to clusters, additional binary variables w_l^k can be used to specify if product k belongs to cluster l. For fixed number of products M and number of clusters M^{cluster}, the map $\mathcal{P}$ and the NLP are then given by

$$\mathfrak{p}_1^k(u,v,w) = u^k \qquad \forall 1 \le k \le M \tag{4.15a}$$

$$\mathfrak{p}_i^k(u,v,w) = \sum_{l=1}^{M^{\text{cluster}}} w_l^k v_i^l \qquad \forall 1 \le k \le M, 2 \le i \le N_{\mathbf{P}} \tag{4.15b}$$

$$
\begin{array}{lllllll}
\max\limits_{u,v,w} & \mathbf{r}(\mathcal{P}(u,v,w)) & & & & & \\
\text{s.t.} & \mathfrak{p}_1^{\text{lb}} & \le & u^k & \le & \mathfrak{p}_1^{\text{ub}} & \forall 1 \le k \le M \\
& \mathfrak{p}_i^{\text{lb}} & \le & v_i^l & \le & \mathfrak{p}_i^{\text{ub}} & \forall 1 \le l \le M^{\text{cluster}}, 2 \le i \le N_{\mathbf{P}}^{\text{cont}} \\
& & & v_i^l & \in & \mathbf{P}_i = \left\{\mathfrak{p}_{i,1}^{\text{d}}, \dots, \mathfrak{p}_{i,n_i}^{\text{d}}\right\} & \forall 1 \le l \le M^{\text{cluster}}, N_{\mathbf{P}}^{\text{cont}} < i \le N_{\mathbf{P}} \\
& & & w_l^k & \in & \{0,1\} & \forall 1 \le l \le M^{\text{cluster}}, 1 \le k \le M \\
& & & \sum\limits_{l=1}^{M^{\text{cluster}}} w_l^k & = & 1 & \forall 1 \le k \le M.
\end{array}
\tag{4.16}
$$

An SOS1 formulation for the discrete cluster attributes in Eqs. (4.15) and (4.16) is analogous to Eqs. (4.14) and (4.13).

4.2.2 Objective function

Following the objective of classic RM, the objective of the pricing problem is to maximize short term expected revenue. In particular, we do not assume customer loyalty (which is often modeled via a customer lifetime value). We also do not consider competitor reactions in a repeated competitive game.

Clearly, given a set of products $\mathcal{P}$, the expected revenue $\mathbf{r}(\mathcal{P})$ that can be achieved heavily depends on the assumed customer model, the booking control mechanism and a suitable optimization algorithm. In order to evaluate the objective function value of the pricing problem for a given solution we have to solve the underlying capacity control problem. For computational reasons we would like to be able to compute the optimal expected revenue efficiently and deterministically. In order to use gradient based optimization, we also need to be able to compute the gradient of expected revenue w.r.t. input parameters.

In our choice of control mechanism and solution algorithms for the capacity control problem we limit ourselves to dynamic methods, which react to the realization to the demand process over the course of the booking horizon. Although working with deterministic solution methods would greatly simplify the pricing problem, we believe that the additional effort is justified for several reasons. Firstly, dynamic control mechanisms provenly lead to better results and are widely used in industry practice. Secondly, the benefit of optimally choosing the fare structure and particularly of prices arguably is higher when combined with dynamic capacity control mechanisms: Consider a special case of pricing optimization that is only concerned with selecting of optimal price points for products whose other attributes are fixed and given. Having different price points for otherwise identical products (i.e. products within a product cluster) allows to

- exploit varying expected customer behavior over the course of the booking horizon, particularly time-varying willingness-to-pay,
- dynamically react to the variance in demand volume and customer behavior.

While the former can be achieved with both static and dynamic control mechanisms, only dynamic models benefit natively from the latter. In practice, static control mechanisms are often turned into semi-dynamic ones by using frequent re-optimization during the booking horizon, always taking into account the materialization of demand up to the current point in time. For such an optimization scheme, the expected revenue cannot be computed easily at the beginning of the booking horizon. Depending on the specific control scheme and optimization method, the deterministic optimization will at best yield an upper bound for the overall expected revenue, sometimes only a rough estimate. We will therefore focus entirely on dynamic control mechanisms and, consequently, dynamic optimization methods.

It is well-known that the relative performance of different heuristic control mechanisms and optimization methods change depending on

- demand volume,
- demand model,
- model parameters, particularly those related to willingness-to-pay,
- and product structure.

Primarily due to the last point, heuristic solutions of the underlying capacity control problem are impractical for the pricing problem, because the quality of the heuristic would depend on the current solution candidate in the optimization.

We use the well-known dynamic programming formulation of the dynamic inventory control problem that was presented in Section 3.4.3, because it is the only dynamic model that can efficiently be solved to optimality under fairly mild assumptions. In order to simplify notation, we will mostly work with the choice-based DP, although for computational purposes one would use the demand transformation (Section 3.5) to obtain an independent demand model. Instead of the standard time-discrete model we will use the continuous time DP that is presented in Chapter 5.

Combining Eqs. (4.7) and (3.47), we can write the single resource pricing problem as

$$\max_{u} \quad V_C(0) \tag{4.17a}$$

$$\text{s.t.} \quad \dot{V}_c(t) = -\max_{\mathcal{S}\subseteq\mathcal{P}(u)} \sum_{\mathfrak{p}\in\mathcal{S}} \lambda_{\mathfrak{p}}(\mathcal{S},t)\,[\mathbf{y}_{\mathfrak{p}} - \pi_c(t)] \qquad \forall 1 \le c \le C \tag{4.17b}$$

$$\pi_c(t) = \begin{cases} V_1(t) & \text{if } c = 1, \\ V_c(t) - V_{c-1}(t) & \text{else} \end{cases} \tag{4.17c}$$

$$V_c(T) = 0 \qquad \forall 1 \le c \le C. \tag{4.17d}$$

$$u_i^k \in [\mathfrak{p}_i^{\mathrm{lb}}, \mathfrak{p}_i^{\mathrm{ub}}] \qquad \forall 1 \le k \le M, 1 \le i \le N_{\mathbf{P}}^{\mathrm{cont}} \tag{4.17e}$$

$$u_i^k \in \left\{\mathfrak{p}_{i,1}^{\mathrm{d}}, \dots, \mathfrak{p}_{i,n_i}^{\mathrm{d}}\right\} \qquad \forall 1 \le k \le M, N_{\mathbf{P}}^{\mathrm{cont}} < i \le N_{\mathbf{P}}, \tag{4.17f}$$

where $\lambda_{\mathfrak{p}}(\mathcal{S},t)$ is the demand rate at time to departure t for product $\mathfrak{p}$ given the set of alternatives $\mathcal{S}$. This rate depends on u, because $\mathcal{P}$ does and $\mathfrak{p} \in \mathcal{S} \subseteq \mathcal{P}(u)$.

Here the maximum in the RHS of Eq. (4.17b) is taken over all subsets of the set of products $\mathcal{P}(u)$. In general the set of feasible actions $\mathbb{S}$ might only be a subset of the power set of $\mathcal{P}(u)$. The rules that exclude certain availability configurations from $\mathbb{S}$ usually depend on product characteristics and can for example be modeled via additional constraints of the form

$$h(\mathcal{S}, \mathcal{P}(u), t) \geq 0. \tag{4.18}$$

Note that these restrictions do not depend on the states V and are therefore constraints for the inner optimization problem in the RHS of Eq. (4.17b) rather than constraints for the DP Eq. (4.17a). Details of the solution algorithm, including efficient derivative computation, are presented in in Chapter 5.

For the network case with multiple resources, we simply use the network dynamic program Eq. (3.52) instead of the single-leg DP. Due to the curse of dimensionality, this network DP cannot be solved to optimality. In Chapter 6 we therefore present a heuristic approach based on dual decomposition to approximate its objective function value and solution sensitivity w.r.t. the constant parameters defining the products.

4.2.3 Demand model

We can see from Eq. (4.17a) that theoretically we can use any demand model that is compatible with the product space $\mathbf{P}$ in the sense that for any finite offer set $\mathcal{S} \subset \mathbf{P}$ and every product $\mathfrak{p} \in \mathcal{S}$ there is a well-defined demand rate $\lambda_{\mathfrak{p}}(\mathcal{S}, t)$. In practice, however, we want to use gradient based optimization and therefore require the demand model to satisfy the following:

Property 2 For any time t, finite set of products $\mathcal{S} \subset \mathbf{P}$ and product $\mathfrak{p} \in \mathcal{S}$ the demand rate $\lambda_{\mathfrak{p}}(\mathcal{S}, t)$ as well as its gradient w.r.t. the product characteristics can be computed *efficiently* and *deterministically.*

The DP formulation of the availability control problem depends on the assumption that demand is memoryless, which means that the arrival process is modeled as an inhomogeneous Poisson process with rate $\lambda(t)$. The booking process is the multivariate filtered process induced by independent booking decisions. As described in Section 2.2, the demand model is entirely defined by the arrival rate and booking probabilities $d_{\mathfrak{p}}(\mathcal{S}, t)$ via

$$\lambda_{\mathfrak{p}}(\mathcal{S}, t) = \lambda(t) d_{\mathfrak{p}}(\mathcal{S}, t). \tag{4.19}$$

In most cases the arrival rate $\lambda(t)$ is given as a piecewise smooth or piecewise constant function. Therefore the main concern is the modeling and computation of the booking probabilities $d_{\mathfrak{p}}(\mathcal{S})$ at a fixed point in time.

In most applications that make use of discrete choice theory the goal is to model choice behavior of a partially unknown individual when confronted with a subset of a fixed set of alternatives $\mathcal{S} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_M\}$. For example, the researcher could be interested in the distribution of choices a customer makes between automobiles, depending on demographic data such as gender, income, or location. In this case the researcher knows the fixed attributes of the alternatives that are available on the market. The attributes defining the decision maker are random variables and the researcher knows or has an estimate of their joint conditional distribution given certain demographic factors. This allows the researcher to use arbitrary labels for the alternatives and construct a model using these labels to link random effects to each product or to each customer/product combination.

Example 4.2.1 Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ denote the finite set of customers and $\mathcal{P} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_M\}$ the finite set of products that are included in a study. A standard linear utility model is given by

$$u_{i,k} = \theta^{\top} \mathbf{x}_i + \eta^{\top} \mathfrak{p}_k + \epsilon_{i,k}, \tag{4.20}$$

where $u_{i,k}$ is the utility of product k to customer i, the vectors $\theta^{\top}$ and $\eta^{\top}$ are the linear coefficients for customer and product attributes respectively, and $\epsilon_{i,k}$ is a random error term. All ϵ are assumed to be IID random variables with an expected value of 0.

In the problem at hand the goal is to optimally define the characteristics of the alternatives, which especially means that attributes are not fixed. During each booking decision the customer is confronted with a set of anonymous alternatives that are purely described by their attributes, which means that after a permutation of the alternatives we would expect a fixed customer to still choose the same product. We can thus not allow the choice to depend on random effects that are tied to arbitrary labels. Translating this idea to the corresponding aggregate model we see that the demand function has to satisfy the following:

Property 3 Let $\mathcal{P} = (\mathfrak{p}_1, \dots, \mathfrak{p}_M)$ be a product matrix and $t \in [0, T]$. Then for any permutation of M elements $\sigma \in S_M$ and every $k = 1, \dots, M$:

$$\lambda_{\mathfrak{p}_{\sigma(k)}}(t, (\mathfrak{p}_{\sigma(1)}, \dots, \mathfrak{p}_{\sigma(M)})) = \lambda_{\mathfrak{p}_k}(t, \mathcal{P}). \tag{4.21}$$

For this reason, the standard models such as MNL (see Section 2.2.1) that are often used in choice-based RM cannot be applied here. Furthermore, it is common in the literature to describe the whole market in a single model, tightly integrating supply and demand. In contrast to this approach, we will strictly separate supply and demand, which not only ensures that Eq. (4.21) is satisfied but also allows us to disaggregate the set of potential customers.

In practice, customers can often be clustered into groups, called *customer types*, so that customers of the same type tend to behave similarly, while customers of different type might have completely different decision rules. It is possible to cover these differences in choice behavior in a single customer model, purely using customer attributes to distinguish customer types, but usually at the expense of additional customer variables and high correlation between attributes.

Example 4.2.2 In the airline case the classic example is the separation of

Leisure customers, who often book early, are very price sensitive and are tolerant towards product restrictions, and

Business customers, who usually book late during the booking horizon, have a high willingness-to-pay and are sensitive to product restrictions.

A customer model describing only leisure customers might completely ignore penalties for abstract restrictions such as *weekend-stay* or *non-refundability* and focus only on willingness-to-pay. Each customer is then described by a single random variable indicating their budget and the utility function is the difference of budget and price, which means that the customer will always purchase the cheapest available product they can afford. The customer type is uniquely defined by the probability distribution the budget attribute.

Business customers, on the other hand, have to be modeled in more detail, because their reaction to differences in product restrictions has to be reflected in the model. As described in Section 2.1.3, there are various ways to do this. The most common approach is to associate a *disutility cost* to each product attribute and have it enter linearly into the utility function. The disutilities can be constants, but can also depend on customer attributes that describe a customers sensitivity to product restrictions. In the latter case a customer's response to varying product characteristics is random, but it is often reasonable to assume that the associated customer variables are independent from each other and other characteristics such as willingness-to-pay, which greatly simplifies numerical aggregation and estimation for the model.

Assume both models given above are accurate for the respective customer types. Considering the union of both customer types as a single group is of course simple, at least theoretically. Each leisure customer can be seen as a business customer with disutility costs of zero for all product restrictions. However, since tourists generally have a lower budget, an unknown random customer's willingness-to-pay is now positively correlated with their disutility costs. In other words, customer attributes are only conditionally independent given a decision maker's customer type.

In addition, the distribution of each individual attribute can become much more difficult to handle. For example, if the willingness-to-pay for each customer type follows an exponential distribution with different rate parameters, the willingness-to-pay for the joint type does not follow an exponential distribution.

We will not restrict ourselves to one specific model but allow the use of a finite number of different models that are compatible with the pricing structure we are optimizing. This kind of disaggregation has several advantages:

Independence of attributes. As described above, assuming conditional independence between product attributes given a customer type is much less restrictive than an independence assumption among all customers.

Simpler models. Behavior specific to a certain group can be captured much more efficiently, or vice versa: Given a fixed amount of complexity each type can be described more accurately using a specially tailored set of attributes and decision rule. The reduced number of model parameters will not only reduce computational complexity but also lead to much more stable estimates.

Constant models. In practice, overall behavior of a random customer changes over the course of the booking horizon. In order to capture this effect, a single model would have to use a time-dependent distribution of customer attributes, which is difficult to handle numerically and statistically. Using a collection of different customer types, this shift in overall behavior can largely be accounted for by a shift in the mix of customers over the booking period. This means that for each customer type the only time-dependent quantity is the arrival rate, while the distribution of customer attributes remains constant.

Specially tailored customer models for different customer types can be constructed in different ways:

- Based on a priori information about customer behavior that is known to the researcher or business analyst.
- Using iterative model generation based on historical data, either using feature selection to add relevant attributes one by one, or using model reduction techniques to remove irrelevant criteria from an initially very complex model.

In the following, we will focus on customer choice models using a deterministic decision rule based on a deterministic utility function, although the framework extends easily to general customer models. In practice, a demand model depends on model parameters that have to be specified by the researcher or estimated from historical data. In particular, coefficients in the utility function and the joint distribution of customer attributes are controlled by model parameters. In order to simplify notation we will omit these parameters in the following sections.

Definition 4.2.8 A *customer type* $\mathbf{T}$ for the product space $\mathbf{P}$ is a triple $(\mathbf{X}, u, \lambda)$.

- $\mathbf{X}$ is a multivariate $\mathbf{C}$–valued random variable $\mathbf{X}$ that holds information about *customer preferences* such as
  - Favorite departure time
  - Inclination to cancel
  - Inclination to change reservations
  - Willingness-to-pay
  - Carrier preference
  - Importance of short travel time,

  where $\mathbf{C} \subseteq \mathbb{R}^n$ is the *customer space*. In general, the components of $\mathbf{X}$ need not be independent. If it exists, we will denote the mixed joint density function of $\mathbf{X}$ by $f\colon \mathbf{C} \to \mathbb{R}_{\geq 0}$.

- u is a *utility function*

$$u\colon \mathbf{C} \times \mathbf{P} \to \mathbb{R} \tag{4.22a}$$

$$(\mathbf{x}, \mathfrak{p}) \mapsto u(\mathbf{x}, \mathfrak{p}) \tag{4.22b}$$

  mapping a customer $\mathbf{x}$ and a product $\mathfrak{p}$ to a a real value that measures the utility of the product to the particular customer. We assume that u is continuously differentiable with respect to both $\mathbf{x}$ and $\mathfrak{p}$ where applicable (i.e. w.r.t. continuous variables).

- λ is a function yielding the *arrival rate* for the customer type at any given time during the booking horizon
$$\lambda\colon [0,T] \to R_{\geq 0}. \tag{4.23}$$

We assume that λ is piecewise continuously differentiable w.r.t. t.

Given the choice between the set of products $\mathcal{S}$, a customer $\mathbf{x} \in \mathbf{C}$ will evaluate their personal utility function $u(\cdot) = u(\mathbf{x}, \cdot)$ for all products $\mathfrak{p} \in \mathcal{S}$ and purchase the product that maximizes this utility unless all utilities are negative, in which case the customer will not buy anything at all. In order to break ties, we denote by $\succ$ the natural lexicographic ordering on $\mathbf{P}$. If there is no unique best product, the customer chooses between all products whose utility is maximal the one that is minimal with respect to $\succ$.

For a fixed product $\mathfrak{p} \in \mathbf{P}$, the utility for an unknown customer is a random variable $u(\mathbf{X}, \mathfrak{p})$ with values in $\mathbb{R}$. For two products $\mathfrak{p}_1, \mathfrak{p}_2$, $u(\mathbf{X}, \mathfrak{p}_1)$ and $u(\mathbf{X}, \mathfrak{p}_2)$ are generally not independent random variables.

Definition 4.2.9 Let $\mathbf{P}$ be a product space and $\mathbf{T} = (\mathbf{X}, u, \lambda)$ a customer type for $\mathbf{P}$. Two products $\mathfrak{p}_1 \neq \mathfrak{p}_2$ are called $\mathbf{T}$-*independent* if and only if
$$\mathrm{P}[u(\mathbf{X}, \mathfrak{p}_1) = u(\mathbf{X}, \mathfrak{p}_2)] = 0. \tag{4.24}$$
A set of products $\mathcal{P} = \{\mathfrak{p}_1, \dots, \mathfrak{p}_M\} \subseteq \mathbf{P}$ is called $\mathbf{T}$-*independent* if the elements of $\mathcal{P}$ are pairwise $\mathbf{T}$-independent.

Remark 4.2.10 This means that a random customer drawn from $\mathbf{T}$ will almost surely prefer one of the products over the others. Hence the probability of a random customer choosing product $\mathfrak{p} \in \mathcal{P}$ does not depend on the choice of the ordering $\succ$.

Definition 4.2.11 Let $\mathcal{S} \subseteq \mathbf{P}$ be an offer set. We introduce an *outside product* or *outside good* $\mathfrak{p}_0$ that indicates - from our perspective - that a customer chooses to purchase nothing at all. In reality the customer might buy a product offered by a competitor or one of our own products that we (falsely) assume to be independent of $\mathcal{S}$.

We define the *extended offer set* $\bar{\mathcal{S}}$ as $\bar{\mathcal{S}} = \mathcal{S} \cup \{\mathfrak{p}_0\}$.

By definition the utility of $\mathfrak{p}_0$ is always zero, i.e.
$$\forall \mathbf{x} \in \mathbf{C} : u(\mathbf{x}, \mathfrak{p}_0) = 0. \tag{4.25}$$

Definition 4.2.12 Let $\mathbf{P}$ be a product space and $\mathbf{T} = (\mathbf{X}, u, \lambda)$ a customer type for $\mathbf{P}$. Let $\mathcal{S} \subseteq \mathbf{P}$ be an offer set and $\bar{\mathcal{S}} = \mathcal{S} \cup \{\mathfrak{p}_0\}$ the extended offer set. For a customer $\mathbf{x} \in \mathbf{C}$ we denote by $\mathfrak{p}^*(\mathbf{x}, \mathcal{S})$ the customer's *choice*, i.e. the product that the customer chooses when offered the set of alternatives $\mathcal{S}$. We denote by $\mathcal{P}^*(\mathbf{x}, \mathcal{S})$ the set of products that maximize their personal utility:
$$\mathcal{P}^*(\mathbf{x}, \mathcal{S}) = \{\mathfrak{p} \in \bar{\mathcal{S}} \mid \forall \mathfrak{p}' \in \bar{\mathcal{S}} : u(\mathbf{x}, \mathfrak{p}) \geq u(\mathbf{x}, \mathfrak{p}')\}. \tag{4.26}$$
If there is a unique optimal product for the customer, i.e. $|\mathcal{P}^*(\mathbf{x}, \mathcal{S})| = 1$, the customer will choose this option and we call it $\mathfrak{p}^*(\mathbf{x}, \mathcal{S})$. In case of a tie the ordering $\succ$ is used to determine $\mathfrak{p}^*(\mathbf{x}, \mathcal{S})$:
$$\mathfrak{p}^*(\mathbf{x}, \mathcal{S}) = \min_{\succ} \mathcal{P}^*(\mathbf{x}, \mathcal{S}) \tag{4.27}$$

Note that the case $\mathfrak{p}' = \mathfrak{p}_0$ is included in Eq. (4.26), which means that the utility of a product hast to be non-negative in order for the customer to buy it.

Aggregation

In this section, let always $f\colon \mathbf{C} \to \mathbb{R}$ denote the generalized density function for the distribution of the customer population $\mathbf{X}$.

As described in Section 2.2, because we assume that each customer has a deterministic utility function and decision rule, the aggregation problem can be decomposed. In the first step, the set of customers choosing a certain product is computed. In the second step, one computes the measure of this set w.r.t. the generalized density function f.

Definition 4.2.13 Let $\mathbf{P}$ be a product space and $\mathbf{T} = (\mathbf{X}, u, \lambda)$ a customer type for $\mathbf{P}$. Let $\mathcal{S} \subseteq \mathbf{P}$ be an offer set. For a product $\mathfrak{p} \in \mathcal{S}$, the *customer set* $\mathcal{X}_{\mathfrak{p}}(\mathcal{S})$ is the set of customers who choose product $\mathfrak{p}$ when offered the set of alternatives $\mathcal{S}$. It is given by

$$\mathcal{X}_{\mathfrak{p}}(\mathcal{S}) = \{x \in \mathbf{C} \mid \mathfrak{p}^*(\mathbf{x}, \mathcal{S}) = \mathfrak{p}\}. \tag{4.28}$$

Remark 4.2.14 By definition we have

$$\bigcup_{\mathfrak{p} \in \bar{\mathcal{S}}} \mathcal{X}_{\mathfrak{p}}(\mathcal{S}) = \mathbf{C}. \tag{4.29}$$

Definition 4.2.15 Let $\mathbf{P}$ be a product space and $\mathbf{T} = (\mathbf{X}, u, \lambda)$ a customer type for $\mathbf{P}$. Let $\mathcal{S} \subseteq \mathbf{P}$ be an offer set. If the set $\bar{\mathcal{S}}$ is $\mathbf{T}$-independent, the booking probability for product $\mathfrak{p} \in \mathcal{S}$ is given by

$$d_{\mathfrak{p}}(\mathcal{S}) = \mathrm{P}\left[\mathfrak{p}^*(\mathbf{X}, \mathcal{S}) = \mathfrak{p}\right] = \int_{\mathcal{X}_{\mathfrak{p}}(\mathcal{S})} f(\mathbf{x}) d\mathbf{x}. \tag{4.30}$$

Given multiple customer types $\mathbf{T}_1, \ldots, \mathbf{T}_L$, we denote by $d^l_{\mathfrak{p}}(\mathcal{S})$ the probability of a random customer from customer type $\mathbf{T}_l$ choosing product $\mathfrak{p}$.

Remark 4.2.16 Note that in the situation of Definition 4.2.15, due to the existence of an outside good, the booking probabilities for $\mathfrak{p} \in \mathcal{S}$ do not necessarily add up to one. With Eq. (4.29) we have

$$\sum_{\mathfrak{p} \in \bar{\mathcal{S}}} d_{\mathfrak{p}}(\mathcal{S}) = 1. \tag{4.31}$$

Definition 4.2.17 Let $\mathbf{P}$ be a product space and $\mathcal{T} = \{\mathbf{T}_1, \ldots, \mathbf{T}_L\}$ a set of customer types for $\mathbf{P}$ with $\mathbf{T}_l = (\mathbf{X}^l, u^l, \lambda^l)$. Let $\mathcal{S} \subseteq \mathbf{P}$ be an offer set. The *arrival rate* $\lambda_{\mathfrak{p}}(\mathcal{S}, t)$ for product $\mathfrak{p} \in \mathcal{S}$ at time t is given by

$$\lambda_{\mathfrak{p}}(\mathcal{S}, t) = \sum_{l=1}^{L} \lambda^l(t) d^l_{\mathfrak{p}}(\mathcal{S}). \tag{4.32}$$

Here, the booking probabilities for each customer type are constant over time, but the mix of customers varies according to the time-dependent arrival rates $\lambda_1, \ldots, \lambda_L$. Note that, independent of the time-discretization, the integral Eq. (4.30) only has to be evaluated L times in order to compute all $\lambda_{\mathfrak{p}}(\mathcal{S}, t)$.

Chapter 5

The single-leg inventory control problem

In this section we will present a formulation of the single-leg dynamic inventory control problem as a stochastic optimal control problem with a terminal constraint on the states. By applying Bellman's principle of optimality one obtains an ODE, the Hamilton-Jacobi-Bellman equation (HJB-equation), that is a continuous time version of the well-known dynamic program from the standard RM literature, which is presented in Section 3.4.3. The corresponding ODE has a continuous right hand side that is only piecewise continuously differentiable. In Section 5.1 we describe the continuous time formulation of the inventory control DP.

The IVP formulation is used in Section 5.2 to derive the adjoint equation of the HJB-equation, which allows to efficiently compute sensitivities of the value function w.r.t. a large number of parameters. Moreover, we show in Section 5.4 that solving the problem in its continuous time version using standard numerical methods is more efficient than solving the time-discrete model.

On the other hand, the problem can be written as a deterministic OCP by replacing a random discrete state by its distribution. We show that the ODE governing the forward dynamics of these new states is the adjoint equation of the HJB-equation. This leads to an intuitive interpretation of the adjoint states, which—based on an extension of a well-known network decomposition heuristic—gives rise to a new heuristic approximation of total expected revenue for the network problem presented in Section 6.2.5.

Notation

Throughout this section we will keep the following notation: Using the marginal revenue transformation described in Section 3.5, we can transform any problem with a dependent demand model into one with independent demand between (virtual) products. We therefore only need to be concerned with the solution of the dynamic program for the independent demand case. Product characteristics then do not enter directly into the DP but only through the transformation of fares and demand. Therefore, we do not consider products as vectors of attributes but can rather use a fixed numbering.

Let $\mathcal{P} = \{1, \dots, M\}$ be the set of products and $\mathbb{S} \subseteq \wp(\mathcal{P})$ the set of feasible offer sets. Again by virtue of the demand transformation we can assume w.l.o.g. that $\mathbb{S} = \wp(\mathcal{P})$: We construct one virtual product per feasible offer set and obtain an equivalent independent demand structure without any restrictions on the availability decisions. Each offer set $\mathcal{S} \in \mathbb{S}$ is a subset of set of products. Alternatively, the set of available products can also be described by an incidence vector $s \in \{0,1\}^{\mathcal{P}}$, where product $k \in \mathcal{P}$ is available at time t if and only if $\mathbf{s}_k = 1$. In the following we will use both notations for the same offer set: A lower case s always denotes a binary vector of availability for each class, and an upper case $\mathcal{S}$ denotes the corresponding set of available products.

Denote by $\mathbf{y}_k$ the yield of product k. Let again $[0, T]$ be the booking horizon and let

$$\lambda_k : [0, T] \to \mathbb{R} \tag{5.1}$$

$$t \mapsto \lambda_k(t) \tag{5.2}$$

be the time-dependent demand rate of the Poisson arrival process $\mathbf{N}_k$ for product k.

It is well-known that the sum of two independent Poisson processes is again a Poisson process with arrival rate equal to the sum of the original arrival rates. This allows us to combine products with equal yield into one product and add their rates. Therefore, we can assume w.l.o.g. that products are ordered by yield in strictly descending order:

$$\mathbf{y}_1 > \ldots > \mathbf{y}_M. \tag{5.3}$$

5.1 Stochastic optimal control problem

As stated in Chapter 3, the dynamic single-leg availability control problem can be written as the stochastic optimal control problem

$$\max_{\mathbf{s}} \quad \mathbb{E}\left[\sum_{k\in\mathcal{P}} \int_0^T \mathbf{y}_k \mathbf{s}_k \, \mathrm{d}\mathbf{N}_k\right] \tag{5.4a}$$

$$\text{subject to} \quad \sum_{k\in\mathcal{P}} \int_0^T \mathbf{s}_k \, \mathrm{d}\mathbf{N}_k \le C \qquad \text{a.s.}, \tag{5.4b}$$

where the objective is the expected revenue over the whole booking horizon and the constraint enforces that the number of bookings does not exceed capacity with probability one. In the above, control is exercised via a stochastic process $\mathbf{s}$ with values in the set $\{0,1\}^{\mathcal{P}}$: Product $k \in \mathcal{P}$ is available at time t if and only if $\mathbf{s}_k = 1$.

Introducing the remaining capacity at any time during the booking horizon as a (random) state variable $\mathbf{c} \in \mathbb{N}$, Eq. (5.4) can be formulated as a stochastic optimal control problem with a terminal constraint on the state:

$$\max_{\mathbf{s}} \quad \mathbb{E}\left[\int_0^T \sum_{k\in\mathcal{P}} \mathbf{s}_k \mathbf{y}_k \, \mathrm{d}\mathbf{N}_k\right] \tag{5.5a}$$

$$\text{subject to} \quad \mathrm{d}\mathbf{c} = -\sum_{k\in\mathcal{P}} \mathbf{s}_k \, \mathrm{d}\mathbf{N}_k \tag{5.5b}$$

$$\mathbf{c}(0) = C \tag{5.5c}$$

$$\mathbf{c}(T) \ge 0 \qquad \text{a.s.} \tag{5.5d}$$

Due to the memorylessness of the arrival process, $\mathbf{c}$ is a controlled Markov process on the countably infinite discrete state space $\mathbb{N}$. More precisely, Eq. (5.5) is a continuous-time Markov Decision Process (MDP), and the optimal control $\mathbf{s}^*(t)$ only depends on the value of the state variable $\mathbf{c}(t)$ and the time t, but not on the history of the processes $\mathbf{c}$, $\mathbf{s}$ or $\mathbf{N}$. Therefore, any control process $\mathbf{s}$ that is optimal for Eq. (5.5) can be written as a deterministic control scheme

$$s\colon [0,T] \times \mathbb{N} \to \{0,1\}^{\mathcal{P}} \tag{5.6a}$$

$$(t,c) \mapsto s(t,c), \tag{5.6b}$$

mapping a pair of time t and remaining capacity c to a feasible action.

The terminal state constraint Eq. (5.5d) can be trivially satisfied by not selling any product once capacity is exhausted, i.e. setting

$$s(t,c) := 0 \qquad \forall c \le 0, t \in [0,T] \tag{5.7a}$$

$$\Rightarrow \quad \mathbf{c}(t) \ge 0 \qquad \forall t \in [0,T]. \tag{5.7b}$$

This is w.l.o.g. because we are ignoring cancellations and therefore $\mathbf{c}$ is monotonically non-increasing over time. On the other hand, with the initial state $\mathbf{c}(0) = C$ and the monotonicity of $\mathbf{c}$, we have $\mathbf{c} \le C$.

We can therefore reduce the state space of the Markov process $\mathbf{c}$ to the finite discrete set $\{0,\ldots,C\}$ w.l.o.g. Likewise, the control scheme (Eq. (5.6)) is now a map

$$s\colon [0,T] \times \{0,\ldots,C\} \to \{0,1\}^{\mathcal{P}} \tag{5.8a}$$

$$(t,c) \mapsto s(t,c) \tag{5.8b}$$

with $s(\cdot,0) \equiv 0$.

5.1.1 Solution via Dynamic Programming

The Markov property of the arrival process allows us to solve Eq. (5.5) via dynamic programming. Let

$$V\colon [0,T] \to \mathbb{R}^{\{0,\dots,C\}} \tag{5.9a}$$
$$t \mapsto (V_c(t))_{c\in\{0,\dots,C\}} \tag{5.9b}$$

be the *expected-revenue-to-come function* or simply *value function*, where $V_c(t)$ is the expected revenue to go from time t and remaining capacity c, assuming optimal control. Note that we always have $V_0 \equiv 0$ due to the capacity restriction. In addition, regardless of remaining capacity no revenue can be generated at or after departure and therefore $V(T) = 0$.

Consistent with RM literature, in the following we will represent the marginal value of capacity by the *bid price function*

$$\pi_c(t) = \begin{cases} \infty & \text{if } c = 0 \\ V_c(t) - V_{c-1}(t) & \text{else.} \end{cases} \tag{5.10}$$

It represents the opportunity cost of selling one seat at time t and remaining capacity c. At $c = 0$ we set the bid price to infinity, because there is no seat left to sell without violating the capacity constraints.

Applying Bellman's principle of optimality, we see that the value function is the solution of the HJB-equation

$$\dot{V}_c(t) = -\max_{s\in\{0,1\}^{\mathcal{P}}} \sum_{k\in\mathcal{P}} s_k \lambda_k(t)\,[\mathbf{y}_k - \pi_c(t)] \qquad \forall 1 \le c \le C, t \in [0,T] \tag{5.11a}$$
$$V_c(T) = 0 \qquad \forall 1 \le c \le C. \tag{5.11b}$$

Proposition 5.1.1
The solution of Eq. (5.11) *satisfies the monotonicity properties*

$$\dot{V}_c(t) \le 0 \quad \Leftrightarrow \quad V_c(t+h) \le V_c(t) \tag{5.12a}$$
$$\pi_c(t) = V_c(t) - V_{c-1}(t) \ge 0 \tag{5.12b}$$
$$\dot{\pi}_c(t) \le 0 \quad \Leftrightarrow \quad \pi_c(t+h) \le \pi_c(t) \tag{5.12c}$$
$$\pi_{c+1}(t) \le \pi_c(t) \tag{5.12d}$$
$$\pi_c(t) \le \mathbf{y}_1 \quad \Rightarrow \quad V_c(t) \le c\mathbf{y}_1 \tag{5.12e}$$

for every $t \in [0,T]$*,* $h > 0$ *and every* $c = 1,\dots,C$*.*

Proof A proof is given in the book of Talluri and van Ryzin [117, p. 60]. □

The inequalities Eq. (5.12), in this order, intuitively state the following:

- Expected revenue-to-go for a fixed remaining capacity decreases over time. Equivalently, expected revenue-to-go increases with increasing time to departure.
- The marginal value of one unit of capacity, or bid price, is non-negative.
- The bid price for a fixed remaining capacity decreases over time.
- The bid price is monotonically non-increasing in remaining capacity. In other words, one unit of capacity is more valuable when capacity is sparse.
- The marginal value of one unit of capacity is bounded above by the yield of the highest-valued booking class.

Remark 5.1.2 (Structure of optimal controls) The RHS of Eq. (5.11a) is separable in k and an optimal control strategy is given by

$$s_k^*(t,c) = \begin{cases} 1 & \text{if } \mathbf{y}_k \geq \pi_c(t), \\ 0 & \text{else.} \end{cases} \tag{5.13}$$

Because products are ordered by decreasing yield (Eq. (5.3)), this means that we only have to take into account the nested structure of offer sets

$$\mathcal{S}_0 := \varnothing \subset \mathcal{S}_1 := \{1\} \subset \mathcal{S}_2 := \{1,2\} \subset \ldots \subset \mathcal{S}_M := \mathcal{P}, \tag{5.14}$$

corresponding to the control vectors of the form

$$\hat{s}^k = (\underbrace{1,\ldots,1}_{k},0,\ldots,0)^\top \tag{5.15}$$

for every $k = 0,\ldots,M$. Note that, due to Eq. (5.12e), the highest-value product is always available as long as there is capacity left. We do, however, include the empty offer set in order to simplify notation for the case $c = 0$.

Each such offer set is uniquely determined by its lowest available booking class, i.e. by the product with lowest yield that is contained in the offer set. We can therefore write every optimal control scheme as a map

$$k\colon [0,T] \times \{0,\ldots,C\} \to \{0,\ldots,M\} \tag{5.16a}$$

$$(t,c) \mapsto k(t,c), \tag{5.16b}$$

where $k_c(t)$ is the lowest available booking class at time t and remaining capacity c.

Exploiting this structure, we can rewrite the RHS of the HJB-equation (Eq. (5.11a)) as follows: For every $1 \leq c \leq C$ and $t \in [0,T]$

$$\dot{V}_c(t) = -\max_k \mathbf{R}_k(t) - \pi_c(t)\mathbf{D}_k(t), \tag{5.17}$$

where π is again the bid price, and for every $k \in \{0,\ldots,M\}$

$$\mathbf{R}_k(t) := \sum_{k'=1}^{k} \lambda_{k'}(t)\mathbf{y}_{k'} = \mathbb{E}\left[\sum_{k'=1}^{k} \mathbf{y}_{k'}\,\mathrm{d}\mathbf{N}_{k'}(t)\right] \tag{5.18a}$$

$$\mathbf{D}_k(t) := \sum_{k'=1}^{k} \lambda_{k'}(t) = \mathbb{E}\left[\sum_{k'=1}^{k} \mathrm{d}\mathbf{N}_{k'}(t)\right] \tag{5.18b}$$

denote the *total revenue* and *total demand* rate at time t for the control vector $\hat{s}^k$ respectively.

Then Eq. (5.17) is the linear ODE

$$\dot{V}(t) = -A(t)V(t) - b(t) \tag{5.19a}$$

$$V(T) = 0 \tag{5.19b}$$

with time-dependent coefficients

$$A = \begin{pmatrix} 0 & & & & & \\ \mathbf{D}_{k_1} & -\mathbf{D}_{k_1} & & & & \\ & \mathbf{D}_{k_2} & -\mathbf{D}_{k_2} & & & \\ & & \ddots & \ddots & & \\ & & & \mathbf{D}_{k_C} & -\mathbf{D}_{k_C} \end{pmatrix} \tag{5.20a}$$

$$b = \begin{pmatrix} 0 \\ \mathbf{R}_{k_1} \\ \vdots \\ \mathbf{R}_{k_C} \end{pmatrix}, \tag{5.20b}$$

where k_c maximizes the RHS of Eq. (5.17) for every c, i.e. (see Eq. (5.13)):

$$k_c(t) = \max\{k \mid \mathbf{y}_k \geq \pi(t,c)\} \tag{5.21}$$

and $k_c(t) = 0$ if all $\mathbf{y}_k < \pi(t,c)$.

5.1.2 Formulation as a deterministic optimal control problem

One way to derive the dynamic programming formulation presented above is to transform the stochastic OCP Eq. (5.5) into an equivalent deterministic OCP. As described above, due to the Markov property of the arrival process, the random control process $\mathbf{S}$ can be described by the deterministic map Eq. (5.8) or, equivalently, written as a time-dependent control matrix $S(t) = (s(t)_{c,k})$, where for every remaining capacity $c \in \{0,\dots,C\}$ a product $k \in \mathcal{P}$ is available at time t if and only if the binary variable $s(t)_{c,k} = s_k(t,c) = 1$.

Consider again the Markov process $\mathbf{c}$. Because the state space is finite, the forward dynamics of the system can be described deterministically by introducing the probability distribution of $\mathbf{c}$ as system states. Let

$$\mu\colon [0,T] \to \mathbb{R}^{\{0,\dots,C\}} \tag{5.22a}$$

$$t \mapsto (\mu_c(t))_{c\in\{0,\dots,C\}}, \tag{5.22b}$$

where for every $c \in \{0,\dots,C\}$

$$\mu_c(t) \coloneqq \mathrm{P}\left[\mathbf{c}(t) = c\right] \tag{5.23}$$

be the probability of the discrete random variable $\mathbf{c}(t)$ taking value c at $t \in [0,T]$.

The system can only transition from state $\mathbf{c} = c$ to state $\mathbf{c} = c-1$ and it will do so whenever a booking occurs. Given a control vector $s = s(t,c)$ or—equivalently—an offer set $\mathcal{S}$, the overall booking rate at time t and remaining capacity $\mathbf{c}(t) = c$ is the sum of the arrival rates of all available products, i.e. it is the total demand rate

$$\mathbf{D}(\mathcal{S},t) = \sum_{k\in\mathcal{S}} \lambda_k(t). \tag{5.24}$$

To simplify notation, let

$$\mathbf{D}(S,t) = \begin{pmatrix} \mathbf{D}_1(S,t) \\ \vdots \\ \mathbf{D}_C(S,t) \end{pmatrix} \coloneqq \begin{pmatrix} \mathbf{D}(S_{1,\cdot},t) \\ \vdots \\ \mathbf{D}(S_{C,\cdot},t) \end{pmatrix} \tag{5.25}$$

be the vector of demand rates for every $c = 1,\dots,C$. Note that $\mathbf{D}_0 \equiv 0$ due to the capacity constraint.

The transition rate from state c to state c' is then given by

$$q_{c,c'}(t) = \begin{cases} \mathbf{D}_c(S,t) & ,\text{if } c' = c-1 \\ -\mathbf{D}_c(S,t) & ,\text{if } c' = c \\ 0 & ,\text{else.} \end{cases} \tag{5.26}$$

The time-dependent transition rate matrix of the Markov process $\mathbf{c}$ is therefore

$$Q = (q_{c,c'}(t))_{c,c'\in\{0,\dots,C\}} \tag{5.27}$$

$$= \begin{pmatrix} 0 & & & & & \\ \mathbf{D}_1(S(t),t) & -\mathbf{D}_1(S(t),t) & & & & \\ & \mathbf{D}_2(S(t),t) & -\mathbf{D}_2(S(t),t)) & & & \\ & & \ddots & \ddots & & \\ & & & \mathbf{D}_C(S(t),t) & -\mathbf{D}_C(S(t),t) \end{pmatrix}. \tag{5.28}$$

Together with the fact that we are starting with initial capacity $\mathbf{c}(0) = C$, this means that the state probabilities satisfy the IVP

$$\dot{\mu}^\top(t) = \mu^\top(t) Q(t) \tag{5.29a}$$

$$\mu^\top(0) = (0,0,\dots,0,1) \tag{5.29b}$$

Given controls S, the revenue rate vector $\mathbf{r}(t)$ is given by

$$\mathbf{r}_c(\mathcal{S},t) = \sum_{k\in\mathcal{S}} \lambda_k(t)\mathbf{y}_k \tag{5.30}$$

for every c.

Example 5.1.1 Given a *lowest available class* control scheme like in Eq. (5.21), the booking rates and revenue rates are equal to the total demand and total revenue rates $\mathbf{D}_{k_c(t)}(t)$ and $\mathbf{R}_{k_c(t)}(t)$ of the corresponding feasible action. The transition rate matrix and the revenue rate vector are then given by $Q = A$ and $\mathbf{r} = b$ respectively, with A and b as in Eq. (5.20)

We can formulate Eq. (5.5) as the deterministic OCP

$$\max_S \quad \int_0^T \sum_{c=0}^{C} \mu_c(t) \sum_{k\in\mathcal{P}} S_{c,k}(t)\lambda_k(t)\mathbf{y}_k \,\mathrm{d}t \tag{5.31a}$$

$$\text{subject to} \quad \dot{\mu}_c(t) = -\mu_c(t) \sum_{k\in\mathcal{P}} S_{c,k}(t)\lambda_k(t) + \mu_{c+1}(t) \sum_{k\in\mathcal{P}} S_{c+1,k}(t)\lambda_k(t) \tag{5.31b}$$

$$\mu_c(0) = \begin{cases} 1 & \text{if } c = C, \\ 0 & \text{else} \end{cases} \tag{5.31c}$$

$$S_{0,\cdot} = 0, \tag{5.31d}$$

where $S{:}\,\mathbb{R} \to \{0,1\}^{\{0,\dots,C\}\times\mathcal{P}}$ is a measurable matrix valued function and we use the convention that $\mu_c \equiv 0$ whenever $c \notin \{0,\dots,C\}$. With Q and $\mathbf{r}$ as defined above, this is the same as

$$\max_S \quad \int_0^T \mathbf{r}^\top(S,t)\mu(t)\,\mathrm{d}t \tag{5.32a}$$

$$\text{subject to} \quad \dot{\mu}(t) = Q^\top(S,t)\mu(t) \tag{5.32b}$$

$$\mu(0) = (0,0,\dots,0,1)^\top \tag{5.32c}$$

$$S_{0,\cdot} = 0,. \tag{5.32d}$$

Applying Pontryagin's maximum principle (see Theorem 1.3.3) to the above OCP, we can now easily derive the DP in Eq. (5.11): The only inequality constraints are the simple bounds on the controls, for which we do not need to compute any Lagrange multipliers. Let $V{:}\,\mathbb{R} \to \mathbb{R}^{\{0,\dots,C\}}$ be the dual states associated to the states μ. Since the objective Eq. (5.32a) contains no Mayer term, the terminal conditions for the dual states are simply $V(T) = 0$.

Omitting time as an argument, the Hamiltonian is given by

$$\mathcal{H}(\mu,S,V) = \mathbf{r}(S)^\top\mu + V^\top Q^\top(S)\mu \tag{5.33}$$

$$= \mu^\top\mathbf{r}(S) + \mu^\top Q(S)V. \tag{5.34}$$

The adjoint states satisfy

$$\dot{V} = -\mathcal{H}_\mu^\top(\mu,S^*,V) \tag{5.35}$$

$$= -\mathbf{r}(S^*) - Q(S^*)V \tag{5.36}$$

Note that each row of the transition matrix Q and each component of the revenue rate vector $\mathbf{r}$ only depend on the corresponding row of the control matrix S. Therefore the optimality conditions for the controls—which state that $\mathcal{H}$ is maximized point-wisely—are separable: For every remaining capacity c we have

$$S^*_{c,\cdot}(t) = \operatorname*{arg\,max}_{s_c\in\{0,1\}^{\mathcal{P}}} \mu_c(t)\left[\mathbf{r}_c(s_c,t) + Q_{c,\cdot}(s_c,t)V(t)\right]. \tag{5.37}$$

Together with Eq. (5.36) and the fact that $\mu_c \geq 0$ we obtain the dynamic programming recursion Eq. (5.11). This proves the first part of our claim that the dynamic programming formulation Eq. (5.11) and the OCP formulation Eq. (5.32) of the single-leg capacity control problem are dual to each other: The value function V in the DP Eq. (5.11) is a solution to the adjoint equations Eq. (5.35) for the OCP Eq. (5.32).

5.2 Sensitivity analysis of the dynamic program

In the previous section we introduced the states μ, which describe the distribution of remaining capacity over the course of the booking horizon, and the corresponding ODE Eq. (5.32). In this section we will show that this forward differential equation is the adjoint equation of the HJB-equation for a canonical functional, namely the expected revenue to come from time $t = 0$ and remaining capacity $c = C$. This means that a solution of Eq. (5.32) can be used to efficiently compute the derivative of $V_C(0;\mathfrak{p})$ with respect to parameters $\mathfrak{p}$.

If product attributes are parametrized by optimization variables u as described in Section 4.2.1, then the value function V is a function of u via the demand rates λ as well as the yields $\mathbf{y}$, both of which depend on u. In this section we will derive the forwards and backwards sensitivity systems for the HJB-equation 5.17. In order to account for the different ways to parametrize the DP, we will add an additional layer of abstraction and write the value function as a function $V(t, \mathbf{p}(u))$ with the parameters $\mathbf{p}$, which in turn depend on the optimization variables u.

Note that in the context of dynamic programming one usually refers to the solution of the HJB-equation, which is a terminal value problem, as the backward sweep. In the context of adjoint sensitivity, the solution of the original IVP is called the forward sweep, while the solution of the adjoint equation is called the backward sweep. In order to avoid confusion, in the following we will always use terms *forward* and *backward* consistently with the parametrization of time, i.e. the HJB-equation is solved *backwards*, and the adjoint equation is solved *forwards*.

Using Eq. (5.19), in the notation of Section 1.2.1 we have

$$\mathbf{y} = V \tag{5.38a}$$

$$\mathbf{f}(t, \mathbf{y}, \mathbf{p}) = -A(t, \mathbf{p})\mathbf{y} - b(t, \mathbf{p}) \tag{5.38b}$$

$$\mathbf{y}(T) = 0. \tag{5.38c}$$

The objective for the pricing problem (Eq. (4.17a)) is to maximize the expected revenue to go across the whole booking horizon when starting with initial capacity C. We therefore have the end-term objective functional

$$g(\mathbf{p}) = V_C(0, \mathbf{p}). \tag{5.39}$$

Written in the form of Eq. (1.34) it is given by

$$\phi\left[0, \mathbf{y}, \mathbf{p}\right] = \mathbf{y}_C. \tag{5.40}$$

Variational equation
The backward sensitivity system is given by

$$\dot{V}^{\mathbf{p}}(t, \mathbf{p}) = -A(t, \mathbf{p})V^{\mathbf{p}}(t, \mathbf{p}) - \frac{\mathrm{d}A}{\mathrm{d}\mathbf{p}}(t, \mathbf{p})V(t, \mathbf{p}) - \frac{\mathrm{d}b}{\mathrm{d}\mathbf{p}}(t, \mathbf{p}) \tag{5.41a}$$

$$\dot{V}^{\mathbf{p}}(T, \mathbf{p}) = 0. \tag{5.41b}$$

where $V^{\mathbf{p}} := \frac{\mathrm{d}V}{\mathrm{d}\mathbf{p}}$. Here, we use an upper index to denote the derivative of the value function w.r.t. parameters, because the lower index is used for capacity. Therefore $\frac{\mathrm{d}g}{\mathrm{d}\mathbf{p}}(\mathbf{p}) = V_C^{\mathbf{p}}(0, \mathbf{p})$.

Adjoint sensitivity
In a typical instance of the single-leg DP the number of states, which equals the initial capacity C, is at least 100. The parameter vector $\mathbf{p}$, on the other hand, describes the—often very complex—price structures offered in the market and can therefore be very large as well. In addition, we are only interested in a directional derivative of the value function, namely the derivative of the last state at time $t = 0$. Together, these facts suggest the use of adjoint (which in this case is forward) sensitivity instead of the variational equation Eq. (5.41).

Proposition 5.2.1
The adjoint states $\boldsymbol{\mu}$ of the HJB-equation, as defined in Section 1.2.1, are equal to the probabilities of the system being in a certain state (i.e. having a certain number of remaining seats) at any given time. In other words:

$$\boldsymbol{\mu}_c(t) = P\left[\mathbf{c}(t) = c\right]. \tag{5.42}$$

Proof Using Eq. (1.45), we see that the adjoint states $\boldsymbol{\mu}$ are the solution to the IVP

$$\dot{\boldsymbol{\mu}} = -\mathbf{f}_{\mathbf{y}}^{\top}\boldsymbol{\mu} \quad \overset{(5.38)}{\Leftrightarrow} \quad \dot{\boldsymbol{\mu}}(t,\mathbf{p}) = A^{\top}\boldsymbol{\mu} \tag{5.43a}$$

$$\boldsymbol{\mu}(0) = L_{\mathbf{y}}^{\top}(0) \quad \overset{(5.40)}{\Leftrightarrow} \quad \boldsymbol{\mu}(0,\mathbf{p}) = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}, \tag{5.43b}$$

which is the same as Eq. (5.29), which governs the evolution of the state probabilities. □

Then, with Eq. (1.44),

$$\frac{\mathrm{d}g}{\mathrm{d}\mathbf{p}}(\mathbf{p}) = \underbrace{L_{\mathbf{p}}(0)}_{=0} + \boldsymbol{\mu}^{\top}(T)\underbrace{\mathbf{y}_{\mathbf{p}}(T)}_{=0} + \int_{T}^{0} \boldsymbol{\mu}^{\top}\mathbf{f}_{\mathbf{p}}\,\mathrm{d}t \tag{5.44a}$$

$$= -\int_{0}^{T} \boldsymbol{\mu}^{\top}\mathbf{f}_{\mathbf{p}}\,\mathrm{d}t \tag{5.44b}$$

$$\overset{(5.19)}{=} \int_{0}^{T} \boldsymbol{\mu}^{\top}\left[\frac{\mathrm{d}A}{\mathrm{d}\mathbf{p}}V + \frac{\mathrm{d}b}{\mathrm{d}\mathbf{p}}\right]\mathrm{d}t. \tag{5.44c}$$

Here, the negative sign in the second equation is due to the reversed integration limits, which are a result of the fact that the HJB-equation is a terminal value problem.

By definition, the derivatives of the coefficients A and b can be computed as

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{D}_k = \sum_{k'=1}^{k} \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\lambda_{k'} \tag{5.45a}$$

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{R}_k = \sum_{k'=1}^{k} \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\lambda_{k'}\mathbf{y}_{k'} + \lambda_{k'}\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{y}_{k'}. \tag{5.45b}$$

The exact form of these derivatives depend on the customer model and on the choice of $\mathbf{p}$.

Example 5.2.1 Assume that the demand rates λ_k are piecewise constant on $[0,T]$. More precisely, let

$$\begin{aligned} I_i &= [t_{i-1}, t_i] \qquad && \forall i = 1,\dots,n \\ t_0 &= 0 \\ t_n &= T \end{aligned}$$

be a fixed discretization of the booking horizon, with

$$\lambda_k(t) = \lambda_{k,i} \qquad \forall t \in I_i, k \in \mathcal{P}. \tag{5.46}$$

Then, by definition (Eq. (3.58)), total revenue $\mathbf{R}$ and total demand $\mathbf{D}$ are piecewise constant as well:

$$\mathbf{D}_k(t) = \mathbf{D}_{k,i} := \sum_{k'=1}^{k} \lambda_{k',i} \qquad \forall t \in I_i, k \in \mathcal{P} \tag{5.47a}$$

$$\mathbf{R}_k(t) = \mathbf{R}_{k,i} := \sum_{k'=1}^{k} \lambda_{k',i}\mathbf{y}_{k'} \qquad \forall t \in I_i, k \in \mathcal{P} \tag{5.47b}$$

An instance of the DP is then uniquely determined by the $2nM$ parameters

$$\mathbf{p} = \left(\begin{array}{ccc|ccc} \mathbf{R}_{1,1} & \cdots & \mathbf{R}_{M,1} & \mathbf{D}_{1,1} & \cdots & \mathbf{D}_{M,1} \\ \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{R}_{1,n} & \cdots & \mathbf{R}_{M,n} & \mathbf{D}_{1,n} & \cdots & \mathbf{D}_{M,n} \end{array}\right). \tag{5.48}$$

With this choice of $\mathbf{p}$, all entries of the coefficients A and b are components of $\mathbf{p}$. As a result, the tensors $\frac{\mathrm{d}A}{\mathrm{d}\mathbf{p}}$ and $\frac{\mathrm{d}b}{\mathrm{d}\mathbf{p}}$ are easy to compute and extremely sparse. In addition, parameters are local in the sense that only one row vector

$$(\mathbf{R}_{1,i}, \dots, \mathbf{R}_{M,i}, \mathbf{D}_{1,i}, \dots, \mathbf{D}_{M,i})^\top$$

actually influences the solution at time $t \in I_i$. Therefore, locally the number of parameters is $2M$.

The derivative of the value function, $\frac{\mathrm{d}V}{\mathrm{d}\mathbf{p}}(0)$, can be computed by solving

- the $(2M+1)C$-dimensional variational equation of the HJB-equation,
- or, with the adjoint approach, ODEs of dimensions C and $C + 2M$ for the forward and reverse sweeps respectively.

The derivative of $V_C(0)$ w.r.t. the controls u is then

$$\frac{\mathrm{d}V_C}{\mathrm{d}u}(0) = \frac{\mathrm{d}V_C}{\mathrm{d}\mathbf{p}}(0)\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}u}, \tag{5.49}$$

where, by definition of $\mathbf{R}$ and $\mathbf{D}$ (Eq. (5.47)),

$$\frac{\mathrm{d}\mathbf{D}_{k,i}}{\mathrm{d}u} = \sum_{k'=1}^{k} \frac{\mathrm{d}}{\mathrm{d}u}\lambda_{k',i} \tag{5.50a}$$

$$\frac{\mathrm{d}\mathbf{R}_{k,i}}{\mathrm{d}u} = \sum_{k'=1}^{k} \left[\frac{\mathrm{d}}{\mathrm{d}u}\lambda_{k',i}\mathbf{y}_{k'} + \lambda_{k',i}\frac{\mathrm{d}}{\mathrm{d}u}\mathbf{y}_{k'}\right], \tag{5.50b}$$

for every $k \in \mathcal{P}$. In Eq. (5.50), the derivative $\frac{\mathrm{d}\mathbf{y}_k}{\mathrm{d}u}$ is determined by how products are parametrized (see Section 4.2.1), while $\frac{\mathrm{d}}{\mathrm{d}u}\lambda_{k,i}$, measuring sensitivity of demand w.r.t. controls, depends on the customer model.

Example 5.2.2 As a special case of Example 5.2.1, consider the demand model described in Section 4.2.3, where choice behavior is constant over time for a fixed customer type, and changing aggregate choice behavior arises from changing, but piecewise constant, mix of customer types. Let again

$$\begin{aligned} I_i &= [t_{i-1}, t_i] \qquad\qquad \forall i = 1, \dots, n \\ t_0 &= 0 \\ t_n &= T \end{aligned}$$

be a fixed discretization of the booking horizon and assume that the arrival rates $\lambda^1, \dots, \lambda^L$ associated with the customer types $1, \dots, L$ are constant on each I_i:

$$\lambda^l(t) = \lambda^l_i \qquad\qquad \forall t \in I_i. \tag{5.51}$$

As a result, with Eqs. (4.32) and (3.58), total demand $\mathbf{D}_k$ and total revenue $\mathbf{R}_k$ for each virtual product k are piecewise constant as well and given by

$$\mathbf{D}_k(t) = \mathbf{D}(\mathcal{S}_k, t) \coloneqq \mathbf{D}_{k,i} = \sum_{l=1}^{L} \lambda^l_i \sum_{k'} d^l_{k'}(\mathcal{S}_k) \qquad \forall t \in I_i \tag{5.52a}$$

$$\mathbf{R}_k(t) = \mathbf{R}(\mathcal{S}_k, t) \coloneqq \mathbf{R}_{k,i} = \sum_{l=1}^{L} \lambda^l_i \sum_{k'} d^l_{k'}(\mathcal{S}_k)\mathbf{y}_{k'} \qquad \forall t \in I_i \tag{5.52b}$$

for every $k \in \mathcal{P}$, where $\mathcal{S}_k$ is the offer set corresponding to the virtual product k. Taking the derivative w.r.t. u yields

$$\frac{\mathrm{d}\mathbf{D}_{k,i}}{\mathrm{d}u} = \sum_{l=1}^{L} \lambda_i^l \sum_{k'=1}^{k} \frac{\mathrm{d}}{\mathrm{d}u} d_{k'}^l(\mathcal{S}_k) \tag{5.53a}$$

$$\frac{\mathrm{d}\mathbf{R}_{k,i}}{\mathrm{d}u} = \sum_{l=1}^{L} \lambda_i^l \sum_{k'=1}^{k} \left[\frac{\mathrm{d}}{\mathrm{d}u} d_{k'}^l(\mathcal{S}_k)\mathbf{y}_{k'} + d_{k'}^l(\mathcal{S}_k) \frac{\mathrm{d}}{\mathrm{d}u}\mathbf{y}_{k'} \right]. \tag{5.53b}$$

Here, $\frac{\mathrm{d}}{\mathrm{d}u} d_{k'}^l(\mathcal{S}_k)$, measuring sensitivity of demand w.r.t. controls, can be computed using the methods presented in Section 7.1. Note that this derivative is independent of the time discretization and therefore only needs to be computed once.

One alternative parametrization of the DP is the canonical choice $\mathbf{p} := u$. This reduces the number of parameters, if the number of (virtual, independent) products in the DP is much larger than the original number of (dependent) products before demand transformation. However, this reduction comes at the cost of a loss of sparsity in the tensors $\frac{\mathrm{d}A}{\mathrm{d}\mathbf{p}}$ and $\frac{\mathrm{d}b}{\mathrm{d}\mathbf{p}}$, increasing the cost of evaluating the RHS of the backward sensitivity system (Eq. (5.41a)) as well as the integrand in Eq. (5.44c) in the adjoint approach by a factor of $O(|u|)$. As shown in Section 5.4, in an efficient implementation this is the governing factor for overall performance.

Example 5.2.3 Assume that the parameter $\mathbf{p}$ does not have any impact on demand, but only influences the yield vector $\mathbf{y}$. In other words, $\mathbf{p}$ does not control any visible product characteristics (including price), but only influences the value that selling one unit of a certain product has for the airline. This case occurs in several different scenarios, for example:

Changing marginal cost If marginal cost is included in the pricing and inventory control problems, then the net yield associated with product k is given by

$$\mathbf{y}_k = f_k - c_k, \tag{5.54}$$

where f_k and c_k are the price and the marginal costs of the product respectively.

Varying exchange rates Typically, an airline sells tickets all over the world and, for marketing reasons, prices are fixed in the local currencies. With a given local price f_k, the airline's utility of selling one unit of product k is

$$\mathbf{y}_k = \alpha f_k, \tag{5.55}$$

where α is the exchange rate between the customer's and the airline's local currencies respectively.

Network decomposition heuristics As described in Section 3.4.4, most advanced network decomposition heuristics involve the solution of a number of single-leg DPs, where the yields associated with each product k are of the form

$$\mathbf{y}_k = f_k - \hat{\pi}_k, \tag{5.56}$$

where $\hat{\pi}_k$ is a displacement adjustment for product k, containing information about the network effects of selling one unit of product k.

In all of these cases, a change of the respective parameter has no influence on customer choice behavior, because the product's end price for the customer remains constant.

With Eq. (5.45), we have

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{D}_k = 0 \tag{5.57a}$$

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{R}_k = \sum_{k'=1}^{k} \lambda_{k'} \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{y}_{k'} \tag{5.57b}$$

and therefore

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}} A = 0 \tag{5.58a}$$

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}} b = \begin{pmatrix} \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}} \mathbf{R}_{k_1} \\ \vdots \\ \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}} \mathbf{R}_{k_C} \end{pmatrix}, \tag{5.58b}$$

where k_c is the lowest open booking class if remaining capacity is c and the last equality follows from Eq. (5.19).

Because sensitivity of the value function w.r.t. the varying parameter is particularly useful for network decomposition heuristics, consider the case where $\mathbf{p}$ is an additive constant in the yield of product $\hat{k}$:

$$\mathbf{y}_{\hat{k}} = f_{\hat{k}} + \mathbf{p}. \tag{5.59}$$

Moreover, assume that the yields of all other products are independent of $\mathbf{p}$ and therefore

$$\frac{\mathrm{d}\mathbf{y}_k}{\mathrm{d}\mathbf{p}} = \delta_{k,\hat{k}}. \tag{5.60}$$

By definition of the total revenue rate,

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}} \mathbf{R}_k = \sum_{k'=1}^{k} \lambda_{k'} \underbrace{\frac{\mathrm{d}\mathbf{y}_{k'}}{\mathrm{d}\mathbf{p}}}_{\delta_{k,\hat{k}}} + \underbrace{\frac{\mathrm{d}\lambda_{k'}}{\mathrm{d}\mathbf{p}}}_{=0} \mathbf{y}_{k'} \tag{5.61}$$

$$= \begin{cases} \lambda_{k'}, & \text{if } \hat{k} \leq k \\ 0, & \text{else.} \end{cases} \tag{5.62}$$

The derivative of the value function is therefore given by (see Eq. (5.44))

$$\frac{\mathrm{d}g}{\mathrm{d}\mathbf{p}}(\mathbf{p}) = \int_0^T \boldsymbol{\mu}^\intercal \frac{\mathrm{d}b}{\mathrm{d}\mathbf{p}} \tag{5.63a}$$

$$= \int_0^T \sum_{c=1}^{C} \boldsymbol{\mu}_c(t) \lambda_{\hat{k}}(t) \mathbf{1}_{\hat{k} \in \mathcal{S}(t,c)} \, \mathrm{d}t, \tag{5.63b}$$

where $\mathcal{S}(t,c)$ is the offer set at time t and remaining capacity c, and $\mathbf{1}_.$ is an indicator function.

With the random variable $\mathbf{c}$ for remaining capacity, the offer set is a random process $\mathbf{S}(t) = \mathcal{S}(t, \mathbf{c})$. Using the fact that $\boldsymbol{\mu}_c(t) = \mathrm{P}[\mathbf{c}(t) = c]$, we obtain

$$\frac{\mathrm{d}g}{\mathrm{d}\mathbf{p}}(\mathbf{p}) = \int_0^T \lambda_{\hat{k}}(t) \sum_{c=1}^{C} \mathrm{P}[\mathbf{c}(t) = c] \, \mathbf{1}_{\hat{k} \in \mathcal{S}(t,c)} \, \mathrm{d}t \tag{5.64a}$$

$$= \int_0^T \lambda_{\hat{k}}(t) \mathrm{P}\left[\hat{k} \in \mathbf{S}(t)\right] \mathrm{d}t \tag{5.64b}$$

$$= \mathbb{E}\left[\int_0^T \mathbf{s}_{\hat{k}} \, \mathrm{d}\mathbf{N}_{\hat{k}}\right], \tag{5.64c}$$

which is the expected number of bookings for product $\hat{k}$ given the control scheme $\mathcal{S}$.

5.3 Sensitivity analysis of the optimal control problem

The sensitivity result Eq. (5.44) can also be obtained by performing a sensitivity analysis on the OCP Eq. (5.32) using the methods described in Section 1.3.2: Let $g(\mathbf{p})$ be the optimal objective function value of a parametric version of Eq. (5.32), where the revenue rate $\mathbf{r}$ and the total demand rates $\mathbf{D}$—and therefore the transition rate matrix Q—depend on a parameter $\mathbf{p}$. Applying Theorem 1.3.10 and using Eq. (5.33), we have

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}} g(\mathbf{p}) = \int_0^T \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \,\mathrm{d}t \tag{5.65a}$$

$$= \int_0^T \mu^\top \frac{\partial \mathbf{r}}{\partial \mathbf{p}}(S^*) + \mu^\top \frac{\partial Q}{\partial \mathbf{p}}(S^*) V \,\mathrm{d}t \tag{5.65b}$$

$$= \int_0^T \sum_{c=0}^{C} \mu_c \left[\frac{\partial \mathbf{r}_c}{\partial \mathbf{p}}(S^*) + \sum_{c'=0}^{C} \frac{\partial Q_{c,c'}}{\partial \mathbf{p}}(S^*) V_{c'} \right] \mathrm{d}t \tag{5.65c}$$

$$= \sum_{c=0}^{C} \int_0^T \mu_c \left[\frac{\partial \mathbf{r}_c}{\partial \mathbf{p}}(S^*) + \frac{\partial \mathbf{D}_c}{\partial \mathbf{p}}(S^*)(-V_c + V_{c-1}) \right] \mathrm{d}t \tag{5.65d}$$

$$= \mathbb{E}_{\mathbf{c}} \left[\int_0^T \frac{\partial \mathbf{r}}{\partial \mathbf{p}}(\mathbf{s}^*) - \frac{\partial \mathbf{D}}{\partial \mathbf{p}}(\mathbf{s}^*) \boldsymbol{\pi} \,\mathrm{d}t \right]. \tag{5.65e}$$

Here, $\mathbf{s}^*$ is the optimal random control process for the stochastic OCP and the random process $\boldsymbol{\pi}$ is the bid price at every point in time. Both depend deterministically on time and remaining inventory, satisfying $\mathbf{s}^*(t \mid \mathbf{c} = c) = S^*_{c,\cdot}(t)$ and $\boldsymbol{\pi}(t \mid \mathbf{c} = c) = \pi_c(t)$. In the last identity we use the interpretation that $\mu_c(t) = \mathrm{P}[\mathbf{c}(t) = c]$ and the fact that

$$\mathbf{r}_c(S^*) = \mathbf{r}(\mathbf{s}^* \mid \mathbf{c} = c) = \mathbf{r}(\mathbf{s}^*) \mid \mathbf{c} = c$$
$$\mathbf{D}_c(S^*) = \mathbf{D}(\mathbf{s}^* \mid \mathbf{c} = c) = \mathbf{D}(\mathbf{s}^*) \mid \mathbf{c} = c$$

the revenue rate and total demand rate given remaining capacity $\mathbf{c} = c$.

5.4 Numerical solution of the DP using higher order methods

In order to solve the pricing problem using gradient based methods, we frequently need to evaluate the objective function—expected revenue under optimal availability control—and its gradient for different parameter vectors. The former is most easily done by solving the DP. The latter is, particularly for a large number of parameters, best accomplished by first computing the adjoint states of the value function, i.e. the primal states of the OCP, and then evaluating the integral Eq. (5.44c). In this section we show how both can be performed very efficiently using simple higher order explicit integration schemes, while maintaining high accuracy in the results.

The most widely used formulation of the single-leg DP is the discrete time version described in Section 3.4.3. As a result, the problem is often solved by directly computing the recursion Eq. (3.44), which is equivalent to the numerical solution of Eq. (5.11) using an explicit Euler method while ignoring the non-differentiability in the RHS.

The problem can be solved much more efficiently using standard numerical methods for the solution of ordinary differential equations (see Section 1.2). However, in general fast convergence for higher order methods can only be guaranteed if $\mathbf{f}$ is sufficiently smooth.

Due to the maximum in the RHS of Eq. (5.11), the ODE has to be treated as an IVP with implicit switches. Compared to a general problem of this kind, the RM dynamic program is in a sense much smoother: First of all, the solution is continuous, in other words there are no jumps in the differential states at switching times. In addition, the RHS is continuous, albeit not continuously differentiable. Therefore we will show in the following that, although technically the continuous time DP should be solved using a suitable solver that can deal with implicit switches, quadratic convergence can still be achieved when a standard method is applied naively without stopping the integrator at switching times.

Let $I = [t_0, T] \subset \mathbb{R}$. For every $k = 1, \ldots, m$ let

$$\mathbf{f}_k \colon I \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_y} \tag{5.66a}$$

$$(t, \mathbf{y}) \mapsto \mathbf{f}_k(t, \mathbf{y}) \tag{5.66b}$$

be continuous in t and Lipschitz continuous in $\mathbf{y}$. Then

$$\mathbf{f}\colon I \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_y} \tag{5.67a}$$
$$(t, \mathbf{y}) \mapsto \mathbf{f}(t, \mathbf{y}) = \max\left\{\mathbf{f}_k(t, \mathbf{y}(t)) \mid k = 1, \dots m\right\}. \tag{5.67b}$$

is Lipschitz continuous in $\mathbf{y}$ as well. Denote by $\mathbf{y}(t; t_0, \mathbf{y}_0)$ the exact solution of the IVP

$$\begin{aligned}\dot{\mathbf{y}}(t) &= \mathbf{f}(t, \mathbf{y}(t))\\ \mathbf{y}(t_0) &= \mathbf{y}_0.\end{aligned}$$

Because all $\mathbf{f}_k$ are continuous in t and $\mathbf{y}$, for every $t' \in I$ there are $\epsilon > 0$ and k such that $\mathbf{f}(t, \mathbf{y}(t)) = \mathbf{f}_k(t, \mathbf{y}(t))$ on all of $[t', t' + \epsilon]$. If the problem is non-degenerate, and in particular if demand rates are assumed to be piecewise constant over time, the number of switches is finite. We can therefore partition I into n intervals

$$\begin{aligned}I_0 &= [t_0, t_1] && \\ I_i &= [t_i, t_{i+1}] && \forall i = 1, \dots, n-2\\ I_{n-1} &= [t_{n-1}, t_n = T] &&\end{aligned}$$

where $t_1, \dots, t_{n-1}$ are the switching times. Choose $k(i)$ such that $\mathbf{f}_k$ maximizes the RHS of Eq. (5.67a) in interval i:

$$\mathbf{f}_{k(i)}(t, \mathbf{y}(t)) = \max\left\{\mathbf{f}_k(t, \mathbf{y}(t)) \mid k = 1, \dots m\right\} \qquad \forall t \in I_i$$

For every $k = 1, \dots, m$ denote by $\mathbf{y}^k(t; t_0, \mathbf{y}_0)$ the solution of the IVP

$$\begin{aligned}\dot{\mathbf{y}}^k(t) &= \mathbf{f}_k(t, \mathbf{y}(t))\\ \mathbf{y}^k(t_0) &= \mathbf{y}_0.\end{aligned}$$

Then $\mathbf{y}$ satisfies

$$\begin{aligned}\mathbf{y}(t) &= \mathbf{y}^{k(i)}(t; t_i, \mathbf{y}_0^i) && \forall t \in I_i\\ \mathbf{y}_0^0 &= \mathbf{y}_0 && \\ \mathbf{y}_0^i &= \mathbf{y}^{k(i-1)}(t_i; t_{i-1}, \mathbf{y}_0^{i-1}) && \forall i = 1, \dots, n-1\end{aligned}$$

Assume we attempt to solve Eq. (5.67a) numerically using $\tilde{n}$ integration steps with fixed size $h = \frac{T - t_0}{\tilde{n}}$, without stopping at the switching times. The corresponding partitioning of the booking horizon is

$$\begin{aligned}\tilde{I}_0 &= [t_0, \tilde{t}_1] && \\ I_i &= [\tilde{t}_i, \tilde{t}_{i+1}] && \forall i = 1, \dots, \tilde{n}-2\\ I_{\tilde{n}-1} &= [\tilde{t}_{n-1}, \tilde{t}_n = T], &&\end{aligned}$$

where $\tilde{t}_i = t_0 + ih$. Instead of taking the maximum in $\mathbf{f}$ point-wise whenever the integrator evaluates the RHS, in each integration step we use the $\mathbf{f}_k$ that is maximal at the beginning of the current step. Ignoring the discretization error, this means that we actually solve the problem

$$\begin{aligned}u(t) &= \mathbf{y}^{\tilde{k}(j)}(t; \tilde{t}_j, u_0^j) && \forall t \in \tilde{I}_j\\ u_0^0 &= \mathbf{y}_0 && \\ u_0^j &= \mathbf{y}^{\tilde{k}(j-1)}(\tilde{t}_j; \tilde{t}_{j-1}, u_0^{j-1}) && \forall j = 1, \dots, \tilde{n}-1,\end{aligned}$$

where $\tilde{k}(j)$ is chosen so that $\mathbf{f}_{\tilde{k}(j)}$ maximizes $\mathbf{f}$ when evaluated at the initial conditions for interval j:

$$\mathbf{f}_{\tilde{k}(j)}(\tilde{t}_j, u_0^j) = \max\left\{\mathbf{f}_k(\tilde{t}_j, u_0^j) \mid k = 1, \dots m\right\}$$

Let $t_i \in \tilde{I}_j$ be the first switching time that is in the interior of one of the discretization intervals. Then $u = \mathbf{y}$ on all prior integration steps and at the switch t_i

$$\mathbf{y}(t_i) = u(t_i) \tag{5.68a}$$
$$\dot{\mathbf{y}}(t_i) = \dot{u}(t_i) \tag{5.68b}$$

The local error

$$\delta(t) = u(t) - \mathbf{y}(t) \tag{5.69}$$

incurred in step j by ignoring the switch at t_i satisfies

$$\dot{\delta}(t) = \tilde{\mathbf{f}}(t) := \mathbf{f}_{\tilde{k}(j)}(t, u(t)) - \mathbf{f}_{k(i)}(t, \mathbf{y}(t)) \tag{5.70}$$
$$\delta(t_i) = 0 \tag{5.71}$$

on the interval $I = [t_i, \tilde{t}_j]$. Since all $\mathbf{f}_k$ are continuously differentiable, $\tilde{\mathbf{f}}$ is continuously differentiable on I. As a result, δ is twice continuously differentiable on I and has a Taylor expansion around t_i

$$\delta(t) = \underbrace{\delta(t_i)}_{0} + \underbrace{\dot{\delta}(t_i)}_{0}(t - t_i) + o\left((t - t_i)^2\right), \tag{5.72}$$

where the constant and first order terms are zero due to Eq. (5.68). In particular,

$$\delta(\tilde{t}_j) = o(h^2).$$

If multiple switches at switching times $t_i, t_{i+1}, \ldots$ are missed within one integrator step, the same analysis can be used for each sub-interval $[t_i, t_{i+1}]$.

Thus, like the local discretization error of a first order method such as the explicit Euler method, the local error arising from missed switches is quadratic in the step size h. However, contrary to the discretization error of the integrator, it only occurs in those time steps that run over a switch. If the number of switches n is finite, it is a problem-dependent but constant upper bound on the number of inaccurate steps. Thus, if error propagation is moderate, asymptotically the overall error is still quadratic in h as $h \to 0$. Whether this is the case of course depends on the condition of the problem as well as the stability of the integrator.

In case of the availability control DP, significant improvements in computational efficiency can be gained by using a second order method instead of explicit Euler. Depending on the constants in the respective errors, methods of order greater than two can lead to even better results, although convergence will still only be quadratic. Generally this will be the case if the overall error is mainly caused by the discretization error instead of missed switches.

Example 5.4.1 A realistically sized problem with an initial capacity of $C = 200$ seats and $M = 20$ products was considered to demonstrate these results. Prices f_k for each product k were generated randomly following an exponential distribution with rate 1. Based on these prices, expected demand λ_k for each product k was generated randomly following an exponential distribution with rate $\frac{1}{f_k}$ and then normalized such that

$$\sum_{k=1}^{M} \lambda_k = C. \tag{5.73}$$

Prices and demand are shown in Table 5.1, with products ordered by decreasing price.

Six instances of the single-leg DP were generated by scaling the demand vector with a factor $\alpha \in \{0.7, 0.85, 1.0, 1.5, 2.0, 4.0\}$, that determines the ratio of overall overall demand to capacity.

Each instance was then solved with the explicit Euler method, the second order Heun method, and the classic fourth order Runge-Kutta method, using various step sizes. Since the DP does not have an easily accessible analytical solution, for each problem instance a numerical estimate of the true value was computed up to a relative error of $\sqrt{\epsilon} \simeq 10^{-8}$ (where ϵ is machine accuracy using double precision arithmetic), which is the round-off error to be expected using numerical integration.

Demand	0.2696	1.5110	0.0755	0.8752	3.5520
Price	4.3039	2.6803	1.8400	1.0492	1.0120
Demand	37.1994	3.8842	1.2034	0.4483	15.6664
Price	0.8681	0.8371	0.8026	0.7349	0.6271
Demand	4.7281	3.9982	4.0505	18.9604	25.2792
Price	0.5094	0.4488	0.4290	0.4048	0.3396
Demand	21.5868	3.2895	30.5751	3.8755	18.9719
Price	0.2881	0.1716	0.1342	0.0596	0.0507

Table 5.1: Demand and prices for example 5.4.1

Demand Factor	Euler	Heun	Rk4
0.7	0.27120	0.03354	0.00684
0.8	0.99082	1.21632	0.31194
1.0	1.00132	1.97806	1.37341
1.5	1.00118	2.00247	1.73240
2.0	1.00156	1.90217	1.41942
4.0	1.00250	1.96829	1.71862

Table 5.2: Slope coefficients of log-linear model fit

The results were produced on a workstation with an AMD Phenom II X6 1055T processor at 1.5 Ghz and 8 GB RAM using Ubuntu 12.04 with the software described in Appendix A.1. Figures 5.1 and 5.3 show plots of the relative error of the value function and bid price respectively compared to these estimates, plotted against the number of integration steps taken. For very low demand the different methods perform similarly, because the global error is mostly determined by machine accuracy. In particular, the actual bid price is close to zero in this case, causing the cancellation error to outweigh the discretization error.

With higher demand, higher order methods become more accurate using the same number of integration steps. More importantly, as shown in Figs. 5.2 and 5.4, they become more efficient per unit of CPU time as well. Table 5.2 shows the slope coefficients of log-linear models, that are plotted as straight lines in Fig. 5.1.

As expected, the convergence rate of the fourth order method is still only quadratic. However, Fig. 5.1 shows that, due to a constant difference in accuracy, the CPU time required to achieve a given relative error using the RK4 method is lower by a factor of about 10 compared to the time required to achieve the same accuracy with a second order method. The improvement over the explicit Euler method, which is widely used in practice, is several orders of magnitude.

Remark 5.4.1 Clearly, the application of a higher order method with the capability to deal with implicit switches would lead to better asymptotic behavior. In addition, standard error estimation techniques will fail if discontinuities are simply ignored, preventing the use of integration schemes with error control and adaptive step sizes.

However, in examples of realistic size the number of switches is only bounded by the product of

- the number of time intervals of a piecewise constant demand model,
- the number of products,
- the initial capacity.

The number of virtual products arising from demand transformation (Section 3.5) of a discrete choice model can easily exceed 100, while initial capacity is commonly between 100 and 400. Even assuming constant demand, this leads to approximately 10^4 switches, while for standard demand models the number will often rise to more than 10^5. In addition to the computational

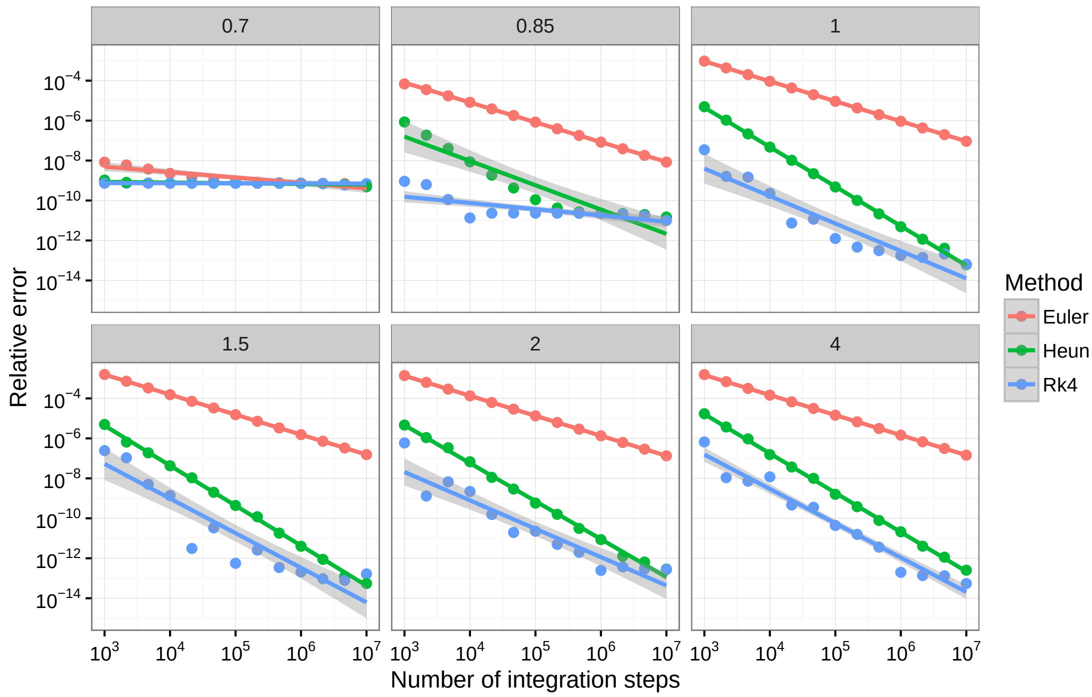


Figure 5.1: Value function: error vs. number of Steps

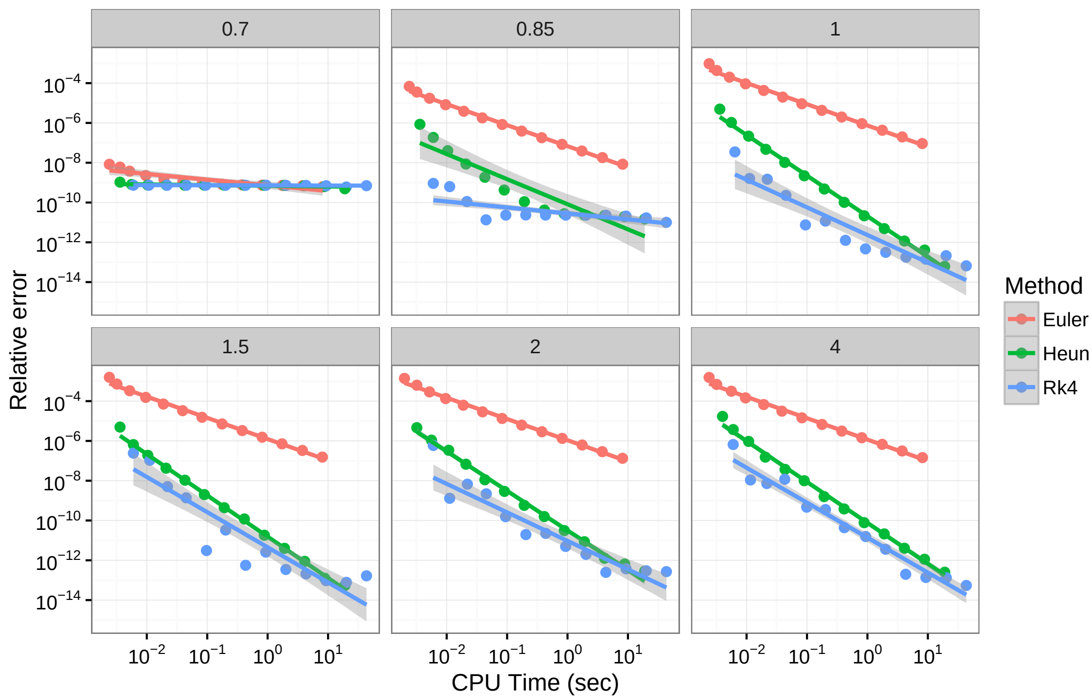


Figure 5.2: Value function: error vs. CPU time

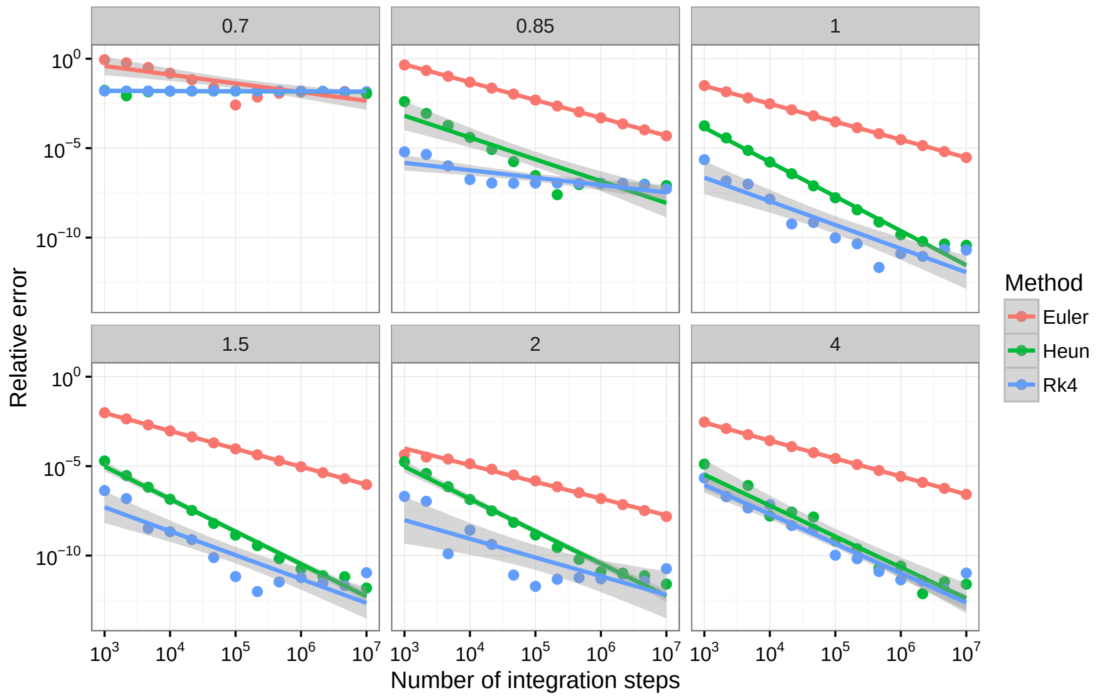


Figure 5.3: Bid price: error vs. number of Steps

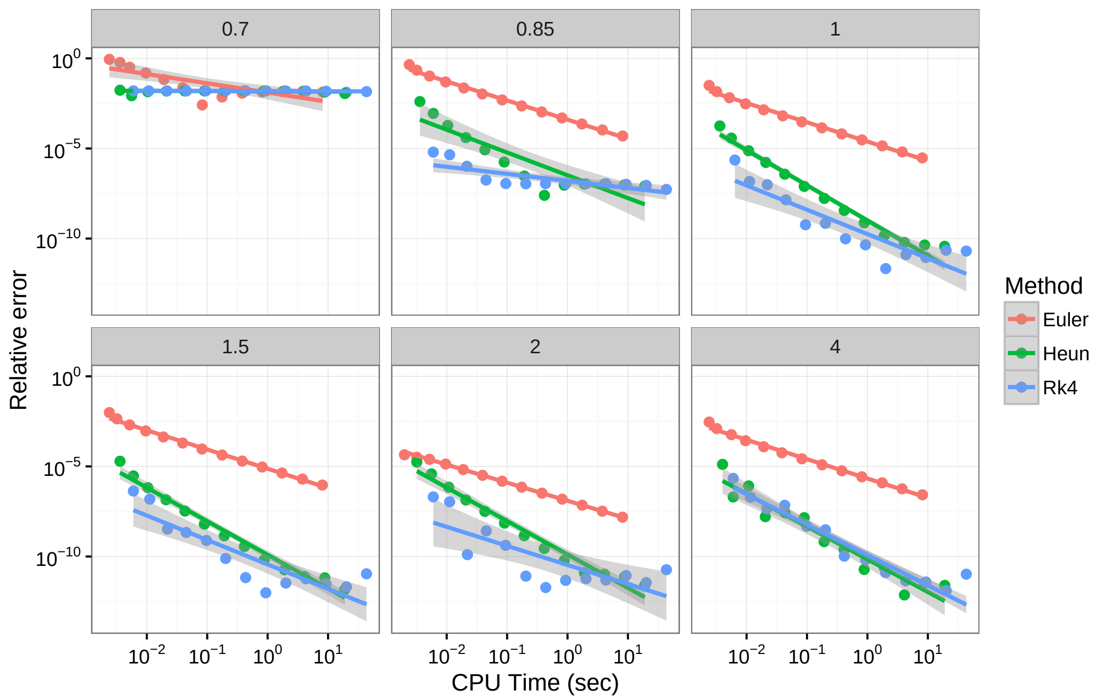


Figure 5.4: Bid price: error vs. CPU time

Demand Factor	Euler	Heun	Rk4
0.7	0.77890		
0.8	0.99344	0.96100	0.84159
1.0	1.00096	1.01791	0.95722
1.5	1.00422	0.99651	0.96421
2.0	1.00441	0.94736	1.02338
4.0	1.00705	1.03205	0.98048

Table 5.3: Slope coefficients of log-linear model fit for derivative error

work required to locate these switches, the integrator will stop at each switching point and will thus take at least one additional integration step for each switching time. As shown in Figs. 5.1 and 5.3, a naive approach with fixed step size and ignoring discontinuities already yields very accurate results with 10^5 steps of fixed size.

5.4.1 Adjoint sensitivity

Since the RHS of Eq. (5.43a) as well as the integrand in Eq. (5.44c) are not continuous at switching times, only linear convergence can be expected when switches are ignored during the solution of the adjoint problem. Numerical results from Example 5.4.1, shown in Table 5.3 and Figs. 5.5 and 5.6, reflect this.

For comparison we again computed a numerical solution of the true derivative up to round-off error for each problem instance. Relative error was measured using the *acute angle*, given by

$$\alpha(x, y) = \frac{x^\top y}{\sqrt{\|x\| \|y\|}}$$

for two vectors x and y, between this solution and the estimates computed by the different integrators at various step sizes.

This error measure is invariant under scaling by a scalar constant and was chosen because, rather than measuring actual differences between vectors, it measures a deviation of directions, which is the relevant quantity if the derivative is used for a Newton-type optimization algorithm, in particular if line-search is used.

In contrast to the results presented in the last section, the classic fourth order Runge-Kutta scheme does not lead to further improvement over the second order Heun method in this case. However, again both higher order methods are more efficient than the explicit Euler method by at least one order of magnitude across all test cases. Both yield acceptable results with errors around 10^{-5} at 10^5 steps with run times in the order of 1 sec.

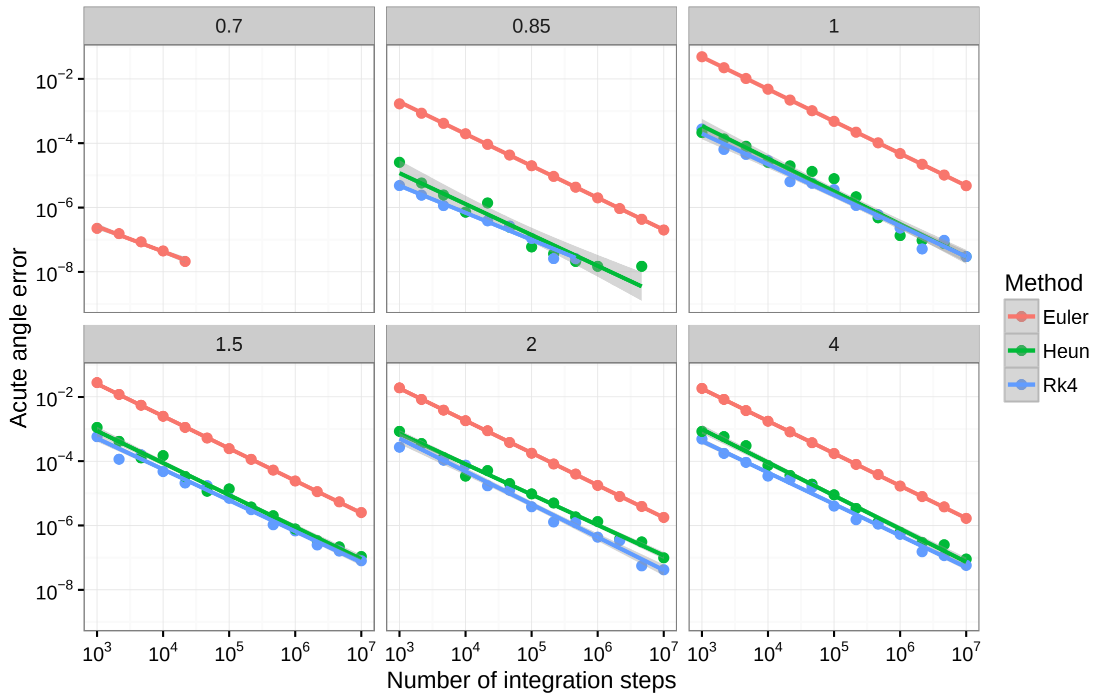


Figure 5.5: Value function derivative: acute angle error vs. number of Steps

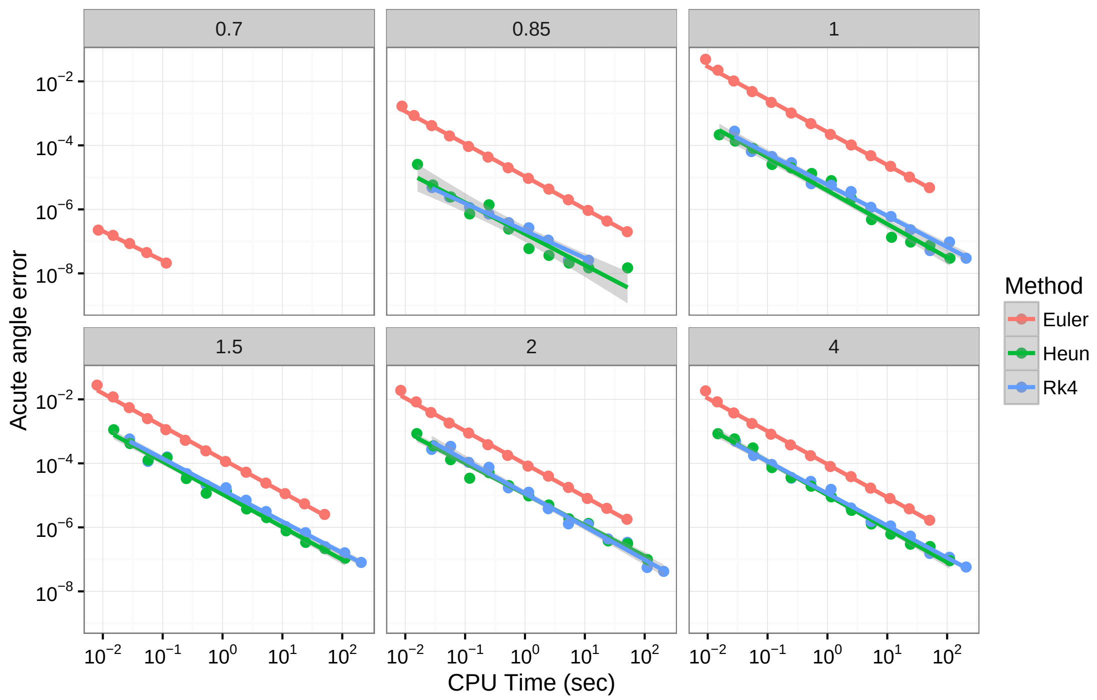


Figure 5.6: Value function derivative: acute angle error vs. CPU time

Chapter 6

The network inventory control problem

In the following, we will extend the results presented in the previous chapter to the network setting, where not all products consume the same single resource. Instead, every product belongs to a certain itinerary, which is a collection of resources that are consumed by one unit of the respective product.

In Section 6.1 we describe an exact solution algorithm via dynamic programming, which is only of theoretical interest, because it is computationally infeasible for realistically sized problems instances due to the curse of dimensionality. In Section 6.2 we describe a well-known decomposition heuristic for the network DP, and use it to introduce a new method for estimating overall expected revenue in the network. We further refine this revenue approximation method and the heuristic control scheme in Section 6.3.

We will make the assumption that demand for a product on one itinerary (or ODI) only depends on the products and availabilities for this product, but is independent of the products offered on other itineraries. In other words, we assume in the demand model that customers do not choose between itineraries, but instead each customer wants to purchase a product on one single itinerary or nothing at all. This assumption allows us to apply the fare transformation of Fiig et al. [41] (see Section 3.5), therefore again reducing the problem to the independent demand case.

Notation

Throughout this chapter, we will use the following notation:

Let m denote the number of different resources (i.e. flight legs) and $C \in \mathbb{N}^m$ the vector of initial capacities. Let

$$A = (a_{r,k})_{\substack{r=1,\dots,m\\k=1,\dots,M}} \tag{6.1}$$

be the *resource consumption matrix*, where $a_{r,k} \in \{0,1\}$ denotes whether one unit of product k consumes a unit of resource r or not. This way, each itinerary is represented by a binary vector of length m indicating which resources are included, and the set of itineraries that are considered in the network is given by the unique columns of A. We will sometimes identify a resource with the set of products consuming this resource, and vice versa we will identify a product with the set of resources it consumes, i.e. we will say $k \in r$ and $r \in k$ if and only if $a_{r,k} = 1$.

Let $\mathcal{P} = \{1,\dots,M\}$ be the set of products and $\mathbb{S} = \wp(\mathcal{P})$ the set of feasible offer sets[1]. Let $\mathbf{y}_k$ be the yield of product k. Let again $[0,T]$ be the booking horizon and let

$$\lambda_k \colon [0,T] \to \mathbb{R} \tag{6.2}$$

$$t \mapsto \lambda_k(t) \tag{6.3}$$

be the time-dependent demand arrival rate of the Poisson process $\mathbf{N}_k$ for product k. Control is exercised via the availability control process $\mathbf{s}$ with values in $\{0,1\}$, where $\mathbf{s}_k(t) = 1$ if product k is

[1] Assuming that all subsets of $\mathcal{P}$ are feasible offer sets is w.l.o.g. by virtue of the demand transformation

available at time t. The booking process

$$\mathrm{d}\mathbf{B}_k(t) = \mathbf{s}_k(t)\,\mathrm{d}\mathbf{N}_k(t). \tag{6.4}$$

for product k is then again a Poisson process with arrival rate at time t equal to $\lambda_k(t)\mathbf{s}_k(t)$.

6.1 Exact formulation

The single-leg problem Eq. (5.5) can be generalized to the network case in a straightforward manner by expanding the state space, which represents remaining capacity, to multiple dimensions. Let $\mathbf{c}$ be the stochastic process on the state space $\mathbf{C} = \times_{r=1}^{m} \{0,\dots,C_r\}$, where $\mathbf{c}_r(t)$ denotes the remaining capacity for resource r at time t.

The dynamic network availability control problem can then be written as the stochastic OCP

$$\begin{aligned}
&\max_{\mathbf{s}} && \mathbb{E}\left[\int_0^T \sum_{k\in\mathcal{P}} \mathbf{s}_k\mathbf{y}_k\,\mathrm{d}\mathbf{N}_k\right] && && \text{(6.5a)}\\
&\text{subject to} && \mathrm{d}\mathbf{c}_r = -\sum_{k\in\mathcal{P}} a_{r,k}\mathbf{s}_k\,\mathrm{d}\mathbf{N}_k && \forall r\in\{1,\dots,m\} && \text{(6.5b)}\\
& && \mathbf{c}(0) = C && && \text{(6.5c)}\\
& && \mathbf{c}_r(T) \ge 0 && \text{a.s. } \forall r\in\{1,\dots,m\} && \text{(6.5d)}
\end{aligned}$$

Remaining inventory $\mathbf{c}$ is again a controlled Markov process. Therefore, optimal controls $\mathbf{s}^*(t)$ at time t only depend on the current state $\mathbf{c}(t)$, but not on the history of $\mathbf{c}, \mathbf{s}$ or $\mathbf{N}$, and can be represented by a deterministic control scheme

$$\begin{aligned}
s\colon [0,T]\times\mathbf{C} &\to \{0,1\}^{\mathcal{P}} && \text{(6.6a)}\\
(t,c) &\mapsto s(t,c), && \text{(6.6b)}
\end{aligned}$$

mapping a pair of time t and remaining capacity c to a feasible action.

The terminal state constraint Eq. (6.5d) is satisfied by not selling a product that would consume a resource that is already depleted, i.e. by requiring that for all $k\in\mathcal{P}$, $t\in[0,T]$ and $c\in\mathbf{C}$

$$\exists r\in\{1,\dots,m\} : a_{k,r} > c_r \Rightarrow s_k(t,c) = 0. \tag{6.7}$$

Let again

$$\begin{aligned}
V\colon [0,T] &\to \mathbb{R}^{\mathbf{C}} && \text{(6.8a)}\\
t &\mapsto (V_c(t))_{c\in\mathbf{C}} && \text{(6.8b)}
\end{aligned}$$

be the *value function*, where $V_c(t)$ is the expected revenue to come from time t and remaining capacity $c\in\mathbf{C}$, assuming optimal control.

In the single-leg setting, for every time t and remaining capacity c the opportunity cost of capacity, represented by the bid price function π, is the same for all products. In the network case this is not true anymore, because products on different ODIs consume different resources with varying opportunity cost. The bid price for a product on itinerary $\mathbf{a}\in\{0,1\}^m$ is given by

$$\pi_c^{\mathbf{a}}(t) = \begin{cases} V_c(t) - V_{c-\mathbf{a}}(t) & \text{if } c\ge\mathbf{a} \text{ element-wise,}\\ \infty & \text{else.}\end{cases} \tag{6.9}$$

Due to Bellman's optimality principle, the value function is the solution of the HJB-equation

$$\begin{aligned}
\dot V_c(t) &= -\max_{s\in\{0,1\}^{\mathcal{P}}} \sum_{k\in\mathcal{P}} s_k\lambda_k(t)\left[\mathbf{y}_k - \pi_c^{A^k}(t)\right] && \forall c\in\mathbf{C}, t\in[0,T] && \text{(6.10a)}\\
V_c(T) &= 0 && \forall c\in\mathbf{C}. && \text{(6.10b)}
\end{aligned}$$

Therefore, the optimal control process satisfies

$$\mathbf{s}_k^*(t) = \begin{cases} 1 & \text{if } \mathbf{y}_k \ge \pi_{\mathbf{c}(t)}^{A^k}(t),\\ 0 & \text{else.}\end{cases} \tag{6.11}$$

Analogously to Section 5.1.2,, we can replace the random state variable $\mathbf{c}$ by a family of deterministic states that model its distribution. For each possible value $c \in \mathbf{C}$ of $\mathbf{c}$, the component $\mu_c(t) = \mathrm{P}[\mathbf{c}(t) = c]$ is the probability of the system being in the respective state. This way, we can write Eq. (6.5) as an equivalent deterministic OCP

$$\max_{S} \quad \int_0^T \sum_{c \in \mathbf{C}} \mu_c(t) \sum_{k \in \mathcal{P}} S_{c,k} \lambda_k(t) \mathbf{y}_k \, \mathrm{d}t \tag{6.12a}$$

$$\text{subject to} \quad \dot{\mu}_c(t) = -\mu_c(t) \sum_{k \in \mathcal{P}} s_{c,k} \lambda_k(t) + \mu_{c+A^k}(t) \sum_{k \in \mathcal{P}} s_{c+1,k} \lambda_k(t) \tag{6.12b}$$

$$\mu_c(0) = \begin{cases} 1 & \text{if } c = C, \\ 0 & \text{else} \end{cases} \tag{6.12c}$$

$$S_{c,k} = 0 \qquad \forall (c,k) \in \left\{ (c,k) \in \mathbf{C} \times \mathcal{P} \mid A^k > c \right\} \tag{6.12d}$$

where the controls are represented by a measurable matrix valued function $S \colon \mathbb{R} \to \{0,1\}^{\mathbf{C} \times \mathcal{P}}$ and we use the convention that $\mu_c \equiv 0$ whenever $c \notin \mathbf{C}$. Here, Eq. (6.12b) models the system dynamics, i.e. the transition rates between possible states of remaining inventory, Eq. (6.12c) defines the initial state of the system, and Eq. (6.12d) ensures that we never sell a product for which we do not have sufficient remaining inventory. With the same arguments as for the single-leg problem one sees the adjoint equation of Eq. (6.12b) is exactly the dynamic program Eq. (6.10). In particular, it is independent of the primal states μ, which means that an optimal solution to Eq. (6.12) can be found by computing the value function V and optimal controls S^* using Eq. (6.10) and then computing the trajectory of the primal states μ as a solution of the IVP defined by Eqs. (6.12b) and (6.12c).

The dynamics across the booking horizon is described by a number of random processes, namely for every resource r the remaining capacity $\mathbf{c}_r$, and for every product k the booking process $\mathbf{B}_k$, the availability control process $\mathbf{s}_k$, and the itinerary bid price $\boldsymbol{\pi}^{A^k}$.

Assuming optimal control, these processes are related over the course of the booking process through a number of effects, which are shown in Fig. 6.1. The full graph of cause and effect relationships for a general network is very complex and hard to visualize. For the sake of simplicity, we have therefore visualized two representative extracts of the network: Figure 6.1a shows the neighborhood of the sub-graph belonging to a product k, where $\left\{ r_1^{(k)}, \dots, r_{m_k}^{(k)} \right\} = \{ r \in R \mid \ a_{r,k} = 1 \}$ is the set of resources consumed by product k, and $\left\{ {r'}_1^{(k)}, \dots, {r'}_{m-m_k}^{(k)} \right\}$ its complement. Figure 6.1b shows the neighborhood of the sub-graph belonging to a resource r, where $\left\{ k_1^{(r)}, \dots, k_{M_k}^{(r)} \right\} = \{ k \in \mathcal{P} \mid \ a_{r,k} = 1 \}$ is the set of products that consume resource r, and $\left\{ {k'}_1^{(r)}, \dots, {k'}_{M-M_r}^{(r)} \right\}$ its complement. Each arrow describes how

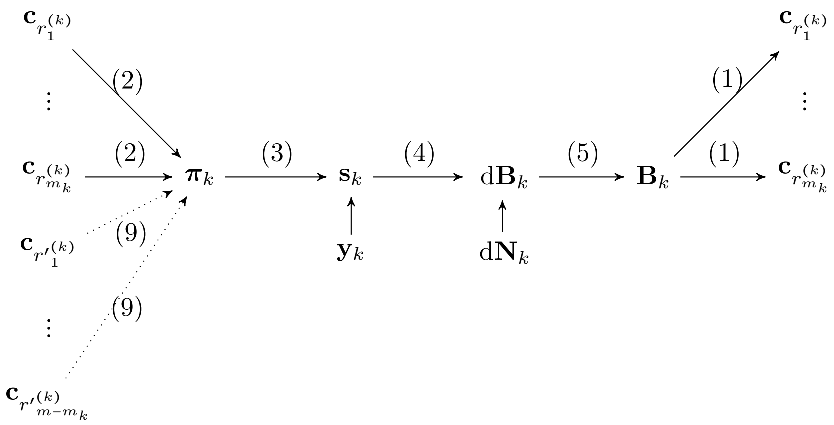
(1) **O&D Bookings reduce remaining inventory.** Every booking of product k until time t will reduce the number of remaining seats $\mathbf{c}_r$ for all flight legs $r \in k$ (see Eq. (6.5b)).

(2) **Remaining inventory has an effect on bid prices for the related products.** Reducing remaining inventory $\mathbf{c}_r(t)$ for resource r while leaving other capacities unchanged will lead to a higher bid price $\pi_{\mathbf{c}}^{A^k}(t)$ for all products using this resource. This is due to a generalization of Proposition 5.1.1 to the network case that holds under weak assumptions.

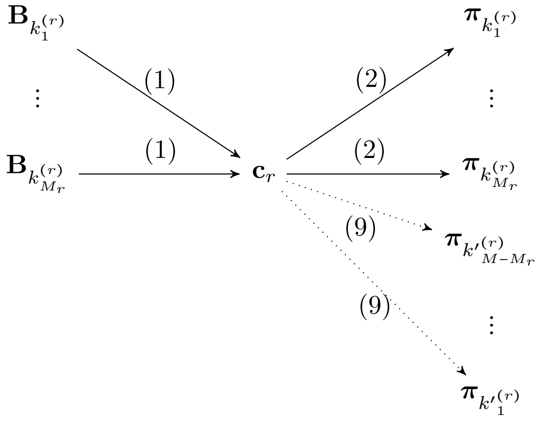
(3) **Bid prices influence the controls.** By virtue of the optimal control policy Eq. (6.11), a higher bid price will (on average) lead to more restrictive controls $\mathbf{s}_k^*(t)$, i.e. close certain booking classes. Note that, due to discrete nature of the problem with a limited number of booking classes, controls might also remain unchanged for small changes in the bid price.

(4) **Controls influence booking rates.** With Eq. (6.4), the booking rate $\mathrm{d}\mathbf{B}_k(t)$ directly depends on booking class availability.

(5) **Booking rates have a (delayed) impact on the number of bookings to expect for the future.** Of course, by definition, the booking process $\mathbf{B}_k$ depends on the booking rate $\mathrm{d}\mathbf{B}_\pi$.



(a) Itinerary view



(b) Resource view

Figure 6.1: Dynamic network control system

As a consequence of these first order effects, we have a number of second order effects:

(6) **Remaining inventory is positively correlated between flights with common itineraries.** This is a direct consequence of Item (1).

(7) **The number of bookings is negatively correlated between itineraries sharing a common resource.** This follows from a combination of the above effects: every booking on a given itinerary will lead to lower remaining inventory for the related resources and therefore decrease booking rates for other itineraries competing for this resource.

(8) **Booking rates for two itineraries sharing a resource are positively correlated.** This follows from the fact that both booking rates are positively correlated with the remaining capacity of the shared resource.

(9) **Remaining inventory has an effect on bid prices for unrelated products.** In addition to Item (2), a change in remaining capacity $\mathbf{c}_r$ also has an impact on the bid price of products that do not require resource r. This is because the increased bid price for some products (Item (2)) reduces the expected demand to come for these products (Items (3) to (5)), which in turn makes capacity less scarce for other resources they consume.

For illustration we will use the following simple examples:

Example 6.1.1 (Single-leg) For the single-resource case, which is covered in detail in Chapter 5, the control process, booking process, and remaining capacity interact as shown in Fig. 6.2.

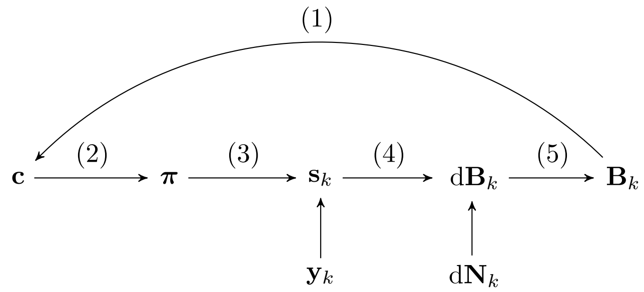


Figure 6.2: Dynamic single leg control system

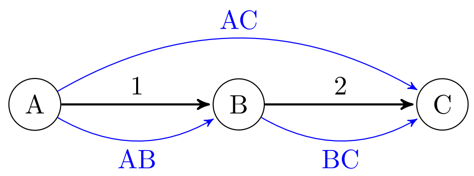


Figure 6.3: Simple two leg network in Example 6.1.2

Example 6.1.2 (Simple two-leg network) Consider the network shown in Fig. 6.3, consisting of three airports A, B and C, one flight from A to B and one from B to C (shown in black) and the three itineraries displayed in blue. The full diagram of network effects for this network is shown in Fig. 6.4.

Let C_1 and C_2 be the initial capacities of the flight legs 1 and 2 respectively. Let M_{AB}, M_{AC} and M_{BC} be the number of products on the three flight itineraries. Then the network DP has $(C_1 + 1)(C_2 + 1)$ states and $(C_1 + 1)(C_2 + 1)(M_{\mathrm{AB}} + M_{\mathrm{AC}} + M_{\mathrm{BC}})$ (time-dependent) controls—one control function for each product and state.

In the following, we will also use the following hub-and-spoke network in order to illustrate how our heuristic scales with growing network size.

Example 6.1.3 (Hub-and-spoke network) Consider a hub-and-spoke network, which consists of one central hub, n origins $O_1, \dots, O_n$ and m destinations $D_1, \dots, D_m$ (see Fig. 6.6). From each origin to the hub and from the hub to each destination there is one flight. We will call these flights feeders and de-feeders respectively. All feeders can be combined with all de-feeders for a transfer itinerary. Ignoring the traffic flows from the origins to the hub and from the hub to the destinations, this means that there are nm different itineraries in the network, three examples of which are shown in blue in the diagram.

Assuming the same initial capacity C for all flight legs and the same number of booking classes M on each itinerary, the network DP has $(C + 1)^{n+m}$ states and $Mnm(C + 1)^{n+m}$ (time-dependent) controls. For our analysis we will assume that demand is equal for all traffic flows, and that total expected demand is proportional to the total capacity nC on the feeders. Therefore, demand for each itinerary is proportional to $\frac{C}{m}$.

Since the network only contains transfer itineraries, it is clear that network effects play a significant role in this network. At the same time, for large n and m, the interdependency between a specific pair of feeder and de-feeder is small, because they are only directly coupled via the itinerary connecting the two.

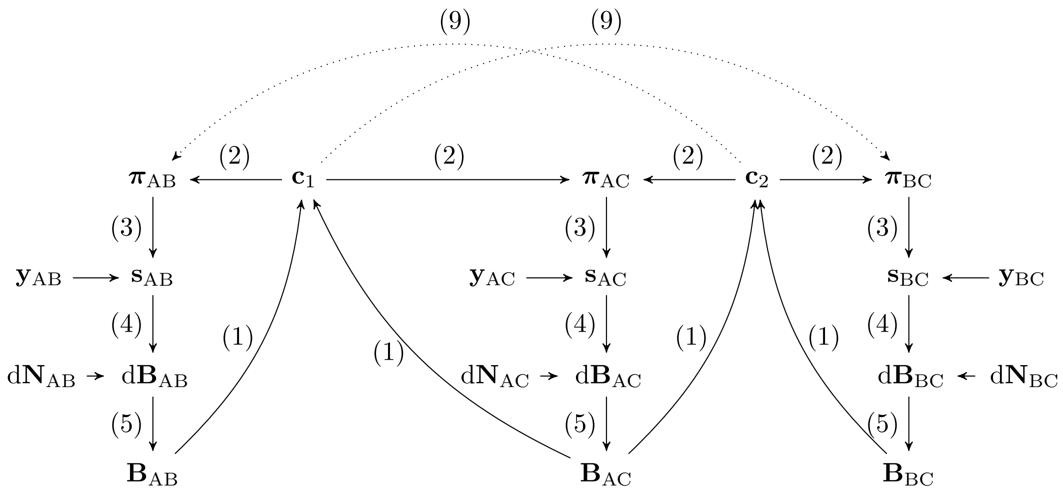


Figure 6.4: Dynamic network control system in Example 6.1.2

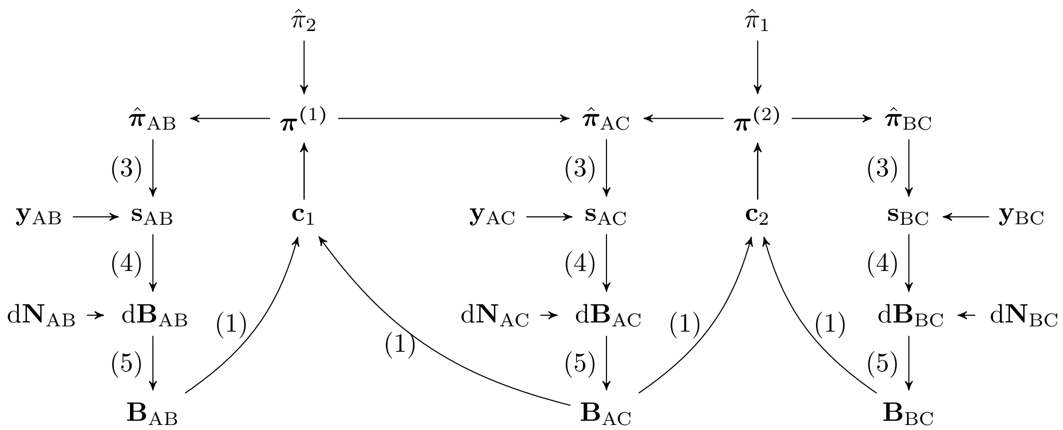


Figure 6.5: Dynamic control system with network decomposition for Example 6.1.2

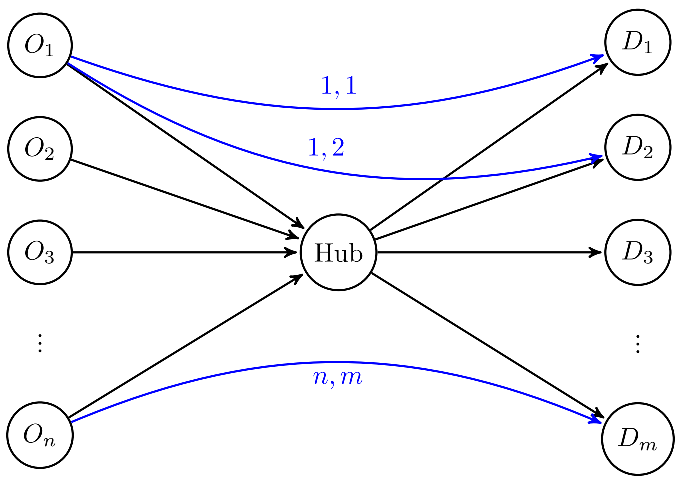


Figure 6.6: Hub and spoke network in Example 6.1.3

6.2 LP-DP network decomposition

Although in theory the network problem is only slightly more complex than the single resource case, it is practically impossible to solve the DP Eq. (6.10) for even a handful of resources due to the curse of dimensionality (the state space $\mathbf{C}$ grows exponentially in the number of resources). As described in Section 3.4.4, there are a number of heuristics that attempt to approximate the solution of the network problem. In the standard RM literature, the goal is usually to approximate the optimal control scheme and to actually compute near-optimal booking class availabilities. This is achieved by combining an approximation of the bid prices with an appropriate control scheme. In order to solve the network pricing problem, we additionally need to be able to efficiently approximate the expected revenue for the whole network as precisely as possible. Furthermore, we would like the approximation to be continuously differentiable in the parameters that define yields $\mathbf{y}$ and arrival rates λ, and we want to be able to compute a good approximation of this gradient.

Of the solution approaches described in Section 3.4.4, we choose to build on a variant of the well-known LP-DP decomposition, because all other ideas have significant drawbacks:

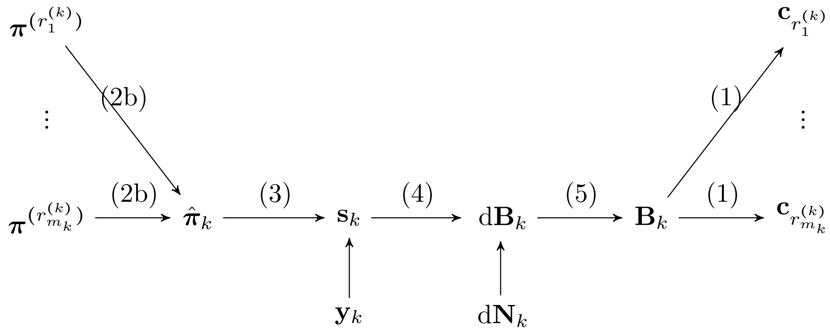
- One of the benefits of having multiple price points is the ability to react dynamically to the realization of the booking process. This rules out the deterministic approximation Eq. (3.53), which ignores these dynamics.
- The prorate method depends on a prorating scheme, which has to be chosen somewhat arbitrarily and does not necessarily reflect network effects correctly.
- DAVN yields a primary control scheme (i.e. availabilities), but no estimate for total expected revenue in the network. Furthermore, DAVN clusters products in a finite number of value buckets based on their yield and approximate displacement costs. Due to this discrete clustering, the results are not necessarily continuously differentiable w.r.t. parameters.
- The functional approximation of the value function is based on a large scale LP, whose objective function is expected network revenue. Therefore, an estimate for expected revenue is provided naturally. Furthermore, its gradient w.r.t. parameters is readily available via LP sensitivity analysis. However, run-time of the solution algorithm seems to scale quadratically with the number of resources in the network, which prohibits its use for realistically sized networks with thousands of resources.

The LP-DP decomposition approximates network bid prices from the solution of a number of appropriately constructed single-leg problems, but at first it does not provide an estimate for overall expected revenue for the network. As shown by Zhang and Adelman [126], each of the underlying single-leg problems provides an upper bound on the optimal expected revenue. However, each of these problems only considers the demand dynamics for the respective flight leg, while the capacity constraints for all other resources are treated deterministically. Therefore, as the number of resources in the network increases these upper bounds will asymptotically approach the objective function value of the deterministic network LP, therefore ignoring demand stochasticity and control dynamics.

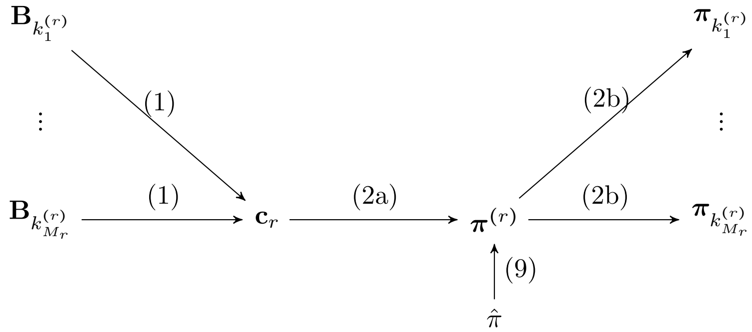
In this section we introduce a new method to approximate the objective function value of the network dynamic program based on the LP-DP decomposition. First, we outline the derivation of the network decomposition, and in particular the construction of the single-leg sub-problems. In Sections 6.2.1 to 6.2.3 we recall the well-known bid price approximation and the corresponding control scheme. These are well-known results [126], that we merely paraphrase in our notation.

Building on the results from Section 5.3 we carry out a sensitivity analysis of the single-leg problems in Section 6.2.4. We then use the adjoint variables of the single-leg problems to approximate the distribution of the random state $\mathbf{c}$ of the network problem. Based on the results from Section 6.2.4, in Section 6.2.5 we present a novel method to estimate overall expected network revenue, which is—unlike the estimates arising from most other network approximations—not an upper bound on total revenue. In addition, we show how an approximate gradient of network revenue can be computed efficiently as well.

The dynamic control system for the decomposition is shown in Fig. 6.7. Comparing it to the dynamics for the exact formulation (see Fig. 6.1), the LP-DP decomposition ignores the second



(a) Itinerary view



(b) Resource view

Figure 6.7: Dynamic control system for network decomposition

order effect Item (9) in the dynamic system, but instead treats this effect deterministically. In addition, the relationship Item (2) between remaining inventory and the dynamic bid prices $\boldsymbol{\pi}_k$ for each product k is simplified. The main dynamic states of the system are bid prices $\boldsymbol{\pi}^{(r)}$ for every resource r, which depend on remaining inventory for the respective resource (2a). The bid price $\boldsymbol{\pi}_k$ for product k is then approximated linearly from the bid prices of the resources used by the product (2b). The LP-DP decomposition heuristic builds on three key insights.

Firstly, the optimal control scheme has the structure of a bid price control (Eq. (6.11)). Therefore we do not have to work with the $\|\mathbf{C}\|M$ availability control functions (see Eq. (6.6)). Instead, if at every time t we can efficiently compute (or approximate) the bid price for each product and every potential state, we can easily derive the optimal control decision from Eq. (6.11).

Secondly, the dynamic program Eq. (6.10) for the value function (and therefore also the bid prices) can be solved to optimality as long as the state space is relatively small, e.g. up to the order of 10^6 possible states, but is practically infeasible for larger problems. With capacities of about 100 seats on short-haul flights and up to 600 seats on large long-haul aircraft, this restricts the size of networks problems that we can solve optimally to two or three resources.

Thirdly, in large networks the network effects between two different flight legs or between two different itineraries are usually relatively weak, even if network effects play a significant role for the network as a whole. Since there is no proof for this rather imprecise proposition, in the following we will motivate the statement through an informal explanation of how network effects act in the dynamic control problem Eq. (6.5).

6.2.1 Bid price approximation

The exponential complexity of the network dynamic program is a consequence of the fact that at every time during the booking horizon the bid price for each product depends on the current remaining inventory for every resource in the network, which implies that we have to solve a DP on the full state space $\mathbf{C}$. Our network decomposition heuristic is based on two major simplifications.

Firstly, we remove the second order effect (9) from the system dynamics and instead model it deterministically. This means that for a product k with resource incidence vector A^k we approximate the dynamic bid price with a function that only depends on the remaining capacity of resources $r \in k$:

$$\pi_c^k(t) \approx \pi_{c_{A^k}}^{A^k}(t), \tag{6.13}$$

where the index c_{A^k} includes the components of c belonging to the resources $r \in k$.

Secondly, we further approximate this bid price using a linear combination of the single-resource bid prices. More precisely, at every time t and state c we use the m bid prices for single-resource resource itineraries as the basis for an affine approximation of the other bid prices. In other words, we approximate the bid price for product k as follows:

$$\pi_c^k(t) \approx \sum_{r \in k} \pi_{c_r}^{(r)}(t), \tag{6.14}$$

where $\pi_{c_r}^{(r)}(t)$ is the approximate bid price for one unit of resource r given on remaining inventory c_r, but independent of the remaining capacity for other resources. For each resource r the bid price function $\pi^{(r)}(t)$ should reflect the system dynamics for the respective resource as well as possible. Although it does not depend on the remaining capacity of other resources, it should also include network effects, if only in a deterministic manner. In the following, we will show how such a bid price estimate can be computed from a single-leg DP derived from the original network problem.

In order to simplify notation, we will look at the first resource and compute $\pi_{c_{(1)}}^1(t)$. In addition, we assume w.l.o.g. that resource 1 is used by the first M_1 products and not required by the other $M - M_1$ products, i.e.

$$a_{1,k} = \begin{cases} 1 & \text{if } k \leq M_1, \\ 0 & \text{else.} \end{cases} \tag{6.15}$$

We reduce the state space by treating the other $m-1$ resources deterministically, i.e. we relax the stochastic network OCP Eq. (6.5) as described in the following.

Firstly, we replace the respective resource constraints Eq. (6.5d) by their expected value. Integrating Eq. (6.5b) and substituting we obtain the stochastic OCP

$$\max_{\mathbf{s}} \quad \mathbb{E}\left[\int_0^T \sum_{k \in \mathcal{P}} \mathbf{s}_k \mathbf{y}_k \, \mathrm{d}\mathbf{N}_k\right] \tag{6.16a}$$

$$\text{subject to} \qquad \mathrm{d}\mathbf{c}_1 = -\sum_{k \in \mathcal{P}} a_{1,k} \mathbf{s}_k \, \mathrm{d}\mathbf{N}_k = -\sum_{k=1}^{M_1} \mathbf{s}_k \, \mathrm{d}\mathbf{N}_k \tag{6.16b}$$

$$\mathbf{c}_1(0) = C_1 \tag{6.16c}$$

$$\mathbf{c}_1(T) \geq 0 \qquad \text{a.s.} \tag{6.16d}$$

$$0 \leq C_r - \mathbb{E}\left[\int_0^T \sum_{k \in \mathcal{P}} a_{r,k} \mathbf{s}_k \, \mathrm{d}\mathbf{N}_k\right] \qquad \forall r \in \{2, \ldots, m\}. \tag{6.16e}$$

where the second equality in Eq. (6.16b) follows from Eq. (6.15).

We then use a Lagrange relaxation on the deterministic constraints Eq. (6.16e) and obtain—for fixed set of Lagrange multipliers $\hat{\pi} = (\hat{\pi}_2, \ldots, \hat{\pi}_m) \in \mathbb{R}^{m-1}$—the objective function

$$\mathbb{E}\left[\int_0^T \sum_{k \in \mathcal{P}} \mathbf{s}_k \mathbf{y}_k \, \mathrm{d}\mathbf{N}_k\right] + \sum_{r=2}^{m} \hat{\pi}_r \left(C_r - \mathbb{E}\left[\int_0^T \sum_{k \in \mathcal{P}} a_{r,k} \mathbf{s}_k \, \mathrm{d}\mathbf{N}_k\right]\right) \tag{6.17a}$$

$$= \mathbb{E}\left[\int_0^T \sum_{k \in \mathcal{P}} \mathbf{s}_k \mathbf{y}_k \, \mathrm{d}\mathbf{N}_k\right] - \mathbb{E}\left[\int_0^T \sum_{r=2}^{m} \sum_{k \in \mathcal{P}} \hat{\pi}_r a_{r,k} \mathbf{s}_k \, \mathrm{d}\mathbf{N}_k\right] + \sum_{r=2}^{m} \hat{\pi}_r C_r \tag{6.17b}$$

$$= \mathbb{E}\left[\int_0^T \sum_{k \in \mathcal{P}} \mathbf{s}_k \left(\mathbf{y}_k - \sum_{r=2}^{m} \hat{\pi}_r a_{r,k}\right) \mathrm{d}\mathbf{N}_k\right] + \sum_{r=2}^{m} \hat{\pi}_r C_r. \tag{6.17c}$$

Here, instead of the yield $\mathbf{y}_k$ for each product we have the term

$$\mathbf{y}_k^{(1)}(\hat{\pi}) := \mathbf{y}_k - \sum_{r=2}^{m} \hat{\pi}_r a_{r,k}, \tag{6.18}$$

which is the yield of the product reduced by a deterministic approximation of the bid prices of other resources it consumes. When the bid prices $\hat{\pi}$ are interpreted as marginal cost of a booking, this is the contribution margin of the product for the network.

Definition 6.2.1 (Network contribution) Let $\hat{\pi} \in \mathbb{R}^m$ be a deterministic vector of bid prices for all resources. Then for every resource r and every product k we call

$$\mathbf{y}_k^{(r)}(\hat{\pi}) := \mathbf{y}_k - \sum_{r' \neq r} \hat{\pi}_{r'} a_{r',k}. \tag{6.19}$$

the *network contribution* or *displacement-adjusted yield* of product k from the perspective of resource r. In addition, let

$$\mathbf{y}_k^{(r)+}(\hat{\pi}) := \max\left\{\mathbf{y}_k^{(r)}(\hat{\pi}), 0\right\} \tag{6.20}$$

denote the maximum of the network contribution and zero.

We will often omit the dependence on $\hat{\pi}$ in order to simplify notation.

Remark 6.2.2 (Contribution of disconnected products) In the relaxed optimization problem the yields, demand rates and control variables for products that do not require resource 1 (i.e. for which $k > M_1$) do not enter into the remaining constraints for resource 1 and are therefore not coupled with each other or with the terms for $k \leq M_1$. The respective control decisions are therefore static (independent of the random state $\mathbf{c}_1$) and only depend on $\hat{\pi}$ and t: for every $t \in [0, T]$ and every $k = M_1 + 1, \ldots, M$ the optimal control is given by

$$\mathbf{s}_k^*(t) = \begin{cases} 1 & \text{if } \mathbf{y}_k > \sum_{r=2}^m \hat{\pi}_r a_{r,k} \\ 0 & \text{else.} \end{cases} \tag{6.21}$$

The contribution of product $k \in \{M_1 + 1, \ldots, M\}$ to the optimal objective function value is

$$\mathbb{E}\left[\int_0^T \mathbf{s}_k^* \left(\mathbf{y}_k - \sum_{r=2}^m \hat{\pi}_r a_{r,k}\right) \mathrm{d}\mathbf{N}_k\right] = \mathbf{y}_k^{(1)+} \mathbb{E}\left[\int_0^T \mathrm{d}\mathbf{N}_k\right]. \tag{6.22}$$

Leaving out these terms and the constant term

$$\sum_{r=2}^m \hat{\pi}_r C_r, \tag{6.23}$$

we are left with the stochastic OCP

$$\max_{\mathbf{s}} \quad \mathbb{E}\left[\int_0^T \sum_{k=1}^{M_1} \mathbf{s}_k \mathbf{y}_k^{(1)} \,\mathrm{d}\mathbf{N}_k\right] \tag{6.24a}$$

$$\text{subject to} \qquad \mathrm{d}\mathbf{c}_1 = -\sum_{k=1}^{M_1} \mathbf{s}_k \,\mathrm{d}\mathbf{N}_k \tag{6.24b}$$

$$\mathbf{c}_1(0) = C_1 \tag{6.24c}$$

$$\mathbf{c}_1(T) \geq 0 \qquad \text{a.s.} \tag{6.24d}$$

This problem has the same structure as the dynamic single-leg availability control problem Eq. (5.5) and can therefore be solved efficiently using the methods described in Chapter 5.

Clearly, we have an analogous problem for the other resources $r = 2, \ldots, m$ as well. In the following, given a resource r, let $\mathcal{P}_r = \{k \mid a_{r,k} = 1\}$ be the set of products consuming resource r.

Definition 6.2.3 (Single-leg OCP) For a given vector of Lagrange multipliers $\hat{\pi} \in \mathbb{R}^m$ and a given resource r, we call

$$\begin{array}{lll} \max\limits_{\mathbf{s}} & \mathbb{E}\left[\int_0^T \sum\limits_{k \in \mathcal{P}_r} \mathbf{s}_k \mathbf{y}_k^{(r)}(\hat{\pi}) \,\mathrm{d}\mathbf{N}_k\right] & \\ \text{s.t.} & \mathrm{d}\mathbf{c}_r = -\sum\limits_{k \in \mathcal{P}_r} \mathbf{s}_k \,\mathrm{d}\mathbf{N}_k & \forall t \in [0, T] \\ & \mathbf{c}_r(0) = C_r & \\ & \mathbf{c}_r(T) \geq 0 & \end{array} \tag{$\mathbf{SOCP}_r(\hat{\pi})$}$$

the *single-leg stochastic OCP* for resource r.

As described in the previous chapters, this problem is usually solved by solving its dual problem:

Definition 6.2.4 (Single-leg DP) For a given vector of Lagrange multipliers $\hat{\pi} \in \mathbb{R}^m$ and a given resource r, we call

$$\begin{aligned} \dot{V}_c^{(r)}(t) &= -\max_{s \in \{0,1\}^{\mathcal{P}_r}} \sum_{k \in \mathcal{P}_r} s_k \lambda_k(t) \left(\mathbf{y}_k^{(r)}(\hat{\pi}) - \pi_c^{(r)}(t) \right) \\ V_c^{(r)}(T) &= 0 \\ \pi_c^{(r)}(t) &= \begin{cases} V_1^{(r)}(t) & \text{if } c = 1, \\ V_c^{(r)}(t) - V_{c-1}^{(r)}(t) & \text{else} \end{cases} \end{aligned} \tag{$\mathbf{DP}_r(\hat{\pi})$}$$

the *single-leg DP* for resource r.

From solving ($\mathbf{DP}_r(\hat{\pi})$) we obtain the value function $V^{(r)}(\cdot;\hat{\pi})$ and, more importantly, the bid price function $\pi^{(r)}(\cdot;\hat{\pi})$. We will omit the dependence on $\hat{\pi}$ whenever it is not of particular importance.

As described in Section 5.1.2, we can write ($\mathbf{SOCP}_r(\hat{\pi})$) as a deterministic OCP

$$\begin{aligned} \max_{S} \quad & \int_0^T \sum_{c=0}^{C_r} \mu_c^{(r)}(t) \sum_{k \in \mathcal{P}_r} S_{c,k}(t) \lambda_k(t) \mathbf{y}_k^{(r)}(\hat{\pi}) \, \mathrm{d}t \\ \text{s.t.} \quad & \dot{\mu}_c^{(r)}(t) = -\mu_c^{(r)}(t) \sum_{k \in \mathcal{P}_r} S_{c,k}(t) \lambda_k(t) \\ & \qquad + \mu_{c+1}^{(r)}(t) \textstyle\sum_{k \in \mathcal{P}_r} S_{c+1,k}(t) \lambda_k(t) \\ & \mu_c^{(r)}(0) = \begin{cases} 1 & \text{if } c = C_r, \\ 0 & \text{else} \end{cases} \\ & S_{0,\cdot} = 0 \end{aligned} \tag{$\mathbf{OCP}_r(\hat{\pi})$}$$

where the controls are represented by a measurable function $S: \mathbb{R} \to \{0,1\}^{\{0,\ldots,C_r\} \times \mathcal{P}_r}$.

When optimal controls S are known (i.e. from a solution of ($\mathbf{DP}_r(\hat{\pi})$)), this reduces to the IVP

$$\dot{\mu}_c^{(r)}(t) = -\mu_c^{(r)}(t) \sum_{k \in \mathcal{P}_r} S_{c,k}(t) \lambda_k(t) + \mu_{c+1}^{(r)}(t) \sum_{k \in \mathcal{P}_r} S_{c+1,k}(t) \lambda_k(t) \tag{6.25a}$$

$$\mu_c^{(r)}(0) = \begin{cases} 1 & \text{if } c = C_r, \\ 0 & \text{else.} \end{cases} \tag{6.25b}$$

6.2.2 Choice of displacement costs

The single-leg problem and its solution—and therefore also the resulting control strategies—depend on the vector of Lagrange multipliers $\hat{\pi}$. In the following we will describe three natural choices for $\hat{\pi}$.

Solution of Lagrange dual problem

So far we have only discussed the relaxed problem obtained from the Lagrange relaxation. In addition, we of course have the Lagrange dual function and the outer optimization problem of minimizing this Lagrange dual.

Definition 6.2.5 (Single-leg Lagrange dual function) The optimal objective function value $F^{(r)}(\hat{\pi}) := V_{C_r}^{(r)}(0;\hat{\pi})$ together with the deterministic terms from Eqs. (6.22) and (6.23) form the Lagrange dual function

$$L^{(r)}(\hat{\pi}) = F^{(r)}(\hat{\pi}) + \sum_{k \in \mathcal{P} \setminus \mathcal{P}_r} \mathbf{y}_k^{(r)+}(\hat{\pi}) \mathbb{E}\left[\int_0^T \mathrm{d}\mathbf{N}_k \right] + \sum_{r' \neq r}^{m} \hat{\pi}_{r'} C_{r'}, \tag{6.26}$$

which for every $\hat{\pi} \geq 0$ provides an upper bound on expected total revenue for the network.

One obvious choice for $\hat{\pi}$ is therefore the optimal solution of the *Lagrange dual problem*

$$\begin{array}{rcl} \min\limits_{\hat{\pi}\in\mathbb{R}^m} & L^{(r)}(\hat{\pi}) & \\ \text{s.t.} & \hat{\pi} \;\; \geq & 0, \end{array} \tag{6.27}$$

which will provide the lowest upper bound that can be obtained from the single-leg problem for resource r.

When computing a control strategy for the network, there are two possible choices. One can either solve the Lagrange dual problem for every resource in the network and use the optimal solution to compute the bid price function for the respective resource. This will lead to bid price functions that are somewhat inconsistent, because they are based on different Lagrange multipliers. Alternatively, one can heuristically choose one vector of Lagrange multipliers for the whole network, e.g. by using the optimal solution to the Lagrange dual problem that provides the lowest upper bound, or by solving a single optimization problem such as, for example,

$$\begin{array}{rcl} \min\limits_{\hat{\pi}\in\mathbb{R}^m} & \frac{1}{m}\sum\limits_{r\in R} L^{(r)}(\hat{\pi}) & \\ \text{s.t.} & \hat{\pi} \;\; \geq & 0, \end{array} \tag{6.28}$$

minimizing the average of the Lagrange dual functions across all resources.

Solution of a deterministic network problem

A common choice in RM practice is the vector of dual variables in an optimal solution to a fully deterministic relaxation of the network problem, which is established by simply replacing the resource constraints of Eq. (6.5) by their expected values:

$$\max_{\mathbf{s}} \quad \mathbb{E}\left[\int_0^T \sum_{k\in\mathcal{P}} \mathbf{s}_k \mathbf{y}_k \, \mathrm{d}\mathbf{N}_k\right] \tag{6.29a}$$

$$\text{s.t.} \quad \mathbb{E}\left[\int_0^T \sum_{k\in\mathcal{P}} a_{r,k} \mathbf{s}_k \, \mathrm{d}\mathbf{N}_k\right] \leq C_r \qquad \forall r \in \{1,\dots,m\}. \tag{6.29b}$$

With the same arguments as in Remark 6.2.2 one easily sees that there is an optimal solution to Eq. (6.29) with deterministic, constant controls. The problem can therefore be written as an LP with constant coefficients and can easily be solved using state-of-the-art software. The vector $\hat{\pi}$ of Lagrange multipliers for the capacity constraints can then be used as the parameter vector for the single resource problems.

Fixed point of bid price mapping

Over the course of the booking horizon, the bid price for one unit of a product with resource consumption vector $\mathbf{a}$ is fully determined by the time to departure and remaining capacity. If remaining capacity $\mathbf{c}(t)$ is not known but considered random, the bid price $\pi^{\mathbf{a}}_{\mathbf{c}(t)}(t)$ is a random variable.

Pang et al. [97] show that in problems with continuous capacity the expected bid price $\mathbb{E}\left[\pi^{\mathbf{a}}_{\mathbf{c}(t)}(t)\right]$ is constant over time. Of course, at the beginning of the booking horizon remaining capacity—and therefore also the bid price—is fixed and known, which particularly implies that

$$\forall t \in [0,T] : \pi^{\mathbf{a}}_{C}(0) = \mathbb{E}\left[\pi^{\mathbf{a}}_{\mathbf{c}(t)}(t)\right]. \tag{6.30}$$

When capacity is discrete, as in our case, the current bid price is infinite whenever remaining capacity is not sufficient to provide one unit of the product in question. Its expected value is therefore only well defined conditionally that remaining capacity is sufficient. The authors show that this conditional expected value is still *approximately* constant over time, and that the deviation approaches zero with increasing capacity. This is intuitive, because in a sense a larger discrete capacity leads to a finer discretization for remaining inventory and therefore better approximates the continuous case. The bid price at time $t = 0$ and initial capacity $c = C$ is still deterministic and we therefore have

$$\forall t \in [0,T] : \pi_C^{\mathbf{a}}(0) \approx \mathbb{E}\left[\pi_{\mathbf{c}(t)}^{\mathbf{a}}(t) \mid \mathbf{c}(t) \geq \mathbf{a}\right]. \tag{6.31}$$

In the network decomposition the Lagrange multiplier $\hat{\pi}_r$ is used as a deterministic approximation for the actual current bid price for the respective resource. With Eq. (6.31) it is therefore natural to choose the vector $\hat{\pi}$ in such a way that

$$\forall r \in R : \hat{\pi}_r = \pi_{C_r}^{(r)}(0;\hat{\pi}). \tag{6.32}$$

Proposition 6.2.6
The fixed point equation Eq. (6.32) *has a solution.*

Proof With the inequalities Eqs. (5.12b) and (5.12e) we have

$$0 \leq \pi_{C_r}^{(r)}(0;\hat{\pi}) \leq \max\{\mathbf{y}_k \mid k \in \mathcal{P}\}. \tag{6.33}$$

In addition we know that the bid price is continuous in the parameters, as shown in Section 5.2. The map

$$F\colon [0,\mathbf{y}_{\max}]^m \to [0,\mathbf{y}_{\max}]^m \tag{6.34a}$$

$$\hat{\pi} \mapsto F(\hat{\pi}) = \left(\pi_{C_1}^{(1)}(0;\hat{\pi}),\ldots,\pi_{C_m}^{(m)}(0;\hat{\pi})\right)^{\top} \tag{6.34b}$$

is therefore a continuous map from a compact subset of $\mathbb{R}^n$ set to itself and, by virtue of Brouwer's fixed point theorem [20], has at least one fixed point. □

With additional assumptions on the structure of the network, we have the following, stronger result.

Proposition 6.2.7
If no product in the network consumes more than two resources, i.e. if for every $k = 1,\ldots,M$: $\sum_{r=1}^{m} a_{r,k} \leq 2$, then the map Eq. (6.34) *is a contraction and, by virtue of the Banach fixed point theorem, has a unique fixed point.*

Proof We prove the claim by showing that $\|\frac{\partial F}{\partial \hat{\pi}}\|_\infty < 1$. In other words, we have to show that all row sums of the Jacobian of F have absolute value less than 1. W.l.o.g. we do this only for the first row.

First, note that with Eq. (6.19)

$$\frac{\partial \mathbf{y}_k^{(1)}}{\partial \hat{\pi}_r} = \begin{cases} 0, & \text{if } r = 0, \\ -a_{r,k}, & \text{else.} \end{cases} \tag{6.35}$$

In Example 5.2.3 we have shown that the gradient of the value function w.r.t. to the yield of a product is equal to the expected number of bookings for this product assuming optimal control:

$$\frac{\partial V_C^{(1)}}{\partial \mathbf{y}_k^{(1)}} = \mathbb{E}\left[\int_0^T \mathbf{s}_k^{(1)}\,\mathrm{d}\mathbf{N}_k \mid \mathbf{c}_1(0) = C\right], \tag{6.36}$$

where $\mathbf{s}_k^{(1)}$ is the availability for product k during the solution of the single-leg dynamic program for resource 1 with initial capacity $\mathbf{c}_1(0) = C$. This expected number of bookings has the following properties:

(1) For any product k which does not use resource 1, i.e. with $a_{1,k} = 0$, the expected number of bookings is independent of C. Therefore

$$\forall k \in \mathcal{P} : a_{1,k} = 0 \Rightarrow \frac{\partial V_C^{(1)}}{\partial \mathbf{y}_k^{(1)}} - \frac{\partial V_{C-1}^{(1)}}{\partial \mathbf{y}_k^{(1)}} = 0. \tag{6.37}$$

(2) The bid price at a fixed time t decreases monotonically with capacity. Therefore expected availability and consequently the expected number of bookings do not decrease with increased initial capacity:

$$\forall k \in \mathcal{P} : \mathbb{E}\left[\mathbf{s}_k^{(1)} \mid \mathbf{c}(0) = C\right] \geq \mathbb{E}\left[\mathbf{s}_k^{(1)} \mid \mathbf{c}(0) = C-1\right] \Rightarrow \frac{\partial V_C^{(1)}}{\partial \mathbf{y}_k^{(1)}} \geq \frac{\partial V_{C-1}^{(1)}}{\partial \mathbf{y}_k^{(1)}} \tag{6.38}$$

(3) The sum over all products using resource 1

$$\sum_{k=1}^{M} a_{1,k} \frac{\partial V_C^{(1)}}{\partial \mathbf{y}_k^{(1)}} = \sum_{k=1}^{M} a_{1,k} \mathbb{E}\left[\int_0^T \mathbf{s}_k^{(1)} \, \mathrm{d}\mathbf{N}_k \mid \mathbf{c}_1(0) = C\right] \tag{6.39}$$

is the expected number of bookings for resource 1 given initial capacity $\mathbf{c}_1(0) = C$. When adding one additional unit to the initial capacity, this expected value increases by less than one, because with demand being a Poisson process there is always a nonzero probability of not being able to sell the additional unit. We therefore have

$$0 \leq \sum_{k=1}^{M} a_{1,k} \left(\frac{\partial V_C^{(1)}}{\partial \mathbf{y}_k^{(1)}} - \frac{\partial V_{C-1}^{(1)}}{\partial \mathbf{y}_k^{(1)}} \right) < 1. \tag{6.40}$$

With the definition of the bid price (see $(\mathbf{DP}_r(\hat{\pi}))$), we have

$$\begin{aligned}
\sum_{r=1}^{m} \left\| \frac{\partial \pi_C^{(1)}}{\partial \hat{\pi}_r} \right\| &= \sum_{r=1}^{m} \left\| \left(\frac{\partial V_C^{(1)}}{\partial \mathbf{y}_k^{(1)}} - \frac{\partial V_{C-1}^{(1)}}{\partial \mathbf{y}_k^{(1)}} \right) \frac{\partial \mathbf{y}_k^{(1)}}{\partial \hat{\pi}_r} \right\| && \text{(6.41a)} \\
&= \sum_{r=1}^{m} \left\| - \sum_{k=1}^{M} a_{r,k} \left(\frac{\partial V_C^{(1)}}{\partial \mathbf{y}_k^{(1)}} - \frac{\partial V_{C-1}^{(1)}}{\partial \mathbf{y}_k^{(1)}} \right) \right\| && \text{(6.41b)} \\
&= \sum_{r=2}^{m} \sum_{\substack{k=1 \\ a_{1,k}=1}}^{M} a_{r,k} \left(\frac{\partial V_C^{(1)}}{\partial \mathbf{y}_k^{(1)}} - \frac{\partial V_{C-1}^{(1)}}{\partial \mathbf{y}_k^{(1)}} \right) && \text{(6.41c)} \\
&\leq \sum_{k=1}^{M} \left(\frac{\partial V_C^{(1)}}{\partial \mathbf{y}_k^{(1)}} - \frac{\partial V_{C-1}^{(1)}}{\partial \mathbf{y}_k^{(1)}} \right) && \text{(6.41d)} \\
&< 1 && \text{(6.41e)}
\end{aligned}$$

where the sum in Eq. (6.41c) only runs over the resources $r = 2, \ldots, m$ because $\mathbf{y}_k^{(1)}$ is independent of $\hat{\pi}_1$, and only over the products with $a_{1,k} = 1$ due to Eq. (6.37). The absolute value can be dropped, because all terms are non-negative. The inequality 6.41d holds, because all terms in the sum are non-negative (Eq. (6.38)) and for every product with $a_{1,k} = 1$ we have $\sum_{r=2}^{m} a_{r,k} \leq 1$, due to the assumption that every product uses at most two resources. The last equality is due to Eq. (6.40). □

6.2.3 Heuristic control scheme

In Section 6.2.1 we have shown how to compute functions $\pi^{(r)} : \mathbb{R} \to \mathbb{R}^{\{0,\ldots,C_r\}}$ for every resource r, such that $\pi_c^{(r)}(t)$ approximates the opportunity cost for one unit of the respective resource, given remaining capacity c at time t. We use these functions to approximate bid prices for every product in the network via Eq. (6.14), which, motivated by the optimality condition Eq. (6.11), is then used in the heuristic control scheme

$$S_{c,k}(t; \hat{\pi}) = \begin{cases} 1 & \text{if } \mathbf{y}_k \geq \sum_{r=1}^{m} a_{r,k}, \pi_{c_r(t)}^{(r)}(t; \hat{\pi}) \\ 0 & \text{else} \end{cases} \tag{6.42}$$

for every time $t \in [0, T]$, remaining inventory $c \in \mathbf{C}$ and every product $k \in \mathcal{P}$. Both bid price approximation and the resulting control scheme depend on a vector of approximate deterministic bid prices $\hat{\pi} \in \mathbb{R}^m$.

6.2.4 State distribution approximation

Based on the results from the previous section, in the following we will derive an approximation for the state distribution μ. Throughout this section we will assume that $\hat{\pi} \geq 0$ is fixed. In order to simplify notation we will omit the explicit dependence of $\pi^{(r)}$ and $S_{c,k}$ on this $\hat{\pi}$.

In the construction of the separate single-leg dynamic programs ($\mathbf{DP}_r(\hat{\pi})$) for every resource, we have replaced the capacity constraints for all other resources by their expected value. In particular, we treat randomness of remaining inventory separately between all resources and therefore ignore all correlation between them. In other words, we have implicitly assumed that remaining capacity (when viewed as a random process) is independent between all resources, i.e. that the random variables $\mathbf{c}_r(t)$ and $\mathbf{c}_{r'}(t)$ are independent for every $r \neq r'$. Under this independence assumption, we have

$$\mathrm{P}\left[\mathbf{c}_r(t) = c \wedge \mathbf{c}_{r'}(t) = c'\right] = \mathrm{P}\left[\mathbf{c}_r(t) = c\right] \mathrm{P}\left[\mathbf{c}_{r'}(t) = c'\right]. \tag{6.43}$$

Expanding across all resources, the distribution of $\mathbf{c}(t)$ is given by

$$\mathrm{P}\left[\mathbf{c}(t) = c\right] = \prod_{r=1}^{m} \mathrm{P}\left[\mathbf{c}_r(t) = c_r\right]. \tag{6.44}$$

Once we have solved the single-leg DP for resource r, we can obtain an approximation for the distribution of its remaining inventory over time $\mathbf{c}_r(t)$:

$$\mu_c^{(r)}(t) \approx \mathrm{P}\left[\mathbf{c}_r(t) = c\right], \tag{6.45}$$

where $\mu^{(r)}$ is a solution of Eq. (6.25). Doing this for all resources in the network, we get an approximation for the distribution of $\mathbf{c}(t)$ for the whole network via

$$\mathrm{P}\left[\mathbf{c}(t) = c\right] \approx \mu_c(t) := \prod_{r=1}^{m} \mu_{c_r}^{(r)}(t). \tag{6.46}$$

This is consistent with the computed bid price approximation Eq. (6.14) and based on the same simplifying assumption that $\mathbf{c}_1, \ldots, \mathbf{c}_m$ are mutually independent.

6.2.5 Expected total revenue and its gradient

Along with the bid price approximation Eq. (6.14), the network decomposition provides upper bounds for the expected revenue for the network via Eq. (6.26). However, we have one upper bound for each resource in the network, and each such bound includes stochasticity of remaining inventory only for the respective resource, while treating all other resources deterministically. In order to heuristically solve the network pricing problem, we do not necessarily require an upper bound for network revenue, but would rather have a good estimate thereof. Based on the results from the previous sections we propose a novel approach to approximate the optimal objective function of the dynamic network optimization problem Eq. (6.5)

$$\mathbb{E}\left[\int_0^T \sum_{k \in \mathcal{P}} \mathbf{s}_k \mathbf{y}_k \, \mathrm{d}\mathbf{N}_k\right]. \tag{6.47}$$

Here, the yields $\mathbf{y}_k$ as well as the arrival rate $\lambda_k(t)$ for the Poisson arrival process $\mathbf{N}_k(t)$ are fixed and known.

In the optimization problem Eq. (6.5), controls and expected revenues for different products $k \neq k'$ are coupled through the capacity constraints. However, once we have (estimates for) the distribution of the optimal dual states $\boldsymbol{\pi}(t)$, we can directly deduce the distribution of the random control process $\mathbf{s}_k(t)$. In order to simplify the notation, we only consider the case $k = 1$ and assume that this product consumes the first m' resources, i.e. that $a_{r,1} = 1 \Leftrightarrow r \leq m'$.

The expected revenue for product $k = 1$ is

$$\mathbb{E}\left[\int_0^T \mathbf{s}_1 \mathbf{y}_1 \, \mathrm{d}\mathbf{N}_1\right] = \int_0^T \mathrm{P}\left[\mathbf{s}_1(t) = 1\right] \mathbf{y}_1 \lambda_1(t) \, \mathrm{d}t. \tag{6.48}$$

Assuming the heuristic control scheme Eq. (6.42) and using the approximation 6.45, the probability that product 1 is available at time t is given by:

$$\mathrm{P}\left[\mathbf{s}_1(t) = 1\right] = \mathrm{P}\left[\mathbf{y}_1 \geq \sum_{r=1}^{m} a_{r,1}\pi^{(r)}_{\mathbf{c}_r(t)}(t)\right] \tag{6.49a}$$

$$\approx \sum_{c\in\mathbf{C}} \mu_c(t)\mathbf{1}_{\mathbf{y}_1 \geq \sum_{r=1}^{m'} \pi^{(r)}_{c_r}(t)} \tag{6.49b}$$

$$= \sum_{c_1=1}^{C_1} \cdots \sum_{c_{m'}=1}^{C_{m'}} \prod_{r=1}^{m'} \mu^{(r)}_{c_r}(t)\mathbf{1}_{\mathbf{y}_1 \geq \sum_{r=1}^{m'} \pi^{(r)}_{c_r}(t)}. \tag{6.49c}$$

We show in Section 6.2.6 how these probabilities can be computed with reasonable effort for typical airline networks.

Sensitivity w.r.t. parameters

In order to solve the network pricing problem using gradient based methods, we not only need to evaluate our objective function, but would also like to be able to compute its gradient w.r.t. parameters. More specifically, consider a parametric version of the stochastic optimal control problem Eq. (6.5), where both the yields $\mathbf{y}_k(\mathbf{p})$ as well as the arrival rates $\lambda_k(t;\mathbf{p})$ depend on a parameter vector $\mathbf{p}$. As a consequence, the bid prices $\boldsymbol{\pi}(t;\mathbf{p})$, the state distribution $\mu(t;\mathbf{p})$ and the control process $\mathbf{s}(t;\mathbf{p})$ all depend on $\mathbf{p}$ as well.

In the previous section we have shown that we can approximate the optimal objective function value

$$g(\mathbf{p}) = \mathbb{E}\left[\int_0^T \sum_{k\in\mathcal{P}} \mathbf{s}_k(\mathbf{p})\mathbf{y}_k(\mathbf{p})\,\mathrm{d}\mathbf{N}_k(\mathbf{p})\right] \tag{6.50a}$$

$$= \sum_{k=1}^{M} \int_0^T \mathrm{P}\left[\mathbf{s}_k(t;\mathbf{p}) = 1\right]\mathbf{y}_k(\mathbf{p})\lambda_k(t;\mathbf{p})\,\mathrm{d}t \tag{6.50b}$$

by simply substituting approximations for $\boldsymbol{\pi}$ and μ to approximate $\mathrm{P}\left[\mathbf{s}_k(t;\mathbf{p}) = 1\right]$.

In this section we simply generalize the results from the single-leg case to the network problem. With the same arguments as in Sections 5.2 and 5.3, one obtains an analogous result to Eq. (5.65), namely that

$$\frac{\mathrm{d}g}{\mathrm{d}\mathbf{p}}(\mathbf{p}) = \mathbb{E}\left[\sum_{k=1}^{M}\int_0^T \mathbf{s}_k(t;\mathbf{p})\left(\frac{\mathrm{d}\mathbf{y}_k}{\mathrm{d}\mathbf{p}}(\mathbf{p})\lambda_k(t;\mathbf{p}) + (\mathbf{y}_k(t;\mathbf{p}) - \boldsymbol{\pi}_k(t;\mathbf{p}))\,\frac{\mathrm{d}\lambda_k}{\mathrm{d}\mathbf{p}}(t;\mathbf{p})\right)\mathrm{d}t\right] \tag{6.51}$$

$$\begin{aligned} = \sum_{k=1}^{M}\int_0^T \mathrm{P}\left[\mathbf{s}_k(t;\mathbf{p}) = 1\right]\Bigg(&\frac{\mathrm{d}\mathbf{y}_k}{\mathrm{d}\mathbf{p}}(\mathbf{p})\lambda_k(t;\mathbf{p}) \\ &+ \big(\mathbf{y}_k(t;\mathbf{p}) - \mathbb{E}\left[\boldsymbol{\pi}_k(t;\mathbf{p}) \mid \mathbf{s}_k(t;\mathbf{p}) = 1\right]\big)\frac{\mathrm{d}\lambda_k}{\mathrm{d}\mathbf{p}}(t;\mathbf{p})\Bigg)\,\mathrm{d}t. \end{aligned} \tag{6.52}$$

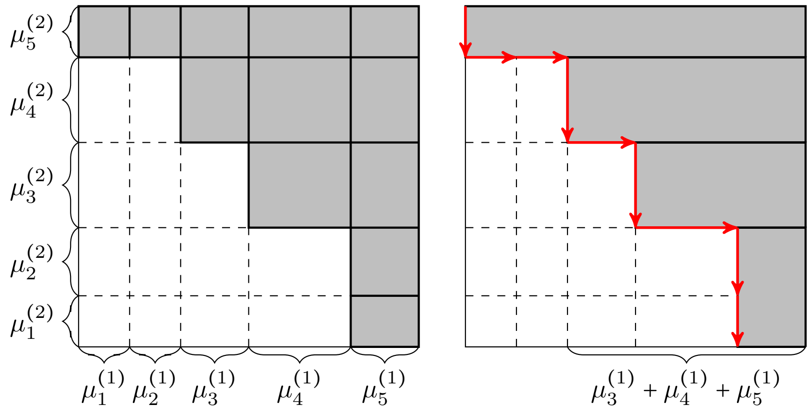
This can be evaluated approximately by substituting the approximations for $\boldsymbol{\pi}, \mu$ and $\mathbf{s}$ arising from the decomposition described above.

6.2.6 Objective function

In this section we describe how to evaluate the objective function approximation from Section 6.2.5 efficiently, if no itinerary in the network uses more than a few resources. This is the case for large airline networks, which often have a hub-and-spoke structure and allow passengers to travel from any origin to any destination with one or at most two stops. Of course, computational effort is especially low for direct travel paths using only one resource.

In order to compute Eq. (6.47) numerically, we have to solve an integral of the form

$$\int_0^T \sum_{k\in\mathcal{P}} \mathrm{P}\left[\mathbf{s}_k(t) = 1\right]\mathbf{y}_k\lambda_k(t)\,\mathrm{d}t. \tag{6.53}$$



(a) Problem structure (b) Iteration over boundary

Figure 6.8: Computation of expected value of product availability

For every product k and every time t, the terms $\mathbf{y}_k$ and $\lambda_k(t)$ are fixed and known and the main effort comes from computing $\mathrm{P}\left[\mathbf{s}_k(t) = 1\right]$ via Eq. (6.49c). For a fixed k and t, we have to compute a sum of the form

$$\mathrm{P}\left[\mathbf{s} = 1\right] \approx \sum_{c_1=1}^{C} \cdots \sum_{c_{m'}=1}^{C} \prod_{r=1}^{m'} \mu_{c_r}^{(r)} \mathbf{1}_{\mathbf{y} \geq \sum_{r=1}^{m'} \pi_{c_r}^{(r)}}, \tag{6.54}$$

where in order to simplify notation we have omitted k and t, and assumed that the product uses the first m' resources. In addition, we have assumed that all resources have the same initial capacity $C_r = C$. This is w.l.o.g. because we can simply set $\mu_{c_r}^{(r)} := 0$ for all $c_r > C_r$.

We will describe the algorithm for the case where $m' = 2$, which is the most common case in typical hub-and-spoke structures. The right-hand-side of Eq. (6.54) is then equal to

$$\sum_{c_1=1}^{C} \sum_{c_2=1}^{C} \mu_{c_1}^{(1)} \mu_{c_2}^{(2)} \mathbf{1}_{\mathbf{y} \geq \pi_{c_1}^{(1)} + \pi_{c_2}^{(2)}}, \tag{6.55}$$

where $\mu^{(1)}, \mu^{(2)}, \pi^{(1)}, \pi^{(2)}$ are all vectors of length C and $\mathbf{y}$ is a constant. Because the bid price vectors $\pi^{(r)}$ are monotonically non-increasing (see Eq. (5.12d)), the index pairs (c_1, c_2) for which $\mathbf{y} \geq \pi_{c_1}^{(1)} + \pi_{c_2}^{(2)}$ holds form an upper right triangle. Computational complexity can be greatly reduced by exploiting this structure.

This is illustrated in Fig. 6.8 for the case $C = 5$, where the rectangles shaded in gray correspond to the index pairs satisfying the bid price inequality. Evaluating Eq. (6.55) is then the problem of computing the gray area. The naive approach of simply iterating over all index pairs—i.e. the gray rectangles in Fig. 6.8a—clearly has complexity $O(C^2)$. Figure 6.8b illustrates our alternative algorithm (see Algorithm 1). We start at the upper left corner and walk along the *boundary* of the shaded area, as indicated by the arrows. On the way, we compute the sum of the $O(C)$ gray rectangles. By using precomputed cumulative sums of the vector $\mu^{(1)}$, total computational complexity is then $O(C)$.

This algorithm generalizes easily to the case with more than two resources. By iterating over the boundary instead of naive enumeration, we can always achieve a complexity of $O(C^{m'-1})$ for $m' \geq 2$. This is sufficient for typical airline networks, where most customers use non-stop or one-stop connections, with rare cases of two or three transfers. However, this algorithm will not be efficient enough for other industries such as hotels or car rentals, where companies frequently sell itineraries with $m' >> 2$.

Algorithm 1 Expected availability

function $\text{ExpAV}(\pi^1, \pi^2, \bar{\mu}^1, \bar{\mu}^2, C, y)$ $\triangleright$ π^1 and π^2 are monotonically decreasing vectors of length C. $\bar{\mu}^1$ and $\bar{\mu}^2$ are reverse sums of μ^1 and μ^2 respectively and therefore monotonically decreasing as well.
  if $\pi^1_C + \pi^2_C > y$ **then return** 0 $\triangleright$ Bid price higher than y for every pair (i, j)
  $i \leftarrow C + 1$
  while $\pi^1_{i-1} + \pi^2_C \leq y$ **do** $\triangleright$ Find upper left corner
    $i \leftarrow i - 1$
  $j \leftarrow C$
  while $j > 0$ **and** $\pi^1_i + \pi^2_{j-1} \leq y$ **do** $\triangleright$ Go down as far as we can
    $j \leftarrow j - 1$
  $x \leftarrow \bar{\mu}^1_i \bar{\mu}^2_j$ $\triangleright$ Area of upper band
  $j' \leftarrow j$ $\triangleright$ Remember vertical position
  while $i \leq C$ **do** $\triangleright$ Move right until the boundary is reached
    if $j \leq 1$ **then return** x $\triangleright$ Exit when the lower boundary is reached
    if $\pi^1_i + \pi^2_{j-1} \leq y$ **then**
      while $j > 0$ **and** $\pi^1_i + \pi^2_{j-1} \leq y$ **do** $\triangleright$ Go down as far as we can
        $j \leftarrow j - 1$
      $x \leftarrow x + \bar{\mu}^1_i (\bar{\mu}^2_j - \bar{\mu}^2_{j'})$ $\triangleright$ Add area of horizontal band
      $j' \leftarrow j$
    $i \leftarrow i + 1$
  return x

6.2.7 Gradient of the objective function

We compute the gradient of expected network revenue by evaluating the integral in Eq. (6.52). For a given parameter vector $\mathbf{p}$ and every product $k \in \mathcal{P}$ and time t, we have to evaluate

$$\mathrm{P}\left[\mathbf{s}_k = 1\right] \left(\frac{\mathrm{d}\mathbf{y}_k}{\mathrm{d}\mathbf{p}} \lambda_k + \left(\mathbf{y}_k - \mathbb{E}\left[\boldsymbol{\pi}_k \mid \mathbf{s}_k = 1\right]\right) \frac{\mathrm{d}\lambda_k}{\mathrm{d}\mathbf{p}} \right), \tag{6.56}$$

where λ, $\mathbf{y}_k$, $\frac{\mathrm{d}\lambda_k}{\mathrm{d}\mathbf{p}}$ and $\frac{\mathrm{d}\mathbf{y}_k}{\mathrm{d}\mathbf{p}}$ are fixed and known. Rearranging the terms we have

$$\mathrm{P}\left[\mathbf{s}_k = 1\right] \left(\frac{\mathrm{d}\mathbf{y}_k}{\mathrm{d}\mathbf{p}} \lambda_k + \left(\mathbf{y}_k\right) \frac{\mathrm{d}\lambda_k}{\mathrm{d}\mathbf{p}} \right) - \mathrm{P}\left[\mathbf{s}_k = 1\right] \mathbb{E}\left[\boldsymbol{\pi}_k \mid \mathbf{s}_k = 1\right] \frac{\mathrm{d}\lambda_k}{\mathrm{d}\mathbf{p}}, \tag{6.57}$$

where the first summand is the gradient of the expected revenue rate of product k at time t. Since it is a constant multiple of $\mathrm{P}\left[\mathbf{s}_k = 1\right]$, it can be computed using the method described in Section 6.2.6.

The second summand is the gradient of the expected opportunity cost rate, which depends on the conditional expected value of $\boldsymbol{\pi}_k$, given that the product is available. Omitting the constant $\frac{\mathrm{d}\lambda_k}{\mathrm{d}\mathbf{p}}$ and with the same assumptions and notation as in Eq. (6.54), this can be approximated via

$$\mathrm{P}\left[\mathbf{s}_k = 1\right] \mathbb{E}\left[\boldsymbol{\pi}_k \mid \mathbf{s}_k = 1\right] \approx \sum_{c_1=1}^{C} \cdots \sum_{c_{m'}=1}^{C} \prod_{r=1}^{m'} \mu^{(r)}_{c_r} \mathbf{1}_{\mathbf{y} \geq \sum_{r=1}^{m'} \pi^{(r)}_{c_r}} \left(\sum_{r'=1}^{m'} \pi^{(r')}_{c_{r'}} \right), \tag{6.58}$$

where this time we do not compute the expected value of an indicator function, but the expected value of the bid price approximation $\boldsymbol{\pi}_k \approx \sum_{r=1}^{m'} \pi^{(r)}_{\mathbf{c}_r}$. We can now use the fact that we linearly approximate the O&D bid price as follows: Rearranging the sums, the right-hand-side of Eq. (6.58) is equal to

$$\sum_{r'=1}^{m'} \left(\sum_{c_1=1}^{C} \cdots \sum_{c_{m'}=1}^{C} \pi^{(r)}_{c_r} \prod_{r=1}^{m'} \mu^{(r)}_{c_r} \mathbf{1}_{\mathbf{y} \geq \sum_{r=1}^{m'} \pi^{(r)}_{c_{r'}}} \right). \tag{6.59}$$

Now, consider the r'-th summand of the outer sum. Replacing the vector $\mu^{(r')}$ with the element-wise product of $\mu^{(r')}$ and $\pi^{(r')}$, while leaving $\mu^{(r)}$ unchanged for all $r \neq r'$, we again have something that looks the same as the right hand side of Eq. (6.54), and can therefore be treated using the algorithm presented in Section 6.2.6.

We have therefore shown that we can evaluate Eq. (6.56) via $m' + 1$ calls to Algorithm 1, which leads to a total computational complexity of $O(m'C^{m'-1})$.

6.3 Probabilistic DP network decomposition

In the classic LP-DP decomposition described in Section 6.2 a deterministic vector of displacement costs $\hat{\pi}$ is used to decompose the network. In other words, when computing the bid price vector for each flight leg we make the simplifying assumption that the bid prices on all other legs are deterministic and constant over time. However, when actually applying the bid price control scheme Eq. (6.42) to a realization of the demand process, the bid prices on all legs depend on the remaining capacities of the respective legs and are therefore random variables. From the point of view of any given flight leg, the network contribution of connecting traffic using this leg varies depending on the bid prices on other legs. In addition, remaining capacities of different resources (and therefore also their bid prices) are correlated, because transfer passengers either book a seat on both legs or none.

The network dynamic program Eq. (3.52) fully addresses both the randomness of remaining capacity and bid prices, and the dependence between these random variables. The single-leg dynamic program ($\mathbf{DP}_r(\hat{\pi})$) with deterministic displacement costs $\hat{\pi}$, on the other hand, ignores both effects. It seems natural that the network decomposition can be improved by incorporating into each single-leg problem the bid prices of other legs as an additional source of variance for the booking process. In the following we describe a variation of the network decomposition, where in the single-leg DP we consider the bid prices of other resources as random variables instead of using a deterministic estimate. In order to keep the problem computationally tractable we still treat them as independent of each other and independent of the remaining capacity of the leg in question.

First, we show in Section 6.3.1 how the single-leg DP can be adapted to use random instead of deterministic displacement costs, and that the structure of the DP does not change at all if the displacement costs have a discrete distribution. We then show in Section 6.3.2 how such a random distribution for the displacement costs can be computed.

6.3.1 Single-leg dynamic program with random displacement costs

Consider a version of the single-leg problem ($\mathbf{DP}_r(\hat{\pi})$), where the deterministic vector of displacement costs $\hat{\pi}$ is replaced with a vector of random displacement costs $\hat{\boldsymbol{\pi}}$. Although the distributions of the true bid prices on all resources are correlated and vary over time t, we assume for simplicity's sake that the components of $\hat{\boldsymbol{\pi}}$ are independently (but not identically) distributed and that their distributions are constant.

If a product consumes only one resource it simply appears in the single-leg DP of the respective resource without any displacement adjustment. In the following consider a product $k \in \mathcal{P}$, which uses resource r and at least one additional resource. With randomly distributed displacement costs, the displacement-adjusted yield (see Definition 6.2.1) of product k on resource r is now a random variable

$$\mathbf{y}_k^{(r)}(\hat{\boldsymbol{\pi}}) = \mathbf{y}_k - \sum_{r' \neq r} \hat{\boldsymbol{\pi}}_{r'} a_{r',k}, \tag{6.60}$$

and the RHS of the ODE for the value function becomes

$$\dot{V}_c^{(r)}(t) = -\mathbb{E}_{\hat{\boldsymbol{\pi}}} \left[\max_{s \in \{0,1\}^{\mathcal{P}_r}} \sum_{k \in \mathcal{P}_r} s_k \lambda_k(t) \left(\mathbf{y}_k^{(r)}(\hat{\boldsymbol{\pi}}) - \pi_c^{(r)}(t) \right) \right], \tag{6.61}$$

where we maximize the expectation of the right hand side over the possible realizations of displacement costs. Now, note that when actually applying the control scheme to a realization of the demand process, the availability for product k always depends on the current bid price for all involved resources. Therefore, in the right hand side of the ODE above we can assume that we can decide on the optimal availability s_k depending on the realization of $\hat{\boldsymbol{\pi}}$.

Just as in the original DP, we have $s_k = 1$ for time t and remaining capacity c if and only if $\mathbf{y}_k^{(r)}(\hat{\boldsymbol{\pi}}) \geq \pi_c^{(r)}(t)$. Substituting into Eq. (6.62) we get

$$\dot{V}_c^{(r)}(t) = -\mathbb{E}_{\hat{\boldsymbol{\pi}}}\left[\sum_{k\in\mathcal{P}_r} \lambda_k(t)\left(\mathbf{y}_k - \sum_{r'\neq r} \hat{\boldsymbol{\pi}}_{r'} a_{r',k} - \pi_c^{(r)}(t)\right)\mathbf{1}_{\mathbf{y}_k \geq \pi_c^{(r)}(t)}\right] \tag{6.62a}$$

$$= \sum_{k\in\mathcal{P}_r} \lambda_k(t)\left(\mathbf{y}_k - \sum_{r'\neq r} \hat{\boldsymbol{\pi}}_{r'} a_{r',k} - \pi_c^{(r)}(t)\right)\mathrm{P}\left[\mathbf{y}_k - \sum_{r'\neq r} \hat{\boldsymbol{\pi}}_{r'} a_{r',k} \geq \pi_c^{(r)}(t)\right]. \tag{6.62b}$$

Now consider the case where product k uses exactly two resources r and r'. Further assume that $\hat{\boldsymbol{\pi}}_{r'}$ has a discrete random distribution on the values $\{\hat{\pi}_{r',1}, \ldots, \hat{\pi}_{r',L}\}$ with probabilities $\alpha_{r',1}, \ldots, \alpha_{r',L}$. Then the summand for product k in the dynamic program above becomes

$$\sum_{k\in\mathcal{P}_r} \lambda_k(t)\left(\mathbf{y}_k - \hat{\boldsymbol{\pi}}_{r'} - \pi_c^{(r)}(t)\right)\mathrm{P}\left[\mathbf{y}_k - \hat{\boldsymbol{\pi}}_{r'} \geq \pi_c^{(r)}(t)\right] \tag{6.63a}$$

$$= \lambda_k(t)\sum_{l=1}^{L} \alpha_{r',l}\left(\mathbf{y}_k - \hat{\pi}_{r',l} - \pi_c^{(r)}(t)\right)\mathbf{1}_{\mathbf{y}_k - \hat{\pi}_{r',l} \geq \pi_c^{(r)}(t)}. \tag{6.63b}$$

This is the same as when product k were to be replaced by L separate products with deterministic yields $\mathbf{y}_k - \hat{\pi}_{r',1}, \ldots, \mathbf{y}_k - \hat{\pi}_{r',L}$ and demand rates $\alpha_{r',1}\lambda_k(\cdot), \ldots, \alpha_{r',L}\lambda_k(\cdot)$. This means that the probabilistic LP-DP decomposition can be implemented very easily if the random displacement costs are discretely distributed, because the DP does not change at all, but one can simply transform the input data by introducing additional products. Since the computational complexity of solving the DP is linear in the number of products, this will increase runtime by a factor of L. The same also directly applies to the dual equation Eq. (6.25), which can be used to compute the state distribution $\mu^{(r)}$.

Products which use more than two resources can theoretically be treated in the same way, but the support of the distribution of the displacement-adjusted yield is essentially the Cartesian product of the sets of possible values for displacement costs of the involved resources. The number of virtual products can therefore quickly become very large.

6.3.2 Computing random displacement costs

Let $\hat{\pi} \in \mathbb{R}^m$ be a vector of displacement costs. As described in Section 6.2.4 we can compute (an approximation of) the distribution of remaining inventory via Eq. (6.25): For every resource r and every time t, $\mu_c^{(r)}(t;\hat{\pi}) \approx \mathrm{P}\left[\mathbf{c}_r(t) = c\right]$ is an estimate for the probability of being in state $\mathbf{c}_r(t) = c$ at time t. Together with the bid price vector $\left(\pi_c^{(r)}(t;\hat{\pi})\right)_{c=1,\ldots,C_r}$ we have an approximation of the discrete distribution of the bid price $\boldsymbol{\pi}^{(r)}(t;\hat{\pi}) = \pi_{\mathbf{c}_r(t)}^{(r)}(t;\hat{\pi})$.

For each resource r the distribution of $\boldsymbol{\pi}^{(r)}$ varies over time t. In order to be able to apply the method described in the previous section, we need to approximate $\boldsymbol{\pi}^{(r)}$ by a constant random variable $\hat{\boldsymbol{\pi}}_r$. To this end, we collapse the time dimension by averaging the time-dependent random variable $\boldsymbol{\pi}^{(r)}(t;\hat{\pi})$ over time to obtain the time-independent random variable

$$\hat{\boldsymbol{\pi}}^{(r)}(\hat{\pi}) := \int_0^T \boldsymbol{\pi}^{(r)}(t;\hat{\pi}) w^{(r)}(t)\,\mathrm{d}t. \tag{6.64}$$

In the above

$$w^{(r)}\colon \mathbb{R} \to \mathbb{R}(t) \tag{6.65}$$

$$t \mapsto w^{(r)}(t) \tag{6.66}$$

is a weighting function that satisfies $\int_0^T w^{(r)}(t)\,\mathrm{d}t = 1$. It is natural to choose $w^{(r)}$ based on the distribution of demand over time, i.e. depending on the arrival rates of the products that use resource r:

$$w^{(r)}(t) := \frac{\sum_{k\in\mathcal{P}} a_{r,k}\lambda_k(t)}{\int_0^T \sum_{k\in\mathcal{P}} a_{r,k}\lambda_k(s)\,\mathrm{d}s} \tag{6.67}$$

The original random variable $\boldsymbol{\pi}^{(r)}(t;\hat{\pi})$ has a discrete distribution for every time t. However, because the support of the distribution varies (continuously) over time, the time-independent $\hat{\boldsymbol{\pi}}^{(r)}(\hat{\pi})$ now has a continuous probability distribution. To further simplify, we approximate this distribution with a finite discrete distribution with L values as follows: First, compute the L-quantiles $\{q_1,\dots,q_{L-1}\}$ of $\hat{\boldsymbol{\pi}}^{(r)}(\hat{\pi})$. Then for each interval $I_1 = (-\infty,q_1], I_2 = (q_1,q_2],\dots,I_L = (q_{L-1},\infty)$ and every $l = 1,\dots,L$ compute the conditional expected value

$$\hat{\pi}^{(r,l)} := \mathbb{E}\left[\hat{\boldsymbol{\pi}}^{(r)}(\hat{\pi}) \mid \hat{\boldsymbol{\pi}}^{(r)}(\hat{\pi}) \in I_l\right]. \tag{6.68}$$

By construction, the uniform distribution on $\{\hat{\pi}^{(r,l)} \mid l = 1,\dots,L\}$ approximates the distribution of $\hat{\boldsymbol{\pi}}^{(r)}(\hat{\pi})$, and approaches it in the limit $L \to \infty$.

In this section we showed how to compute a random vector of displacement costs $\hat{\boldsymbol{\pi}}$ based on bid price functions $\pi^{(r)}$ and the state distribution $\mu^{(r)}$, which can be computed based on a vector of deterministic displacement costs $\hat{\pi}$ via the standard LP-DP-decomposition. With the results of the previous section, we can now use $\hat{\boldsymbol{\pi}}$ in the probabilistic DP-decomposition to compute new bid price functions and new state distributions for all resources, which can again be used to computed updated random displacement costs. This process can be iterated for a fixed number of times or until convergence.

6.4 Numerical Results

In this section we present numerical results on the accuracy of the objective function value estimate described in Eqs. (6.48) and (6.50). The estimate depends on bid price vectors $\pi^{(r)}_{c_r}(t)$ and state distribution approximations $\mu^{(r)}_{c_r}(t)$ for every resource r. By using both the original LP-DP decomposition as well as the probabilistic decomposition introduced in the previous section, we obtain two different value function estimates. We compare both to the upper bounds obtained from the deterministic LP, and the upper bounds obtained from the individual DPs in the LP-DP-decomposition.

In addition we evaluate (in terms of expected revenue) the performance of the control scheme based on the probabilistic LP-DP decomposition compared to the original decomposition.

The software that was used is described in Appendix A.

6.4.1 Scenarios

We can evaluate the quality of the objective function approximation using independent demand scenarios. This is without loss of generality, because the choice-based network dynamic program is equivalent to the independent demand network dynamic program via the fare transformation (Section 3.5). We use a network such as the one in Example 6.1.3 with one hub and m spokes, where half of the spokes are origins and the other half are destinations.

Clearly, for networks with no transfer traffic the network decomposes into a set of single-leg problems and the network decomposition is exact. On the other hand, the network decomposition heuristic can be expected to have the highest error when network effects are strongest. In this simulation we therefore assume that there is no local traffic from the spokes into the hub or from the hub to any of the spokes, but instead that all traffic is transfer traffic.

Each of the $\frac{m^2}{4}$ itineraries has 10 products with yields $\mathbf{y}_k$ drawn randomly from a gamma distribution with expected value 1 and standard deviation $\frac{1}{\sqrt{5}}$ for every product k. For the sake of simplicity we assume that demand arrives homogeneously over the booking horizon, in other words that the demand rates $\lambda_k(t)$ for every product k are constant over time. Scaling the booking horizon to the unit interval, the demand rate is equal to the total demand across the booking horizon.

In order to explore how they affect the quality of the heuristic, we vary the scenario along the following dimensions:

Network size: The number of spokes is $m \in \{4, 8, 16\}$.

Capacity: In every scenario all legs have the same capacity $C \in \{50, 100, 200\}$.

Demand volume: For every product k expected demand λ_k is drawn randomly from a gamma distribution with expected value equal to $\frac{1}{\mathbf{y}_k}$ and standard deviation $\frac{1}{\sqrt{5}\mathbf{y}_k}$, i.e. with coefficient of variation $\frac{1}{\sqrt{5}}$. This way, high yield products have lower expected demand than low yield products. Demand is then scaled by a constant such that total demand for the whole network is a multiple of total capacity. Because every passenger—being a transfer passenger—requires two seats, we compute the demand to capacity ratio as $\alpha = \frac{\sum_{k \in \mathcal{P}} \lambda_k}{2mC}$. In a network with low overall demand there is only a small probability that capacity will be scarce, which implies that bid prices tend to be close to zero regardless of the realization of the demand process, and the optimal control strategy is therefore deterministic: Always open all booking classes with positive yield. Expected revenue can then be computed easily. In our simulations we therefore lean towards cases where demand is higher than capacity and use the demand-to-capacity ratios $\alpha \in \{1, 1.2, 1.5\}$.

With all combinations of network size, capacity and demand we have 27 scenarios. For each scenario we generate 100 problem instances, which differ in the randomly chosen demand and yield of the products.

6.4.2 Optimization

For each of the 2700 problem instances, we solved the following problems based on the actual yields and demand rates (i.e. assuming a perfect forecast):

Deterministic network LP. The deterministic LP Eq. (3.53) yields an upper bound on expected network revenue, which was later compared to the bounds produced by the single-leg DPs and the revenue estimate from the method introduced in Section 6.2.5. In addition, the dual solution $\hat{\pi}$ of the LP was used as displacement cost for the LP-DP-decomposition.

LP-DP decomposition. Based on the dual solution of the LP, we solved the displacement-adjusted dynamic program $(\mathbf{DP}_r(\hat{\pi}))$ and its dual problem Eq. (6.25). The results were used two-fold: Firstly, we simulated the booking process using a bid price control scheme based on the dynamic bid prices computed from the displacement-adjusted DP in order to estimate the actual expected revenue given this control scheme. Secondly, we used the value function and state distribution to compute an estimate of overall network revenue based on the results from Section 6.2.5.

Probabilistic DP decomposition. Based on the solution of the LP-DP decomposition, we applied the results from Section 6.3.2 to compute a random vector of displacement costs $\hat{\boldsymbol{\pi}}$. As described in Section 6.3.1 we used these random displacement costs in the probabilistic DP decomposition to compute new dynamic bid prices and state distribution, and correspondingly updated $\hat{\boldsymbol{\pi}}$. This process was iterated ten times, where in each iteration the continuous distribution of the actual displacement costs was approximated using $L = 10$ discrete values. The bid prices and state distribution from the last iteration were then used both in a simulation (in order to measure the quality of the corresponding control scheme in terms of expected revenue) and to compute a revenue estimate as described in Section 6.2.5 (in order to measure the quality of the revenue estimate in terms of deviation from actual revenue).

In practice, the LP-DP decomposition is often re-solved multiple times during the booking horizon, where demand and capacity in the LP are replaced with demand-to-come for the rest of the booking horizon and remaining capacity respectively. In this study we did not use re-optimization during the booking horizon for either of the two decomposition methods. The effect of re-optimization on the performance of the probabilistic DP decomposition are a potential question for future research.

6.4.3 Simulation

For every problem instance we simulate the booking process for 10000 realizations of the random demand process. Each realization is generated and simulated as follows:

(1) For every product $k \in \mathcal{P}$ draw the number of arrivals $\mathbf{N}_k$ from a Poisson distribution with expected value λ_k.

(2) For every request draw its arrival time t from a uniform distribution on the booking horizon $[0, 1]$, and order the requests by arrival time.

(3) Starting at the initial capacities for each leg, for every request compute the accept/reject decision based on the control strategy Eq. (6.42). If the request is accepted, remaining capacity for the respective flight legs is decreased by one.

6.4.4 Results

Statistical significance

For every scenario type (defined by capacity, number of legs, and demand-to-capacity ratio) the simulation results contain two sources of variance, which we want to differentiate. Firstly, we have 100 random secnario instances, which differ in prices and demands as described in Section 6.4.1. In addition, for each such instance we drew 10000 independent realizations of the demand process, which we will call the runs of the simluation, and simulated the booking process for different control mechanisms. For each scenario instance the control schemes as well as the revenue predictions from the different methods are identical between the 10000 runs, while the actual controls vary between the runs, because the bid price (and therefore also booking class availability) always depends on the currently remaining inventory.

For each scenario instance we would like to compare actual expected revenue for the different control schemes, as well as compare these expected revenues with the predictions provided by the different methods. The variance between the 10000 different runs for the same scenario instance is not interesting for us, but is simply a consequence of the fact that we cannot determine the actual expected revenue analytically but have to estimate it via simulation. We therefore use the observed variance between the achieved revenues for each run to compute the variance of the sample mean, which ideally we would like to be close zero.

The variance arising from the demand and price differences between the 100 scenario instances, on the other hands, is interesting on its own, because it allows us to judge how sensitive the results are with respect to the choice of scenario. Of course, this strongly depends on the way the instances were randomly generated and in particular on the distributions for prices and demands.

Both when analyzing revenue performance of the different methods an when comparing the accuracy of revenue predictions we will show that all results are statistically significant in the sense that increasing the number of simulation runs to higher than 10000 would not alter the findings.

Revenue gain from the probabilistic DP-decomposition

In order to evaluate the quality of the controls generated with the probabilistic DP-decomposition (**DecompProb**) compared to the standard decomposition (**DecompStd**), we simply compare the revenues that were achieved in the simulations using the respective controls. Figure 6.9 shows the relative difference between the revenue that was achieved with **DecompProb** and the revenue achieved with **DecompStd**. The box plot was created as follows: For each of the 2700 scenario instances, we computed the revenue gain of **DecompProb** as the relative difference of the mean revenues that were computed in the 10000 simulation runs for each method, ignoring the statistical error of these sample means (see previous section). In the plot, each boxplot represents one of the 27 scenario types, and the distribution for each scenario type is the distribution of the revenue gain over the 100 scenario instances of the respective type. We see that **DecompProb** consistently leads to revenue improvements of up to 0.2%.

Clearly, the revenue gains are larger for small capacities. This is intuitive, because the strength of **DecompProb** compared to **DecompStd** is the fact that it better accounts for variance in the demand arrival process, and it is well known that with increasing capacity and demand the stochastic network problem behaves more and more like the deterministic problem, i.e. variance of the demand process becomes less relevant.

As the demand to capacity ratio α increases, revenue gains seem to increase. The reasons for this are not clear. One possible explanation is that with higher demand the system is more likely to

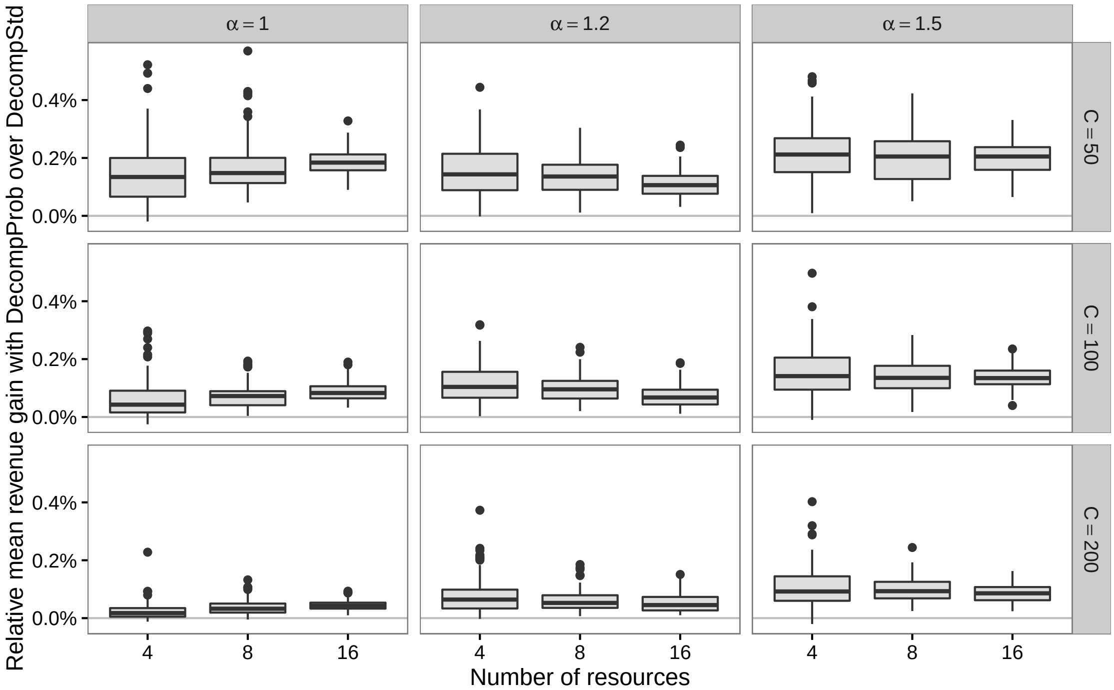


Figure 6.9: Revenue improvement of DecompProb compared to DecompStd

encounter situations of very low remaining capacities, and (as explained in the previous paragraph) **DecompProb** performs better in these situations.

The most important finding, however, is that the revenue gains seem to be largely independent of the number of resources in the network. This means that one can expect that the increased performance translates directly to real world airline networks with hundreds or thousands of resources.

To illustrate that results are statistically significant, in Fig. 6.10 we plot the revenue gain of **DecompProb** over **DecompStd** for each individual scenario instance, ordered increasingly. In order to make the plot more readable, we only do this for the case $m = 16$. This means that each line in the plot essentially shows the cumulative distribution functions of the values that make up the respective boxplots in Fig. 6.9. For each single scenario instance, the shaded area shows the 0.95% confidence interval for the sample mean over the 10000 simulation runs. One sees that the revenue gains are often statistically significant even on the level of individual scenario instances, and certainly so when taking the average over the 100 instances.

Quality of revenue predictions

Lastly, we evaluate the quality of the revenue predictions computed from the different methods. Clearly there are multiple different choices for actual revenue for each scenario instance, depending on which optimization method and which control scheme is used in the simulation. In our analysis we used the revenue achieved with the standard LP-DP decomposition method **DecompStd**, because it is the most widely used method in practice. However, the assessment of revenue predictions would not change significantly if one were to use the revenue achieved with the probabilistic decomposition **DecompProb** as a baseline, because the actual improvement of **DecompProb** compared to **DecompStd** (see previous section) is, although significant, much smaller than the prediction errors of the different revenue prediction methods.

For every scenario instance, we have one revenue prediction for each of the following four methods. Firstly, we compute predicted revenue as described in Section 6.2.6 for both **DecompStd** and **DecompProb**. Secondly, we have the optimal objective function value of the deterministic **LP**, which is an upper bound on overall network revenue (see Section 3.4.4). Thirdly, each dis-

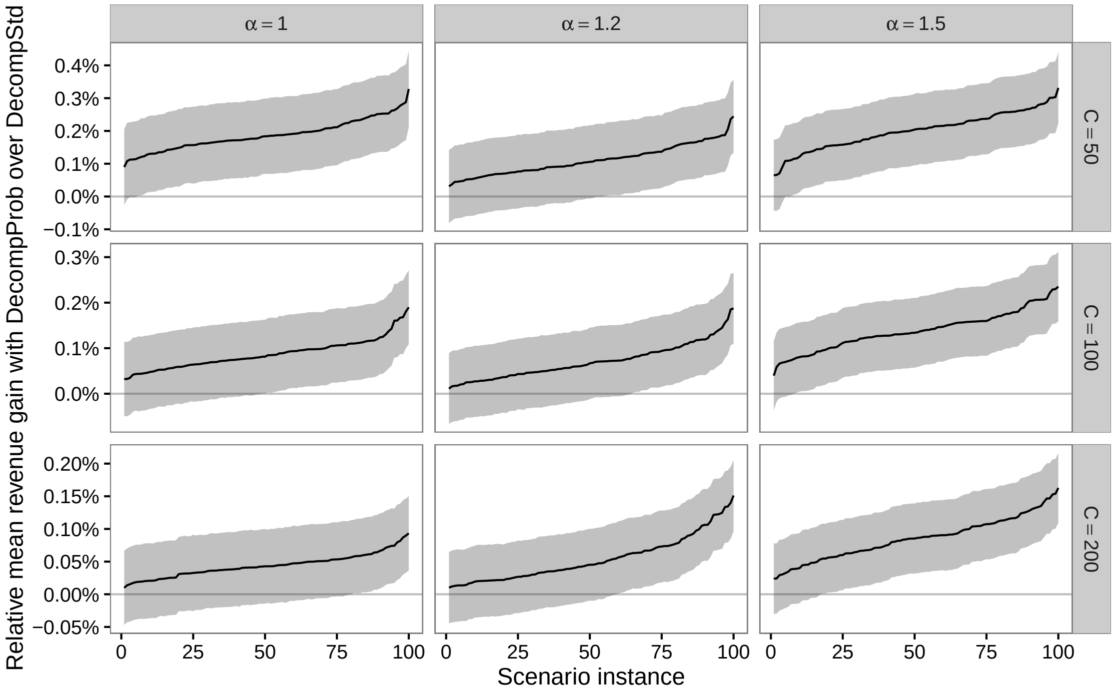


Figure 6.10: Distribution of revenue improvement of DecompProb compared to DecompStd over scenario instances ($R = 16$)

placement adjusted DP provides an upper bound on overall network revenue (see [126]), and we used the minimum of these bounds over all resources as the revenue prediction called **DP**.

Figure 6.11 shows the relative difference between various revenue predictions and actually achieved revenue in the simulation using **DecompStd** as the control mechanism. Each boxplot shows the distribution of the prediction error over the 100 different scenario instances, where for each single instance the revenue prediction of each method was compared to the average revenue over the 10000 simulation runs for the respective scenario instance. One sees clearly that both the **LP** and **DP** methods always overestimate expected revenue, and **LP** consistently shows a higher error than **DP**. This is expected, because both are proven to be upper bounds, with the bound from **DP** proven to be tighter ([126]). The prediction from **DecompStd** tends to underestimate expected revenue significantly. In many cases the median deviation is smaller than for the **LP** and **DP** methods, but the variance of the prediction error across the different scenario instances seems to be much higher. **DecompProb** provides the most accurate revenue predictions for all 27 scenario types with very low variance.

For all methods, predictions become more accurate with increasing capacity, which is explained by the fact that the variance of demand relative to capacity decreases, making the system more predictable. With increasing number of resources in the network, prediction quality remains largely unchanged for all methods except for **DP**, where the prediction error increases. This is due to the fact that the **DP** bound is obtained from a displacement adjusted DP for one resource, where this one resource is treated dynamically, while the rest of the network is treated statically. With increasing number of resources, the single resource that is treated more accurately becomes less relevant and the solution approaches that from the **LP** method.

To illustrate that results are statistically significant, Fig. 6.12 again shows the mean relative revenue prediction error of **DecompStd** compared to the actually observed revenue (using **DecompStd** as a control mechanism) for each individual scenario instance, ordered increasingly. Again, each line in the plot essentially represents the cumulative distribution function of the values that make up each of the boxplots for the method **DecompStd** in Fig. 6.11. Again for each single scenario instance, a shaded area shows the 0.95% confidence interval for the sample mean. In the plot this confidence interval is barely even visible, which shows that in all cases variance

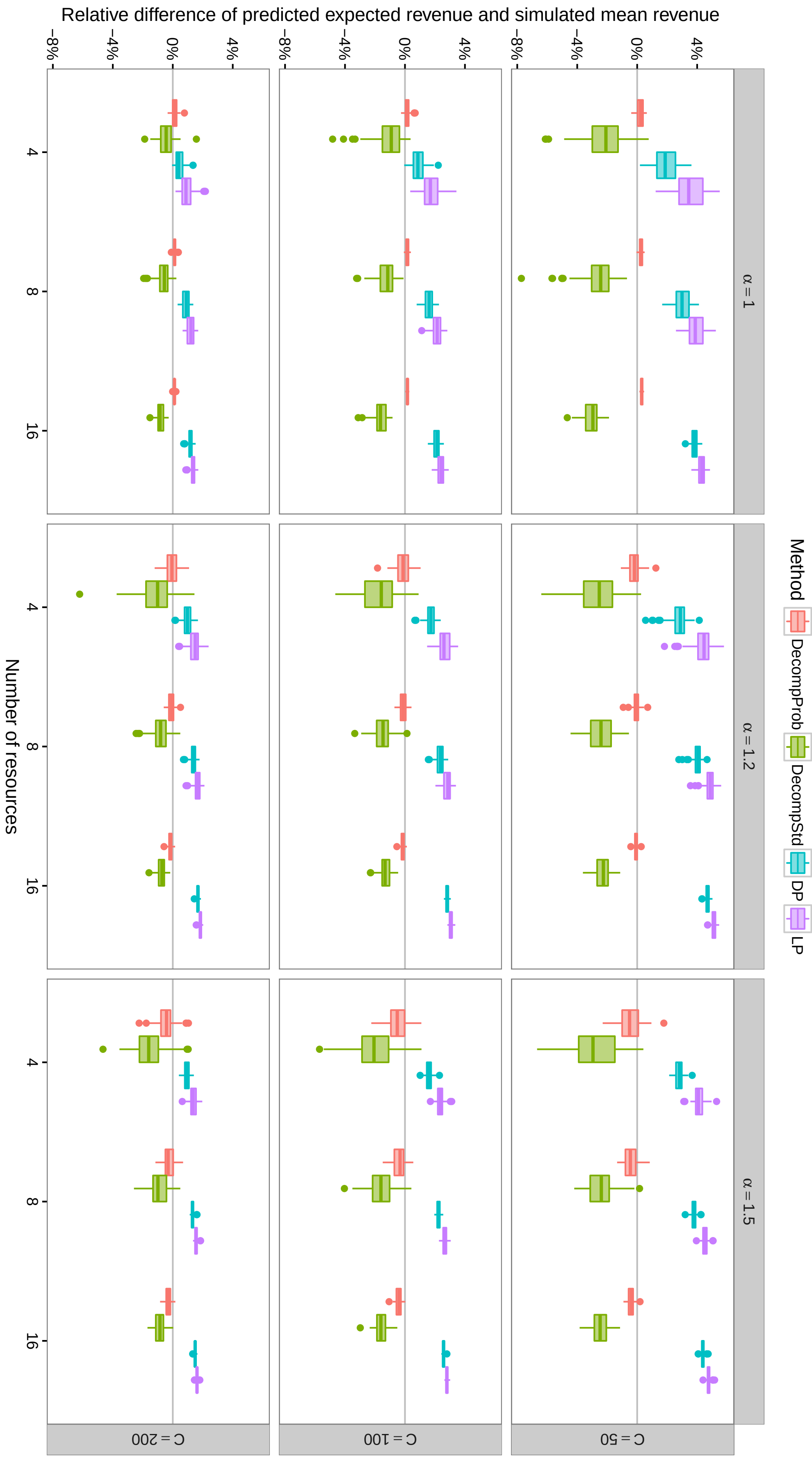


Figure 6.11: Revenue predictions by method vs actual revenue with DecompStd

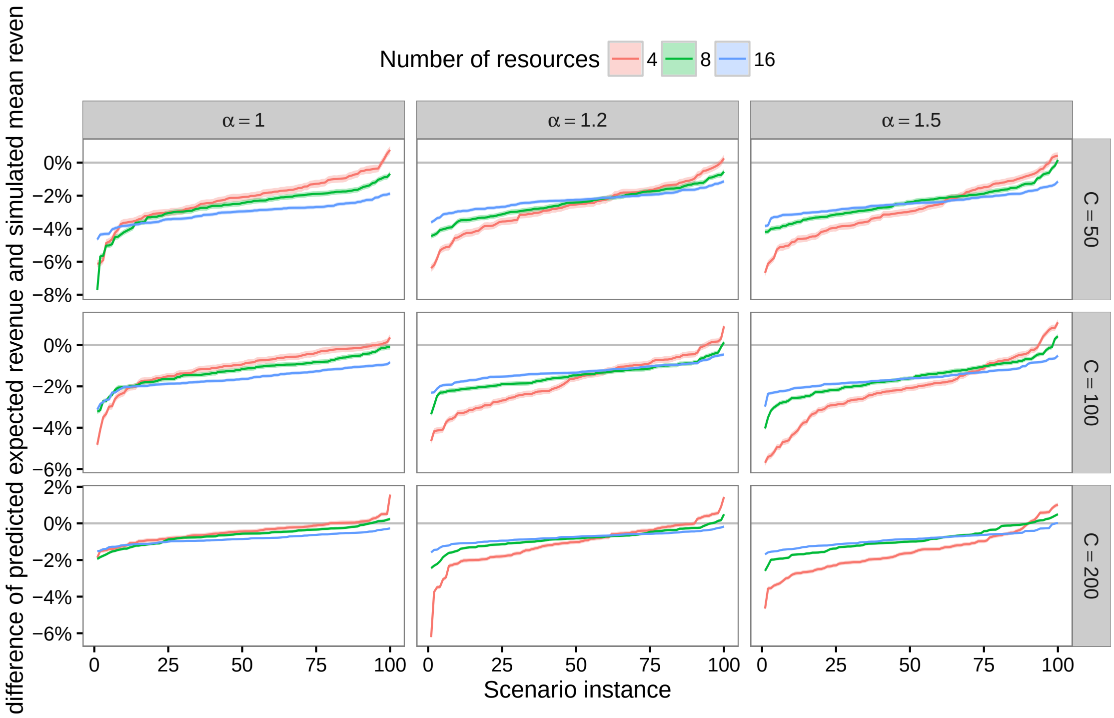


Figure 6.12: Distribution of revenue prediction error from DecompStd over scenario instances

between different scenario instances is much larger than the error of the sample mean. This means that increasing the number of simulation would not alter the findings.

All results are summarized Table 6.1, where the numbers in parentheses represent the 95% confidence intervals for the respective sample means, taking into account both the variance across the 100 scenario instances and the variance across the 10000 simulation run for each scenario instance.

			Revenue gain	Revenue prediction error			
m	C	α	DecompProb vs. DecompStd	LP	DP	DecompStd	DecompProb
4	50	1.0	0.151% (±0.039%)	3.539% (±0.200%)	1.946% (±0.160%)	−2.181% (±0.262%)	0.208% (±0.052%)
4	50	1.2	0.156% (±0.036%)	4.319% (±0.141%)	2.776% (±0.116%)	−2.630% (±0.281%)	−0.195% (±0.090%)
4	50	1.5	0.217% (±0.035%)	4.132% (±0.079%)	2.793% (±0.062%)	−2.804% (±0.313%)	−0.518% (±0.152%)
4	100	1.0	0.062% (±0.026%)	1.784% (±0.128%)	0.906% (±0.100%)	−1.060% (±0.184%)	0.156% (±0.041%)
4	100	1.2	0.118% (±0.025%)	2.602% (±0.091%)	1.696% (±0.067%)	−1.761% (±0.236%)	−0.143% (±0.102%)
4	100	1.5	0.155% (±0.027%)	2.352% (±0.055%)	1.614% (±0.047%)	−2.092% (±0.293%)	−0.517% (±0.154%)
4	200	1.0	0.024% (±0.017%)	0.941% (±0.085%)	0.466% (±0.067%)	−0.465% (±0.106%)	0.125% (±0.041%)
4	200	1.2	0.078% (±0.020%)	1.435% (±0.081%)	0.944% (±0.068%)	−1.073% (±0.215%)	−0.062% (±0.094%)
4	200	1.5	0.109% (±0.021%)	1.373% (±0.050%)	0.953% (±0.040%)	−1.539% (±0.217%)	−0.449% (±0.114%)
8	50	1.0	0.171% (±0.029%)	3.885% (±0.112%)	2.987% (±0.103%)	−2.633% (±0.224%)	0.257% (±0.031%)
8	50	1.2	0.137% (±0.025%)	4.843% (±0.066%)	4.021% (±0.061%)	−2.438% (±0.178%)	−0.065% (±0.049%)
8	50	1.5	0.199% (±0.026%)	4.536% (±0.044%)	3.784% (±0.039%)	−2.429% (±0.185%)	−0.430% (±0.088%)
8	100	1.0	0.073% (±0.018%)	2.113% (±0.071%)	1.577% (±0.065%)	−1.239% (±0.126%)	0.154% (±0.023%)
8	100	1.2	0.099% (±0.018%)	2.813% (±0.058%)	2.334% (±0.054%)	−1.484% (±0.116%)	−0.126% (±0.049%)
8	100	1.5	0.140% (±0.019%)	2.656% (±0.032%)	2.239% (±0.028%)	−1.594% (±0.158%)	−0.376% (±0.085%)
8	200	1.0	0.038% (±0.012%)	1.191% (±0.050%)	0.875% (±0.047%)	−0.655% (±0.094%)	0.123% (±0.020%)
8	200	1.2	0.061% (±0.013%)	1.648% (±0.044%)	1.367% (±0.040%)	−0.835% (±0.103%)	−0.101% (±0.047%)
8	200	1.5	0.099% (±0.013%)	1.556% (±0.020%)	1.320% (±0.018%)	−0.908% (±0.131%)	−0.269% (±0.080%)
16	50	1.0	0.187% (±0.018%)	4.289% (±0.053%)	3.829% (±0.050%)	−3.050% (±0.116%)	0.305% (±0.018%)
16	50	1.2	0.111% (±0.018%)	5.116% (±0.031%)	4.698% (±0.028%)	−2.273% (±0.104%)	−0.071% (±0.027%)
16	50	1.5	0.200% (±0.019%)	4.761% (±0.028%)	4.378% (±0.026%)	−2.462% (±0.107%)	−0.412% (±0.045%)
16	100	1.0	0.087% (±0.013%)	2.396% (±0.049%)	2.114% (±0.047%)	−1.610% (±0.088%)	0.166% (±0.014%)
16	100	1.2	0.071% (±0.013%)	3.071% (±0.025%)	2.820% (±0.022%)	−1.297% (±0.075%)	−0.148% (±0.027%)
16	100	1.5	0.138% (±0.013%)	2.795% (±0.019%)	2.574% (±0.018%)	−1.582% (±0.088%)	−0.414% (±0.041%)
16	200	1.0	0.044% (±0.009%)	1.344% (±0.031%)	1.173% (±0.030%)	−0.837% (±0.054%)	0.109% (±0.010%)
16	200	1.2	0.054% (±0.010%)	1.832% (±0.020%)	1.677% (±0.018%)	−0.756% (±0.055%)	−0.154% (±0.027%)
16	200	1.5	0.085% (±0.010%)	1.615% (±0.014%)	1.490% (±0.014%)	−0.873% (±0.072%)	−0.289% (±0.041%)

Table 6.1: Comparison of simulation results and revenue prediction by scenario and method

Chapter 7

Numerical treatment of customer choice models

In this section we will analyze how the aggregation problem for a fairly general class of customer choice models can be solved efficiently and deterministically using numerical methods for the computation of higher dimensional integrals. Furthermore, we will prove a theoretical result that allows to compute derivatives of booking probabilities w.r.t. input parameters such as product characteristics using the same methods.

Following the notation of Chapter 2, let always $\mathbf{P}$ be a product space and $\mathbf{T} = (\mathbf{X}, u, \lambda)$ a customer type for $\mathbf{P}$ with a deterministic utility function u and a deterministic decision rule. We assume that the distribution of $\mathbf{X}$ is described by the generalized density function $f: \mathbb{R}^n \to \mathbb{R}$. Let $\mathcal{S} = \{\mathfrak{p}_1, \dots, \mathfrak{p}_M\} \subset \mathbf{P}$ be a finite subset of the product space and denote by $\mathfrak{p}_0$ the outside good. We will again denote the choice of a customer $\mathbf{x}$ given the set of alternatives $\mathcal{S}$ by $\mathfrak{p}^*(\mathbf{x}, \mathcal{S})$.

Our goal is to compute the booking probabilities $d_{\mathfrak{p}}(\mathcal{S}) = \mathrm{P}[\mathfrak{p}^*(\mathbf{X}, \mathcal{S}) = \mathfrak{p}]$. If the offer set $\mathcal{S}$ is $\mathbf{T}$-independent, we have

$$d_{\mathfrak{p}}(\mathcal{S}) = \int_{\mathcal{X}_{\mathfrak{p}}(\mathcal{S})} f(\mathbf{x}) \, \mathrm{d}\mathbf{x}. \tag{7.1}$$

where $\mathcal{X}_{\mathfrak{p}}(\mathcal{S})$ is the set of customers preferring product $\mathfrak{p}$ over all other products, including the no-purchase option.

For each product $k = 1, \dots, M$ denote by u_k the utility function for product k:

$$u_k: \mathbf{C} \to \mathbb{R} \tag{7.2a}$$
$$\mathbf{x} \mapsto u_k(\mathbf{x}) := u(\mathbf{x}, \mathfrak{p}_k) \tag{7.2b}$$

Analogously utility for the outside product, which has zero utility by definition, is denoted by $u_0 \equiv 0$.

The customer set $\mathcal{X}_{\mathfrak{p}_k}$ for product k is defined by

$$\mathcal{X}_{\mathfrak{p}_k} = \{\mathbf{x} \in \mathbf{C} \mid \forall k' = 0, \dots, M : u_k(\mathbf{x}) - u_{k'}(\mathbf{x}) \geq 0\}. \tag{7.3}$$

In Eq. (7.3) we include the irrelevant inequality for $k' = k$ in order to simplify notation. We will always do so in the following sections as well.

7.1 Derivatives w.r.t. product variables

In this section we will prove a statement that allows us to compute derivatives of the choice probabilities $d_{\mathfrak{p}}(\mathcal{S})$ using Eq. (7.1).

Theorem 7.1.1
Let $X \subset \mathbb{R}^n$ be an open set. Let

$$f: \mathbb{R}^n \to \mathbb{R} \tag{7.4a}$$
$$x \mapsto f(x) \tag{7.4b}$$

be Lebesgue integrable with support $A = \operatorname{supp}(f) \subset X$*. Let* $V \subset \mathbb{R}$ *be an open interval and* $p_0 \in V$*. Let*

$$g\colon \mathbb{R}^n \times V \to \mathbb{R} \tag{7.5a}$$
$$(x,p) \mapsto g(x,p). \tag{7.5b}$$

For every $p \in V$ *let*

$$X_p \coloneqq \{x \in \mathbb{R}^n \mid g(x,p) \geq 0\} \tag{7.6a}$$
$$Y_p \coloneqq \{x \in \mathbb{R}^n \mid g(x,p) = 0\}, \tag{7.6b}$$

satisfying

$$X_{p_0} \subset X. \tag{7.7}$$

Assume that for every $x \in Y_{p_0}$: g *is continuously differentiable w.r.t.* x *and* p *at* (x,p_0) *with partial derivatives* $\frac{\partial g}{\partial x}$ *and* $\frac{\partial g}{\partial p}$ *and*

$$\forall x \in Y_{p_0} : \frac{\partial g}{\partial x}(x,p_0) \neq 0. \tag{7.8}$$

Furthermore assume that f *is continuous in* x *for almost every* $x \in Y_{p_0}$*. We define the function* $F\colon V \to \mathbb{R}$ *by*

$$F(p) \coloneqq \int_{X_p} f(x)\,\mathrm{d}x. \tag{7.9}$$

Then

$$F_p(p_0) = \frac{\partial F}{\partial p}(p_0) = \int_{Y_{p_0}} \frac{\frac{\partial g}{\partial p}(y,p_0)}{\left\|\frac{\partial g}{\partial x}(y,p_0)\right\|} f(y)\,\mathrm{d}y. \tag{7.10}$$

Proof In order to compute $F_p(p_0)$ we can replace V by an arbitrarily small open neighborhood of p_0. Since g is continuously differentiable w.r.t. p, the arguments below remain valid for small variations of p around p_0. We will prove the statement in two steps:

(1) **Reduce the problem to the case where X is an arbitrarily small neighborhood of a point $x_0 \in Y_p$.**

(2) **Derive a parametrization of X in terms of p and an $n-1$-dimensional variable** and prove the statement by transforming the integral to the new coordinates.

Step 1
Let

$$\{\mu_U\colon U \to \mathbb{R} \mid U \in \mathcal{U}\} \tag{7.11}$$

be a partition of unity subordinate to the open cover $\mathcal{U} = \{U_1, U_2, \ldots,\}$ of $\mathbb{R}^n$. For each open subset $U \in \mathcal{U}$ let $U_p \coloneqq X_p \cap U$ and

$$F_U(p) \coloneqq \int_{U_p} \mu_U(x) f(x)\,\mathrm{d}x. \tag{7.12}$$

Then

$$F(p) = \sum_{U \in \mathcal{U}} F_U(p) \tag{7.13a}$$

and

$$\frac{\partial F}{\partial p}(p) = \sum_{U \in \mathcal{U}} \frac{\partial F_U}{\partial p}(p). \tag{7.13b}$$

Therefore it is sufficient to compute the value of $\frac{\partial F_U}{\partial p}(p_0)$ for every such U.

Now consider a fixed open set $U \in \mathcal{U}$ and assume w.l.o.g. that U is connected. If $\varnothing = U \cap Y_p$ then either $U \subset X_p \Rightarrow U_p = U \cap X_p = U$ and

$$F_U(p) = \int_U \mu_U(x) f(x)\, \mathrm{d}x \tag{7.14}$$

is independent of p, or $U_p = \varnothing$ and $F_U(p) = 0$. In both cases $\frac{\partial F_U}{\partial p}(p) = 0$.

By definition the partition of unity μ satisfies $\operatorname{supp}(\mu_U) \subset U$ and thus $A' = \operatorname{supp}(\mu_U f) \subset U$. Therefore there exists a continuously differentiable

$$g' : \mathbb{R}^n \times V \to \mathbb{R}, \tag{7.15a}$$

such that

$$g'\big|_{A' \times V} \equiv g\big|_{A' \times V} \tag{7.15b}$$

$$X'_{p_0} \coloneqq \{x \in \mathbb{R}^n \mid g'(x, p_0) \geq 0\} \subset U \tag{7.15c}$$

$$x \in Y'_{p_0} \coloneqq \{x \in \mathbb{R}^n \mid g'(x, p_0) = 0\} \Rightarrow \frac{\partial g'}{\partial x}(x, p_0) \neq 0. \tag{7.15d}$$

With Eq. (7.15b) we have

$$F_U(p) = F'(p) \coloneqq \int_{X'_p} \mu_U(x) f(x)\, \mathrm{d}x \tag{7.16a}$$

$$\int_{Y'_{p_0}} \frac{g'_p(x, p_0)}{\|g'_x(x, p_0)\|} \mu_U(x) f(x)\, \mathrm{d}x = \int_{Y_{p_0}} \frac{g_p(x, p_0)}{\|g_x(x, p_0)\|} \mu_U(x) f(x)\, \mathrm{d}x. \tag{7.16b}$$

Therefore U, g' and $\mu_U f$ satisfy the original assumptions about X, g and f (Eqs. (7.7) and (7.8)), which means that in the following we can always replace X by U. Whenever a statement only holds on a small open neighborhood of a point, we can simply refine $\mathcal{U}$ and μ. We will use this fact multiple times in the remainder of this proof by assuming w.l.o.g that X is small enough whenever necessary. As shown above, we can restrict ourselves to the case where X is a small open neighborhood of a point $x \in Y_p$.

Step 2

In step 1 we showed that $F_p(p_0) = 0$ if $Y_{p_0} = \varnothing$ and that it is sufficient to consider arbitrarily small neighborhoods of points $x_0 \in Y_{p_0}$ otherwise. Let $h \coloneqq g(\cdot, p_0) : X \to \mathbb{R}$ and $x_0 \in Y_{p_0}$, i.e. with

$$g(x_0, p_0) = h(x_0) = 0. \tag{7.17}$$

Since h is continuously differentiable in x, and $\frac{\partial h}{\partial x}$ is nonzero at x_0 (Eq. (7.8)), 0 is a regular value of h. By virtue of the submersion theorem $Y_{p_0} = h^{-1}(0)$, is a smooth $n-1$ dimensional sub-manifold of X. Let

$$\iota : Y_{p_0} \hookrightarrow X \tag{7.18}$$

be the inclusion of Y_{p_0} into X and $y_0 = \iota^{-1}(x_0)$ be x_0 interpreted as an element of Y_{p_0}.

In the following we will omit ι whenever possible and instead simply regard an element y of Y_{p_0} as an element of X. In particular we will write $g(y, p)$ instead of $g(\iota(y), p)$. By definition of Y_{p_0} we have $h \circ \iota \equiv 0$ and thus

$$\frac{\partial h}{\partial x}(y) \frac{\partial \iota}{\partial y}(y) \equiv 0. \tag{7.19}$$

We will now derive a local parametrization of X in terms of Y_{p_0} and an open interval. Let $I = (t^{\text{start}}, t^{\text{end}}) \subset \mathbb{R}$ be an open interval and $t_0 \in I$. For a fixed $y \in Y_{p_0}$ let $\hat{x}(t; y)$ be the solution of the IVP

$$\dot{\hat{x}}(t) = \frac{\partial h}{\partial x}(\hat{x}(t))^\top \tag{7.20a}$$

$$\hat{x}(t_0) = \iota(y). \tag{7.20b}$$

$$
\begin{tikzcd}
Y_{p_0} \arrow[d, hook, "y\mapsto(t_0,y)"'] \arrow[dr, "\iota"] & & \\
I \times Y_{p_0} \arrow[r, "\Phi"] \arrow[d, hook, "(t,y)\mapsto(t,y,p_0)"'] & X \arrow[d, hook, "x\mapsto(x,p_0)"'] \arrow[dr, "h"] & \\
I \times Y_{p_0} \times V \arrow[r, "\Phi\times\mathbb{I}"] \arrow[rr, bend right, "G"'] & X \times V \arrow[r, "g"] & \mathbb{R}
\end{tikzcd}
$$

Figure 7.1: Maps constructed in step 2 of proof of Theorem 7.1.1

This yields a map

$$\Phi: I \times Y_{p_0} \to X \tag{7.21a}$$
$$(t, y) \mapsto \Phi(t, y) \coloneqq \hat{x}(t; y) \tag{7.21b}$$

with

$$\Phi(t_0, y) = y. \tag{7.22a}$$

Denote the differential of Φ by

$$D(\Phi)(t,y) = \frac{\partial \Phi}{\partial(t,y)} = \begin{pmatrix} \frac{\partial \Phi}{\partial t}(t,y) & \frac{\partial \Phi}{\partial y}(t,y) \end{pmatrix} \tag{7.23}$$

where with Eq. (7.20a)

$$\frac{\partial \Phi}{\partial t}(t,y) = \frac{\partial h}{\partial x}(\Phi(t,y))^\top. \tag{7.24}$$

For every $y \in Y_{p_0}$ we have

$$\frac{\partial \Phi}{\partial t}(t_0,y) = \frac{\partial h}{\partial x}(y)^\top \tag{7.25a}$$
$$\frac{\partial \Phi}{\partial y}(t_0,y) = \frac{\partial \iota}{\partial y}(y). \tag{7.25b}$$

Since ι is injective, $\frac{\partial \Phi}{\partial y}(t_0,y)$ has rank $n-1$. Equation (7.8) implies that $\frac{\partial \Phi}{\partial t}(t_0,0) \neq 0$. Moreover, with Eq. (7.19) we see that

$$\frac{\partial \Phi}{\partial t}(t_0,y)^\top \frac{\partial \Phi}{\partial y}(t_0,y) = 0 \Rightarrow \frac{\partial \Phi}{\partial t}(t_0,y) \perp \frac{\partial \Phi}{\partial y}(t_0,y). \tag{7.26}$$

Thus the differential of Φ at (t_0, y_0)

$$D(\Phi)(t_0,y_0) = \begin{pmatrix} \Phi_t(t_0,y_0) & \Phi_y(t_0,y_0) \end{pmatrix} \tag{7.27}$$

has full rank. By virtue of the inverse function theorem Φ is a diffeomorphism in a neighborhood of (t_0, y_0).

For every $y \in Y_{p_0}$ choose an orthogonal basis $\mathcal{V} = \{v_1, \dots, v_{n-1}\}$ of the tangent space $T_y Y_{p_0}$. Then with Eq. (7.26) $\mathcal{V}' = \mathcal{V} \cup \left\{ \frac{\frac{\partial h}{\partial x}(y)}{\left\| \frac{\partial h}{\partial y}(y) \right\|} \right\}$ is an orthogonal basis of the tangent space $T_y X$. Again using Eq. (7.26) we see that the differential $D(\Phi)(t_0, y)$ can be written as

$$D(\Phi)(t_0,y) = W \begin{pmatrix} \left\| \frac{\partial h}{\partial x}(y) \right\| & 0 \\ 0 & \mathbb{I}_{n-1} \end{pmatrix} \tag{7.28}$$

with an orthogonal matrix W. Therefore we have

$$\det\left(D(\Phi)\right)(t_0, y) = \underbrace{\det(W)}_{=1}\left\|\frac{\partial h}{\partial x}(y)\right\|. \tag{7.29a}$$

$$= \left\|\frac{\partial h}{\partial x}(y)\right\| \tag{7.29b}$$

Let

$$G: I \times Y_{p_0} \times V \to \mathbb{R} \tag{7.30a}$$

$$(t, y, p) \mapsto G(t, y, p) = g(\Phi(t, y), p) \tag{7.30b}$$

and consider the equation

$$G(t, y, p) = 0. \tag{7.31}$$

For every $y \in Y_{p_0}$ we have

$$G(t_0, y, p_0) = g(\Phi(t_0, y), p_0) = g(y, p_0) = 0 \tag{7.32}$$

and, with Eq. (7.8),

$$G_t(t_0, y, p_0) = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=t_0} g(\Phi(t, y), p_0) \tag{7.33a}$$

$$= \frac{\partial h}{\partial x}(\Phi(t_0, y))\frac{\partial \Phi}{\partial t}(t_0, y) \tag{7.33b}$$

$$= \frac{\partial h}{\partial x}(y)\frac{\partial h}{\partial x}(y)^\top \tag{7.33c}$$

$$= \left\|\frac{\partial h}{\partial x}(y)\right\|^2 > 0. \tag{7.33d}$$

Thus, according to the implicit function theorem, there is a map

$$s: Y_{p_0} \times V \to I \tag{7.34a}$$

with

$$G(s(y, p), y, p) = 0 \tag{7.34b}$$

for every (y, p) in a small neighborhood of (y_0, p_0). With Eq. (7.32) we have

$$s(y, p_0) = t_0 \tag{7.35}$$

for every $y \in Y_{p_0}$. Moreover, the derivative $\frac{\partial s}{\partial p}(y, p_0)$ w.r.t. p satisfies

$$\frac{\partial s}{\partial p}(y, p_0) = -\frac{\frac{\partial G}{\partial p}(t_0, y, p_0)}{\frac{\partial G}{\partial t}(t_0, y, p_0)} \tag{7.36a}$$

$$= -\frac{\frac{\partial g}{\partial p}(\Phi(t_0, y), p_0)}{\left\|\frac{\partial g}{\partial x}(\Phi(t_0, y), p_0)\right\|^2} \tag{7.36b}$$

$$= -\frac{\frac{\partial h}{\partial p}(y)}{\left\|\frac{\partial h}{\partial x}(y)\right\|^2}. \tag{7.36c}$$

Now for $x \in X$ let $(t, y) = \Phi^{-1}(x)$. If $X \times V$ is a sufficiently small neighborhood of (x_0, p_0) we have

$$x \in X_p \Leftrightarrow g(x, p) \geq 0 \tag{7.37a}$$

$$\Leftrightarrow g(\Phi(t, y), p) \geq 0 \tag{7.37b}$$

$$\Leftrightarrow G(t, y, p) \geq 0 \tag{7.37c}$$

$$\Leftrightarrow t \geq s(y, p) \tag{7.37d}$$

where the last equivalence follows from Eqs. (7.33d) and (7.34b), but only holds for x close to x_0 or, equivalently, (t, y) close to (t_0, y_0). However, this can always be achieved (Step 1). In other words,

$$X_p = \Phi\left(\{(t, y) \in I \times Y_{p_0} \mid t \geq s(y, p)\}\right). \tag{7.38}$$

Using substitution of variables we can transform Eq. (7.9) as follows:

$$\begin{aligned} F(p) &= \int_{X_p} f(x)\,\mathrm{d}x && (7.39a)\\ &= \int_{\Phi(\{(t,y)\in I\times Y_{p_0} \mid t\geq s(y,p)\})} f(x)\,\mathrm{d}x && (7.39b)\\ &= \int_{\{(t,y)\in I\times Y_{p_0} \mid t\geq s(y,p)\}} f\left(\Phi(t,y)\right)\left|\det(D(\Phi))(t,y)\right|\,\mathrm{d}t\,\mathrm{d}y && (7.39c)\\ &= \int_{s(y,p)}^{t^{\mathrm{end}}}\int_{Y_{p_0}} f\left(\Phi(t,y)\right)\left|\det(D(\Phi))(t,y)\right|\,\mathrm{d}y\,\mathrm{d}t. && (7.39d) \end{aligned}$$

In Eq. (7.39d) the only term that depends on p is the lower limit $s(y, p)$ of the inner integral. Because f is continuous at $x = \Phi(t_0, y)$ for almost every $y \in Y_{p_0}$, the value of the inner integral is continuous in p at $p = p_0$. By virtue of the fundamental theorem of calculus we can therefore have

$$F_p(p) = -\int_{Y_{p_0}} \frac{\partial s}{\partial p}(y,p) f\left(\Phi(s(y,p),y)\right)\left|\det(D(\Phi))(s(y,p),y)\right|\,\mathrm{d}y. \tag{7.40a}$$

Using the fact that $s(y, p_0) = t_0$ for every y (Eq. (7.35)) we have

$$\begin{aligned} F_p(p_0) &= -\int_{Y_{p_0}} \frac{\partial s}{\partial p}(y,p_0) f\left(\Phi(t_0,y)\right)\left|\det(D(\Phi))(t_0,y)\right|\,\mathrm{d}y && (7.41a)\\ &= \int_{Y_{p_0}} \frac{\frac{\partial h}{\partial p}(y)}{\left\|\frac{\partial h}{\partial x}(y)\right\|^2} f(y)\left|\det(D(\Phi))(t_0,y)\right|\,\mathrm{d}y && (7.41b)\\ &= \int_{Y_{p_0}} \frac{\frac{\partial h}{\partial p}(y)}{\left\|h_x(y)\right\|^2} f(y)\left\|\frac{\partial h}{\partial x}(y)\right\|\,\mathrm{d}y && (7.41c)\\ &= \int_{Y_{p_0}} \frac{\frac{\partial h}{\partial p}(y)}{\left\|\frac{\partial h}{\partial x}(y)\right\|} f(y)\,\mathrm{d}y, && (7.41d) \end{aligned}$$

where Eq. (7.41b) follows from Eqs. (7.36c) and (7.22a) and Eq. (7.41c) follows from Eq. (7.29b). This completes the proof. □

The theorem generalizes to multiple g as follows:

Corollary 7.1.2
Let X, V, $p_0 \in V$ and f be as in Theorem 7.1.1. Let

$$\begin{aligned} g\colon \mathbb{R}^n \times V &\to \mathbb{R}^m && (7.42a)\\ (x,p) &\mapsto \begin{pmatrix} g_1(x,p)\\ \vdots \\ g_m(x,p)\end{pmatrix}. && (7.42b) \end{aligned}$$

For every $p \in V$ *let*

$$X_p^{(i)} \coloneqq \{x \in \mathbb{R}^n \mid g_i(x,p) \geq 0\} \tag{7.43a}$$

$$X_p \coloneqq \{x \in \mathbb{R}^n \mid \forall i = 1,\dots,m : g_i(x,p) \geq 0\} = \bigcap_{i=1}^{m} X_p^{(i)} \tag{7.43b}$$

$$Y_p^{(i)} \coloneqq \{x \in \mathbb{R}^n \mid g_i(x,p) = 0\} \cap X_p \tag{7.43c}$$

$$Y_p \coloneqq \{x \in \mathbb{R}^n \mid \exists i = 1,\dots,m : g_i(x,p) = 0\} \cap X_p = \bigcup_{i=1}^{m} Y_p^{(i)} \tag{7.43d}$$

$$Y'_p \coloneqq \{x \in \mathbb{R}^n \mid \exists i = 1,\dots,m : g_i(x,p) = 0 \wedge \forall i' \neq i : g_i(x,p) > 0\} \tag{7.43e}$$

$$= Y_p \setminus \bigcup_{i=1}^{m} \bigcup_{i'=i+1}^{m} Y_p^{(i)} \cap Y_p^{(i')}, \tag{7.43f}$$

satisfying $Y'_{p_0} \subset Y_{p_0} \subset X_{p_0} \subset X$*. Assume that for every* $x \in Y_{p_0} : g$ *is continuously differentiable w.r.t.* x *and* p *at* (x,p_0)*. For every* $x \in Y_{p_0}$ *let* $I(x) = \{i \in 1,\dots,m \mid g_i(x,p_0) = 0\}$ *be the set of active inequalities and let* $i(x)$ *be the index of the unique active inequality for every* $x \in Y'_{p_0}$*. Assume that for every* $x \in Y_{p_0}$ *the gradients* $\left\{\frac{\partial g_i}{\partial x}(x,p_0) \mid i \in I(x)\right\}$ *of the active components are linearly independent. Furthermore assume that* f *is continuous in* x *for almost every* $x \in Y_{p_0}$*. We define the function* $F\colon V \to \mathbb{R}$ *by*

$$F(p) \coloneqq \int_{X_p} f(x)\,\mathrm{d}x. \tag{7.44}$$

Then

$$F_p(p_0) = \frac{\partial F}{\partial p}(p_0) = \int_{Y'_{p_0}} \frac{\partial \frac{g_{i(x)}}{\partial p}(y,p_0)}{\left\|\frac{g_{i(x)}}{\partial x}(y,p_0)\right\|} f(y)\,\mathrm{d}y. \tag{7.45}$$

Proof We will prove the statement for a simplified case where only one of the level set functions depends on the parameter, and then deduce the general case in the second step.

Step 1
First, consider the case that only g_1 depends on p and all other components of g are independent of p. We can reduce the problem to the case of Theorem 7.1.1 by moving all but the first component of g into the integrand. To this end, we replace f by

$$f^{(1)}(x) = f(x)\prod_{i=2}^{m} H(g_i(x)), \tag{7.46}$$

where $H(\cdot)$ is the Heaviside function, which is equal to zero if the argument is negative and equal to one otherwise. For every $i \neq 1$ the set $Y_p^{(i)}$ is independent of p and we will simply write $Y^{(i)}$. Because the Jacobian of the active components has full rank, the submersion theorem implies that the set

$$\bigcup_{i=2}^{m} Y_{p_0}^{(1)} \cap Y^{(i)}, \tag{7.47}$$

of points where both g_1 and another component of g are equal to zero is a zero set in Y_{p_0}. Therefore, $f^{(1)}(x)$ is continuous for almost every $x \in Y_{p_0}^{(1)}$. By definition of $f^{(1)}$ we have

$$\int_{X_p} f(x)\,\mathrm{d}x = \int_{X_p^{(1)}} f^{(1)}(x)\,\mathrm{d}x. \tag{7.48}$$

Therefore

$$\frac{\partial F}{\partial p}(p_0) = \int_{Y_{p_0}^{(1)}} \frac{\frac{\partial g_1}{\partial p}(y, p_0)}{\left\|\frac{\partial g_1}{\partial x}(y, p_0)\right\|} f^{(1)}(y)\, \mathrm{d}y \tag{7.49}$$

$$= \int_{Y_{p_0}^{(1)}} \frac{\frac{\partial g_{i(x)}}{\partial p}(y, p_0)}{\left\|\frac{\partial g_{i(x)}}{\partial x}(y, p_0)\right\|} f(y)\, \mathrm{d}y \tag{7.50}$$

$$= \int_{Y'_{p_0}} \frac{\frac{\partial g_{i(x)}}{\partial p}(y, p_0)}{\left\|\frac{\partial g_{i(x)}}{\partial x}(y, p_0)\right\|} f(y)\, \mathrm{d}y. \tag{7.51}$$

where the first equality follows from Theorem 7.1.1 with integrand $f^{(1)}$ and level set function g_1, the second equality holds because $f(x) = f^{(1)}(x)$ for almost every $x \in Y_{p_0}^{(1)}$, and the last equality holds because $\frac{\partial g_i}{\partial p} \equiv 0$ for all $i \neq 1$.

Step 2
The case where all g depend on p is a direct consequence. First, assume that each g_i has their own parameter $p^{(i)}$. Clearly, with the above, the gradient of F w.r.t. to the vector $p' = (p^{(1)}, \dots, p^{(m)})^\top$ is equal to

$$\frac{\partial F}{\partial p'}(p'_0) = \left(\int_{Y_{p_0}^{(1)}} \frac{\frac{\partial g_1}{\partial p^{(1)}}(y, p_0^{(1)})}{\left\|\frac{\partial g_1}{\partial x}(y, p_0^{(1)})\right\|} f(y)\, \mathrm{d}y, \quad \dots, \quad \int_{Y_{p_0}^{(m)}} \frac{\frac{\partial g_m}{\partial p^{(m)}}(y, p_0^{(m)})}{\left\|\frac{\partial g_m}{\partial x}(y, p_0^{(m)})\right\|} f(y)\, \mathrm{d}y \right). \tag{7.52}$$

By setting $p'(p) \coloneqq (p, \dots, p)^\top$ and applying the product rule we have

$$\frac{\mathrm{d}F}{\mathrm{d}p}(p_0) = \sum_{i=1}^{m} \int_{Y_{p_0}^{(i)}} \frac{\frac{\partial g_i}{\partial p}(y, p_0)}{\left\|\frac{\partial g_i}{\partial x}(y, p_0)\right\|} f(y)\, \mathrm{d}y \tag{7.53}$$

$$= \int_{Y'_{p_0}} \frac{\frac{\partial g_{i(x)}}{\partial p}(y, p_0)}{\left\|\frac{\partial g_{i(x)}}{\partial x}(y, p_0)\right\|} f(y)\, \mathrm{d}y, \tag{7.54}$$

where we use the fact that $Y_p \bigcup_{i=1}^m Y_p^{(i)}$ and that the pairwise intersections of the $Y_p^{(i)}$ is a zero-set in Y_p. □

7.2 Linear utility function

Under certain conditions the integral Eq. (7.1) can be computed efficiently. Most notably this is the case if the utility function is linear in customer variables. In other words if

$$u: \mathbf{C} \times \mathbf{P} \to \mathbb{R} \tag{7.55a}$$

$$(\mathbf{x}, \mathfrak{p}) \mapsto u(\mathbf{x}, \mathfrak{p}) = a(\mathfrak{p})^\top \mathbf{x} + b(\mathfrak{p}), \tag{7.55b}$$

where a and b are coefficient functions

$$a: \mathbf{P} \to \mathbb{R}^n \tag{7.56a}$$

$$b: \mathbf{P} \to \mathbb{R}. \tag{7.56b}$$

For every $k = 1, \dots, M$ let $a_k = a(\mathfrak{p}_k)$ and $b_k = b(\mathfrak{p}_k)$. Then, with Eq. (7.3), we see that the customer set

$$\mathcal{X}_{\mathfrak{p}_k} = \left\{ \mathbf{x} \in \mathbf{C} \mid \forall k' = 0, \dots, M : (a_k - a_{k'})^\top \mathbf{x} + b_k - b_{k'} \geq 0 \right\} \tag{7.57}$$

is an n-dimensional polyhedron with at most M facets. In general $\mathcal{X}_{\mathfrak{p}_k}$ need not be bounded. The assumption in Corollary 7.1.2 that the gradients of the active inequalities are linearly independent simply means that the polyhedron is non-degenerate.

7.3 Implementation

For a given value of the control vector u, let $\mathcal{P}(u) = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_M\}$ be the corresponding set of products. We compute the booking probabilities $d_k(\mathcal{P})$ for a fixed customer type $\mathbf{T}$ using the integral representation from Eq. (7.1):

$$d_{\mathfrak{p}}(\mathcal{P}) = \int_{\mathcal{X}_{\mathfrak{p}}(\mathcal{P})} f(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{7.58}$$

Since the utility function u is linear in the customer variables $\mathbf{x}$, the customer set $\mathcal{X}_{\mathfrak{p}}$ for product $\mathfrak{p} \in \mathcal{P}$ is a convex polyhedron (Eq. (7.57)) and can be written as

$$\mathcal{X}_{\mathfrak{p}} = \{\mathbf{x} \in \mathbf{C} \mid A(u)\mathbf{x} + b(u) \geq 0\}\,. \tag{7.59}$$

7.3.1 Unbounded customer sets

In general $\mathcal{X}_{\mathfrak{p}}$ is unbounded. In order to solve the integral Eq. (7.58) numerically it is necessary to reduce the integration region to a bounded subset of $\mathbb{R}^n$. When solving a one-dimensional improper integral

$$\int_0^\infty f(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{7.60}$$

with infinite bounds it is common to apply a change of variables $\mathbf{x} = \phi(\mathbf{y})$ that maps the unbounded integration region to a finite interval, for example $[0,\infty) = \phi([0,1])$, and compute the value of the definite integral

$$\int_0^1 f(\phi(\mathbf{y}))\phi'(\mathbf{y})\,\mathrm{d}\mathbf{y}. \tag{7.61}$$

The equivalent approach for higher-dimensional integrals is to apply a nonlinear transformation of variables Φ such that the pre-image $\Phi^{-1}(\mathcal{X}_{\mathfrak{p}}(\mathcal{P}))$ is a bounded subset of $\mathbb{R}^n$ and solve

$$d_{\mathfrak{p}}(\mathcal{P}) = \int_{\Phi^{-1}(\mathcal{X}_{\mathfrak{p}}(\mathcal{P}))} f(\Phi(\mathbf{y}))\,|\det(D(\Phi))(\mathbf{y})|\,\mathrm{d}\mathbf{y}. \tag{7.62}$$

However, since Φ is nonlinear, the transformed integration region $\Phi^{-1}(\mathcal{X}_{\mathfrak{p}}(\mathcal{P}))$ is no longer a polyhedron or even necessarily convex.

For this reason we avoid changing variables and instead approximate the improper integral with a proper one while controlling the hereby induced error. More precisely, we choose an error tolerance $\epsilon^{\mathrm{ign}} > 0$ that determines the fraction of customers that may be ignored during the computation of d. We then choose lower and upper bounds $\mathbf{x}_j^{\mathrm{lb}}$ and $\mathbf{x}_j^{\mathrm{ub}}$ for every customer attribute $j = 1, \ldots, n$ such that a random customer $\mathbf{X}$ is an element of the set

$$\hat{\mathcal{X}} = \left\{\mathbf{x} \in \mathbf{C} \mid \forall j = 1, \ldots, n : \mathbf{x}_j^{\mathrm{lb}} \leq \mathbf{x}_j \leq \mathbf{x}_j^{\mathrm{ub}}\right\}, \tag{7.63}$$

with a probability of at least $1 - \epsilon^{\mathrm{ign}}$, in other words

$$\int_{\hat{\mathcal{X}}} f(\mathbf{x})\,\mathrm{d}\mathbf{x} \geq 1 - \epsilon^{\mathrm{ign}}. \tag{7.64}$$

An estimate $\hat{d}_{\mathfrak{p}}(\mathcal{P})$ of $d_{\mathfrak{p}}(\mathcal{P})$ can be computed as

$$\hat{d}_{\mathfrak{p}}(\mathcal{P}) = \int_{\hat{\mathcal{X}}_{\mathfrak{p}}(\mathcal{P})} f(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{7.65}$$

where the integration region is the bounded set

$$\hat{\mathcal{X}}_{\mathfrak{p}}(\mathcal{P}) = \mathcal{X}_{\mathfrak{p}}(\mathcal{P}) \cap \hat{\mathcal{X}}. \tag{7.66}$$

The absolute error caused by this approximation is bounded by ϵ^{ign}:

$$\begin{aligned} d_{\mathfrak{p}}(\mathcal{P}) - \hat{d}_{\mathfrak{p}}(\mathcal{P}) &= \int_{\mathcal{X}_{\mathfrak{p}}(\mathcal{P}) \setminus \hat{\mathcal{X}}_{\mathfrak{p}}(\mathcal{P})} f(\mathbf{x})\,\mathrm{d}\mathbf{x} && (7.67a)\\ &\leq \int_{\mathbf{C} \setminus \hat{\mathcal{X}}} f(\mathbf{x})\,\mathrm{d}\mathbf{x} && (7.67b)\\ &\leq \epsilon^{\mathrm{ign}}. && (7.67c) \end{aligned}$$

Moreover, we have

$$\mathbf{x} \notin \hat{\mathcal{X}} \Leftrightarrow \exists j : \mathbf{x}_j > \mathbf{x}_j^{\mathrm{lb}} \vee \mathbf{x}_j < \mathbf{x}_j^{\mathrm{ub}} \tag{7.68a}$$

$$\Rightarrow \qquad \mathrm{P}\left[\mathbf{X} \notin \hat{\mathcal{X}}\right] \leq \sum_{j=1}^{n} \left(\mathrm{P}\left[\mathbf{X}_j > \mathbf{x}_j^{\mathrm{lb}}\right] + \mathrm{P}\left[\mathbf{X}_j < \mathbf{x}_j^{\mathrm{ub}}\right]\right). \tag{7.68b}$$

Therefore, if the quantile function of each individual random attribute $\mathbf{X}_j$ can be evaluated, Eq. (7.64) can be satisfied by simply choosing the $\frac{\epsilon^{\mathrm{ign}}}{2n}$- and $1 - \frac{\epsilon^{\mathrm{ign}}}{2n}$-quantiles of $\mathbf{X}_j$ as $\mathbf{x}_j^{\mathrm{lb}}$ and $\mathbf{x}_j^{\mathrm{ub}}$ respectively.

Note that the bounds $\mathbf{x}^{\mathrm{lb}}$ and $\mathbf{x}^{\mathrm{ub}}$ do not depend on $\mathcal{P}$, in other words they are independent of the current evaluation point but only depend on the (fixed) distribution of customers. Therefore for a fixed ϵ^{ign} they only need to be computed once for each scenario. This allows us to interpret the approach not as a method to approximate the booking probability d with an estimate $\hat{d}$, but to approximate the customer type $\mathbf{T}$ with alternative customer type $\hat{T}$ that deviates only slightly but has a bounded customer set and therefore is better suited for computational purposes.

7.3.2 Derivatives of booking probabilities

We will compute derivatives of $d_{\mathfrak{p}}(\mathcal{P})$ using the methods described in Section 7.1. With Eq. (7.58) the booking probabilities are equal to the integral over a parametrized level set. The system of linear inequalities

$$g(\mathbf{x}, u) \coloneqq A(u)\mathbf{x} + b(u) \geq 0 \tag{7.69}$$

that defines the level set (Eq. (7.59)) satisfies the continuity and differentiability conditions of Corollary 7.1.2. Let $\partial\mathcal{X}_{\mathfrak{p}}$ denote the boundary of the customer set $\mathcal{X}_{\mathfrak{p}}$. Denote again by $g_{\mathbf{x}}$ and g_u the partial derivatives of g w.r.t. $\mathbf{x}$ and u respectively. For every $\mathbf{x} \in \partial\mathcal{X}_{\mathfrak{p}}(\mathcal{P}(u))$ let $\hat{g}(\mathbf{x}, u)$ be a component of g that is active at $(\mathbf{x}, u)$, in other words $\hat{g}(\mathbf{x}, u) = 0$. This function is almost everywhere continuously differentiable on $\partial\mathcal{X}_{\mathfrak{p}}(\mathcal{P}(u))$ and can be written as

$$\hat{g}(\mathbf{x}, u) = \hat{A}(u)\mathbf{x} + \hat{b}(u) \tag{7.70}$$

with a vector-valued function $\hat{A}(u)$ and a scalar function $\hat{b}(u)$.

Corollary 7.1.2 states that the derivative of $d_{\mathfrak{p}}(\mathcal{P})$ is given by

$$\frac{\mathrm{d}}{\mathrm{d}u} d_{\mathfrak{p}}\left(\mathcal{P}(u)\right) = \int_{\partial\mathcal{X}_{\mathfrak{p}}} \frac{g_u(\mathbf{x}, u)}{\|g_{\mathbf{x}}(\mathbf{x}, u)\|} f(\mathbf{x})\,\mathrm{d}x \tag{7.71a}$$

$$= \int_{\partial\mathcal{X}_{\mathfrak{p}}} \frac{\hat{A}_u(u)\mathbf{x} + \hat{b}_u(u)}{\|\hat{A}(u)\|} f(\mathbf{x})\,\mathrm{d}x. \tag{7.71b}$$

Note that, due to the fact that the zero-sets of the individual components of g intersect normally, i.e. the respective rows of $A(\mathbf{x}, u)$ are linearly independent whenever multiple constraints are active, $\hat{g}$ is uniquely defined almost everywhere on $\partial\mathcal{X}_{\mathfrak{p}}(\mathcal{P}(u))$.

It is a well-known fact that the boundary of a polytope with dimension n is a finite collection of polytopes of dimension $n-1$. This allows us to rewrite Eq. (7.71b) as

$$\frac{\mathrm{d}}{\mathrm{d}u} d_{\mathfrak{p}}\left(\mathcal{P}(u)\right) = \sum_k \int_{Y_{\mathfrak{p}}^{(k)}} \frac{\hat{A}_u^{(k)}(u)\mathbf{x} + \hat{b}_u^{(k)}(u)}{\|\hat{A}^{(k)}(u)\|} f(\mathbf{x})\,\mathrm{d}x, \tag{7.72}$$

where each facet $Y^{(k)}$ is of the same general form as the customer set $\mathcal{X}_{\mathfrak{p}}$ (Eq. (7.59)). Once the set of facets $\left\{Y^{(k)}\right\}$ is known this allows us to compute the derivatives of booking probabilities using the same numerical methods that are used in the computation of the values of $d_{\mathfrak{p}}(\mathcal{P}(u))$. The number of facets is bounded by

$$\underbrace{M}_{\text{utility ineq.}} + \underbrace{2n.}_{\text{bounding box ineq. (Eq. (7.63))}} \tag{7.73}$$

The problem of identifying facets is equivalent to converting an outer representation of $\mathcal{X}_{\mathfrak{p}}$ to an inner representation, i.e. computing the set of vertices of the polytope. The worst-case run-time of all known vertex-enumeration algorithms to date is super-polynomial in input and output size [2]. Moreover, in general output length is exponential in the length of the input: McMullen proved a sharp upper bound with asymptotic growth of $O(m^d)$ for the number of vertices of a d-dimensional polytope defined as the intersection of m half-spaces [94]. However, instances of low dimension can be solved fairly efficiently and for a fixed dimension complexity only grows polynomially with the problem size.

Unbounded customer sets

To deal with unbounded customer sets we use the approach described above, approximating $\mathcal{X}$ with a bounded set $\hat{\mathcal{X}}$ by adding additional constraints (Eq. (7.63)). We compute an approximate value of Eq. (7.72) by replacing each facet $Y_{\mathtt{p}}^{(k)}$ with $\hat{Y}_{\mathtt{p}}^{(k)} = Y_{\mathtt{p}}^{(k)} \cap \hat{\mathcal{X}}$. The absolute error of the derivative of $d_{\mathtt{p}}(\mathcal{P}(u))$ induced by this approximation is not bounded in a similar fashion to Eq. (7.67). However,

$$\frac{\mathrm{d}}{\mathrm{d}u}\hat{d}_{\mathtt{p}}\left(\mathcal{P}(u)\right) = \sum_k \int_{\hat{Y}_{\mathtt{p}}^{(k)}} \frac{\hat{A}_u^{(k)}(u)\mathbf{x} + \hat{b}_u^{(k)}(u)}{\|\hat{A}^{(k)}(u)\|} f(\mathbf{x})\,\mathrm{d}x. \tag{7.74}$$

In other words, we have an exact representation of the derivatives of $\hat{d}$, which further encourages the interpretation of the approximation as an approximate scenario which is then solved exactly.

7.3.3 Algorithm

In this section we will give an overview over part of the software used in the simulations presented in Chapter 8. The software used to implement this algorithm is described in Appendix A.3. We will consider the sub-task that evaluates $\hat{d}_{\mathtt{p}}(\mathcal{P}(u))$ and its derivatives w.r.t. u, given a product mapping $\mathcal{P}(\cdot)$, a control vector u and the set of relevant customers $\hat{\mathcal{X}}$. We restrict ourselves to the case of linear utility functions that is presented above.

Booking probabilities are computed with the following steps:

(1) Compute the set of inequalities $A(u)\mathbf{x}+b(u) \geq 0$ that defines the customer set $\mathcal{X}_{\mathtt{p}}$ (Eq. (7.59)).

(2) Add the linear constraints that define the set $\hat{\mathcal{X}}$ to obtain $\hat{\mathcal{X}}_{\mathtt{p}} \cap \mathcal{X}_{\mathtt{p}}$.

(3) Compute a triangulation of the convex polytope $\hat{\mathcal{X}}_{\mathtt{p}}$.

(4) Compute a numerical approximation for the integral in the RHS of Eq. (7.65)

In case derivatives are required, proceed as follows:

(5) Compute the set of facets of $\hat{\mathcal{X}}_{\mathtt{p}}$.

(6) For each facet compute a triangulation. The triangulation does not have to be computed explicitly by an additional call to `CDDLIB` but can be efficiently derived from the triangulation of $\hat{\mathcal{X}}_{\mathtt{p}}$ and the incidence relation between vertexes and inequalities that is returned by `CDDLIB`.

(7) Numerically evaluate the RHS of Eq. (7.72) by computing a numerical approximation for the respective integral over each facet using `CUBPACK`.

Chapter 8

Numerical results

In this part we will present numerical results that were generated using the methods presented above.

This numerical study was not designed to estimate the potential benefit of applying these methods in a realistic scenario. In fact, we believe that this question is almost impossible to answer, because the baseline for comparison is unknown. There are no alternative methods to which we could compare our results. In industry practice, the fare structures consisting of price points and product characteristics are generated manually by expert analysts. The decisions are supported with data about historical sales, economic outlook, the competitive environment and other factors, but in the end are the result of the analysts' expert decision. The revenue gain that can be achieved by optimizing fare structures using mathematical optimization techniques therefore not only depends on the quality of the customer choice model that is used (which also has to account for the outside influences listed above), but also on highly depends on the quality of the decisions made by expert analysts.

Instead the primary purpose is to gain qualitative insight about the structure of the pricing problem and the behavior of the numerical methods presented in this work. The pricing problem is a two-level problem, where the inner one is the standard RM inventory control problem, for which there is extensive scientific literature. This inventory control problem cannot be solved exactly for networks with more than a handful of resources. Because the focus of this thesis lies on the outer optimization problem of choosing optimal prices and product characteristics, this study is restricted to the case of a single flight leg. This way we can efficiently solve the inventory control problem to optimality, and none of the observations, e.g. regarding non-convexity of the problem, can be artifacts of heuristic solutions to the inner problem. The main questions we wanted to address are the following:

How efficient are our numerical methods? Is the proposed algorithm efficient enough to solve small toy problems reasonably fast? How does the computational effort scale with the problem size, i.e. the number of products?

How significantly does the quality of the solution depend on the number of products? From a purely mathematical standpoint, it is clear that the ability to price and sell a larger number of products can only improve the outcome for the airline, because the respective larger pricing problem is essentially a relaxation of the smaller problem. From a business perspective, the complexity of manually designing and managing a larger number of products and price points can lead to suboptimal decisions and therefore may very well degrade overall performance. Here, we will ignore the latter and only focus on the former.

There are three main reasons why having a larger number of products, or booking classes, would benefit the airline.

- A larger number of booking classes offers more room for customer segment specific products.

- The customer mix varies over the course of the booking horizon, while products remain constant over time. A larger number of different products and price points together with availability control therefore helps to match pricing to changing customer behavior.

- With a dynamic control policy, the airline can adjust booking class availability dynamically, depending on the realization of the stochastic booking process. Again, having more booking classes, and therefore more price points, allows to control the booking process more precisely, depending on the realization of demand.

How non-convex is the pricing problem? Even a very small problem like this one cannot be expected to be convex. To the contrary, abstract product restrictions are pricing tools, which are aimed at segmenting demand into customer segments and therefore allowing the airline to offer segment-specific products. This way, the airline can charge different prices to different customer segments at the same time, and exploit the fact that a customer's willingness-to-pay and their reaction to the tariff conditions are highly correlated. Intuitively, a stronger segmentation increases the airline's ability to exploit the willingness-to-pay of business type customers. It therefore seems natural that for good solutions the restriction variables are often near or at their bounds.

In addition, during the fare transformation step (Section 3.5) it is possible that a product is inefficient, meaning that it is not contained in any efficient offer set (see Eq. (3.63)). This means, that an equivalent solution with fewer products is given by removing the inefficient product from $\mathcal{P}$. Because an inefficient product is never offered and never sold, its product characteristics do not have any impact on the total expected revenue for the airline. Except for degenerate cases, slight perturbations of product characteristics will not change efficiency of a product. Combining these facts, it is clear that locally the objective function value does not depend on the product attributes of inefficient products. Therefore, adding an inefficient product to a locally optimal solution will yield a local optimum for a higher dimensional instance of the pricing problem. As a result, an instance of the pricing problem may have a large number of local optima, including plateaus, that arise from lower dimensional solutions.

How well can our algorithm handle the non-convexity? Due to the high computational complexity, we do not apply a global optimization algorithm but a quasi-Newton method, which only converges to local optima. The quality of our solution therefore depends strongly on the starting value we use to initialize the optimizer. We are therefore interested in estimating the average quality of the solution or, more precisely, the probability of ending up in each local optimum, when starting from random initial guesses.

Particularly the second question requires a very detailed analysis of the results data. In order to keep the amount of data manageable, we decided to focus on a small set of problem instances, which only differ in the number of products, but all share the same demand model with one fixed set of model parameters.

8.1 Problem setup

8.1.1 Network

We consider a scenario with a single flight leg with a capacity of 100 seats.

8.1.2 Price structure

We assume that there is no differentiation by POS or other criteria. Since there is only one itinerary in the network, this means that there is only one set of products that defines the offer for all customers.

Apart from price $\mathfrak{p}^{\text{price}}$, there is only one additional binary attribute that indicates rebooking flexibility. Each product is therefore given as a two-dimensional vector $\mathfrak{p} = (\mathfrak{p}^{\text{price}}, \mathfrak{p}^{\text{flex}}) \in \mathbb{R} \times \{0, 1\}$. In order to simplify notation in the demand model, we use the convention that $\mathfrak{p}^{\text{flex}} = 0$ for flexible products and $\mathfrak{p}^{\text{flex}} = 1$ for non-flexible products.

Amongst other things, in this study we wish to explore the effect of the number of products on overall revenue. We therefore solve multiple instances of the problem, which only differ in the number of products $M = 1, \dots, 9$.

	Exp. arrivals			Willingness-to-pay		Non-flex disutility	
	μ_1	μ	μ_3	μ	σ	μ	σ
Leisure	20	30	15	0.3	0.3		
Business	0	6	24	1	0.5	0.5	0.5

Table 8.1: Arrival rates and choice parameters

8.1.3 Demand

Demand is modeled as described in Section 4.2.3. Each customer is of one of two customer types, which differ in their arrival rates over the booking horizon as well as choice behavior. The booking horizon $[0,1]$ is partitioned into three time periods I_1, I_2, I_3, where I_3 is the closest to departure. For each customer type, arrivals follow a non-homogeneous Poisson process, where arrival rates vary between time intervals, but are constant within each time interval. The Poisson arrival process for a customer type is therefore uniquely determined by the expected number of arrivals μ_1, μ_2, μ_3 for the three time periods. The two customer types are:

Leisure customers tend to book earlier, have a low willingness-to-pay, and do not care about product flexibility at all, i.e. they base their decision purely on price and—if at all—will always buy the cheapest product available. Willingness-to-pay is the only attribute of the leisure customer type, which is therefore described by a single random variable $\mathbf{X}^{\text{wtp}}_{\text{leisure}}$. The utility function is given by

$$u_{\text{leisure}}(\mathbf{x}_{\text{leisure}}, \mathfrak{p}) = \mathbf{x}^{\text{wtp}}_{\text{leisure}} - \mathfrak{p}^{\text{price}}.$$

Business customers tend to book later, have a higher willingness-to-pay, and associate a non-negative disutility cost with non-flexible products. In addition to willingness-to-pay, the business type has an additional attributes characterizing the customers valuation of non-flexible products. It is described by the random vector $\mathbf{X}_{\text{business}} = (\mathbf{X}^{\text{wtp}}_{\text{business}}, \mathbf{X}^{\text{flex}}_{\text{business}})$, and the utility function is given by

$$u_{\text{business}}(\mathbf{x}_{\text{business}}, \mathfrak{p}) = \mathbf{x}^{\text{wtp}}_{\text{business}} - \mathfrak{p}^{\text{price}} - \mathbf{x}^{\text{flex}}_{\text{business}}\mathfrak{p}^{\text{flex}}.$$

Willingness-to-pay for both customer types as well as the disutility cost associated with non-flex-products for the business type are independent random variables, each following a truncated normal distribution that is censored below at zero. Table 8.1 shows the parameters of the underlying normal distributions and the expected number of customer arrivals for each time interval. For example, the willingness-to-pay of a random customer of the Business type has the same distribution as $\mathbf{X} \mid \mathbf{X} \geq 0$, where $\mathbf{X} \sim \mathcal{N}(1, 0.5^2)$.

8.2 Solution algorithm

As shown in the following sections, the pricing problem is highly non-convex even for this simple toy problem. We do not apply a global optimization algorithm, but instead attempt to find the global optimum—or at least a good local optimum—by applying local optimization with a large number of different initial guesses. More precisely, for each problem instance (determined by the number of products M) we randomly choose 1000 vectors of starting values, where every product $\mathfrak{p} = (\mathfrak{p}^{\text{price}}, \mathfrak{p}^{\text{flex}})$ is drawn from a uniform distribution on $[0,2] \times [0,1]$. Local optimization is performed using the quasi-Newton method of Byrd et al. [24], which uses a limited memory BFGS method to approximate the Hessian and allows for box-constraints.

The algorithm for evaluating the objective function for a given set of products is shown in Algorithm 2. The algorithm for computing the gradient of the objective function w.r.t. product attributes is very similar and therefore not listed explicitly. There are two main differences. Firstly, one not only computes the booking probabilities (Line 6), but also their gradients w.r.t the control parameters (i.e. all product attributes). Secondly, when solving the dynamic program (Line 15) one also solves the adjoint equation and computes the gradient of the value function w.r.t. parameters (see Eq. (5.44)).

Algorithm 2 Expected revenue for a single-leg

function ExpRevSingle($\mathcal{P}$, n, $\mathcal{T}$, C) ▷ $\mathcal{P} \subset \mathbf{P}$ is a finite set of products, n is the number of time intervals, $\mathcal{T} = \{\mathbf{T}_1, \ldots, \mathbf{T}_L\}$ is a set of customer types, and C is capacity
  $M \leftarrow |\mathcal{P}|$
  for all $\mathbf{T} = (\mathbf{X_T}, u_\mathbf{T}, \lambda_\mathbf{T}) \in \mathcal{T}$ **do** ▷ For every customer type,
    for all $\mathcal{S} \subseteq \mathcal{P}$ **do** ▷ every offer set,
      for all $\mathfrak{p} \in \mathcal{S}$ **do** ▷ and every offered product
        $d_{\mathcal{S},\mathfrak{p},\mathbf{T}} \leftarrow \mathrm{P}\left[\mathfrak{p}^*(\mathbf{X_T}, \mathcal{S}) = \mathfrak{p}\right]$ ▷ compute the probability that a random customer of type $\mathbf{T}$ will purchase $\mathfrak{p}$ given the offer set $\mathcal{S}$ (see Section 7.3)
  for all i=1, . . . , n **do** ▷ For every time period
    for all $\mathcal{S} \subseteq \mathcal{P}$ **do** ▷ For every offer set
      $\mathbf{D}_\mathcal{S} \leftarrow \sum_{\mathbf{T}\in\mathcal{T}} \sum_{\mathfrak{p}\in\mathcal{S}} \lambda_\mathbf{T} d_{\mathcal{S},\mathfrak{p},\mathbf{T}}$ ▷ compute total demand
      $\mathbf{R}_\mathcal{S} \leftarrow \sum_{\mathbf{T}\in\mathcal{T}} \sum_{\mathfrak{p}\in\mathcal{S}} \lambda_\mathbf{T} d_{\mathcal{S},\mathfrak{p},\mathbf{T}} \mathbf{y}_\mathfrak{p}$ ▷ compute total revenue
    $\mathbb{S}^*{}_i \leftarrow$ EfficientFrontier$(\mathbf{D}, \mathbf{R})$ ▷ Compute the set of efficient offer sets (see Section 3.5) for time period i
    for all $\mathcal{S} \in \mathbb{S}^*$ **do** ▷ For every efficient set compute transformed demand and revenue (see Definition 3.5.7)
      $\tilde{D}_{i,\mathcal{S}} \leftarrow$ TransformedDemand$(\mathcal{S})$
      $\tilde{\mathbf{y}}_{i,\mathcal{S}} \leftarrow$ TransformedRevenue$(\mathcal{S})$
  $V \leftarrow$ ValueFunction$(\mathbb{S}^*, \tilde{D}, \tilde{\mathbf{y}})$ ▷ Compute value function by solving dynamic program Eq. (5.17)
return $V_C(0)$

8.3 Results

8.3.1 Computational time

All computations were performed on a workstation with an AMD Phenom II X6 1055T processor at 1.5 Ghz and 8 GB RAM using Ubuntu 12.04. As shown in Fig. 8.1 and Table 8.2, CPU time grows exponentially with the number of products. This is due to the fact that we compute booking probabilities for every product and every possible offer set $\mathcal{S} \in \wp(\mathcal{P})$, which means that $M2^M$ integrals of the form Eq. (7.58) have to be solved for each evaluation of the value function. A large number of the computed values belong to inefficient offer sets and are discarded during the fare transformation step. Therefore, in this naive approach we expend a high amount of computational time on computing quantities that are irrelevant for the actual value of the objective function. If and how computational effort can be reduced by avoiding these unnecessary computations is a question for future research.

8.3.2 Benefit of additional products

Figure 8.2 shows the expected revenue that can be achieved depending on the number of products. Each value displayed on the vertical axis in Fig. 8.2a is the maximum over the objective function values of all local optima that were found with the 1000 random starting values for the given number of products. As is to be expected, with an increasing number of products the maximal revenue that can be achieved grows, while the marginal benefit of having additional products decreases (Fig. 8.2b). The large revenue increase of almost 10% between the cases with one and two products compared to the rather moderate increases with each additional product suggest that—at least for this customer model—the ability to segment customers has a much stronger impact on revenue than having additional price points to choose from when dynamically varying prices over the course of the booking period.

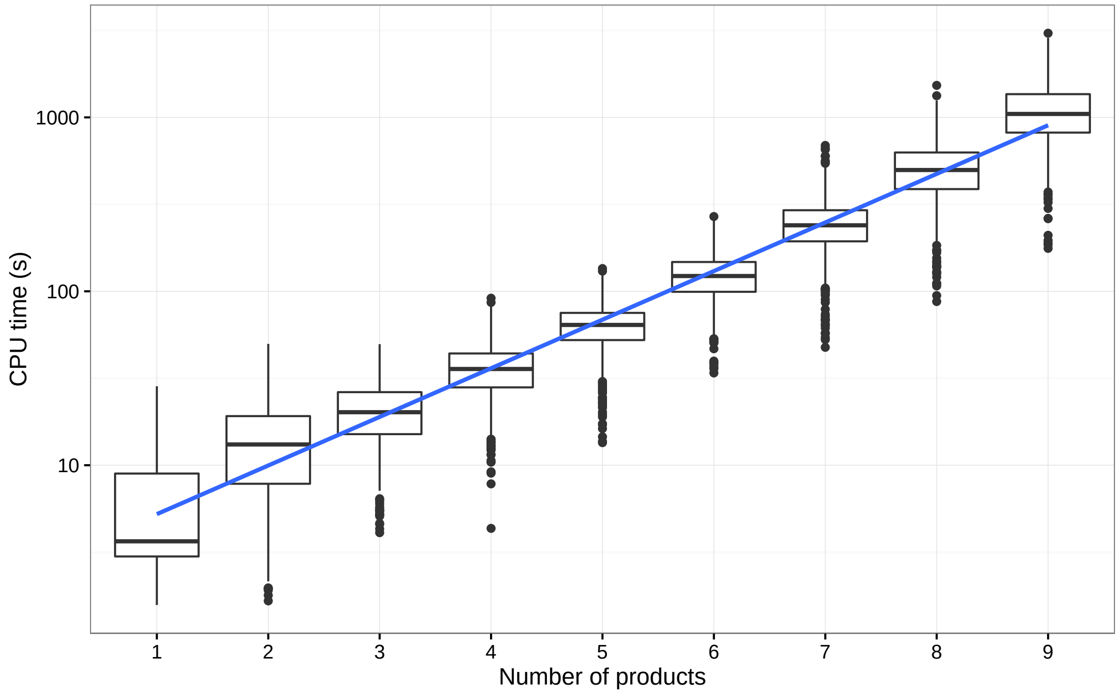


Figure 8.1: CPU time depending on the number of products

# Products	mean	median	min	max
1	6.42	3.65	1.57	28.45
2	14.08	13.16	1.66	49.85
3	21.18	20.18	4.10	49.67
4	36.49	35.76	4.34	91.32
5	63.84	64.08	13.49	135.19
6	123.95	122.45	33.94	269.18
7	250.87	239.62	47.63	689.83
8	520.87	498.20	87.36	1528.21
9	1113.10	1046.59	176.64	3049.13

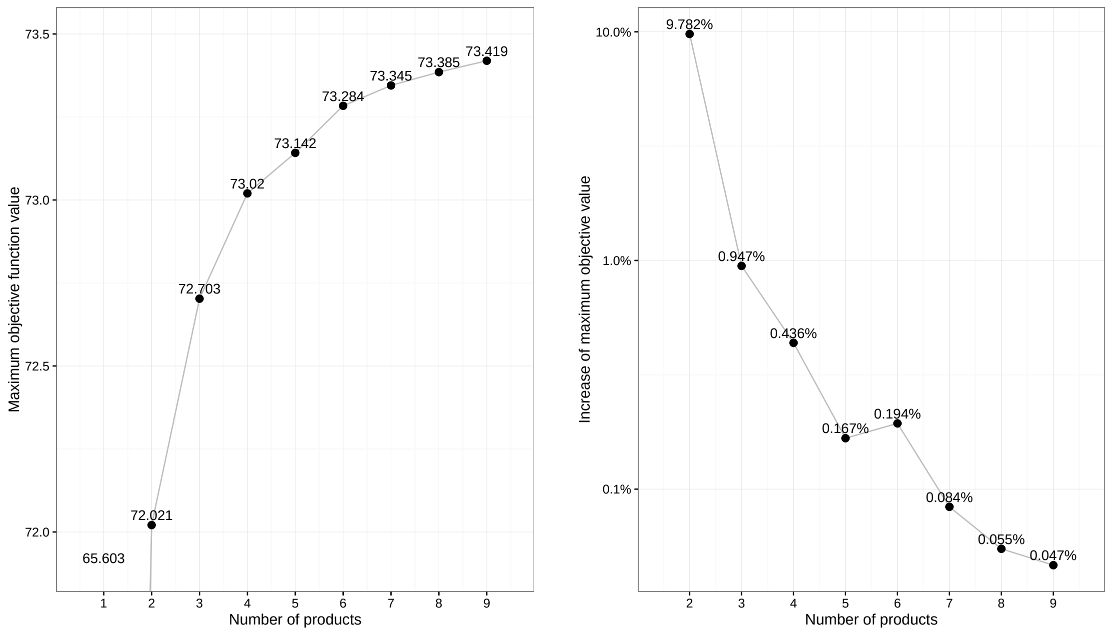
Table 8.2: CPU time

Note that adding a sixth product leads to a higher marginal revenue gain that the fifth product. This is possibly a sign that for the case $\hat{M} = 5$ the global optimum was not found.

8.3.3 Non-convexity of the objective function

It is one of the goals of this thesis to understand and describe in qualitative terms the non-convexity of the pricing problem. We will do so by identifying the unique local optima of our example problem that were found in the numerical experiments and describe the convergence behavior of the optimization algorithm. We see that even small instances of this simple scenario have many local optima.

First, we observe that the problem instances are nested in the sense that each solution for a problem with M products can be extended to an equivalent solution of the $M + 1$ product case. Let $\mathcal{P} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_M\}$ be a set of products and $\mathbf{S}(t)$ an optimal offer process. We call a product $\mathfrak{p} = \mathfrak{p}_k \in \mathcal{P}$ *inefficient* w.r.t to $\mathbf{S}$, if its expected number of bookings—and therefore its expected



(a) Objective function value

(b) Relative improvement

Figure 8.2: Maximum objective function value and relative improvement depending on the number of products

revenue—over the whole booking horizon is zero, i.e.:

$$\mathbb{E}\left[\int_0^T \mathrm{d}\mathbf{N}_k(\mathbf{S}(t))\right] = 0 \tag{8.1}$$

This will for example be the case if $\mathfrak{p}$ is either so cheap that the airline would never want to offer it (i.e. $\mathcal{P}(\mathfrak{p} \notin \mathbf{S}(t)) = 0$ for all t) or so expensive that no customer would be willing to purchase it. Given a local optimum $\mathcal{P}$ of the pricing problem, we can always add an additional product $\mathfrak{p}'$ that satisfies Eq. (8.1). If $\mathfrak{p}'$ is chosen such that all $\mathfrak{p}''$ in a neighborhood of $\mathfrak{p}'$ are inefficient as well, small perturbations $\mathfrak{p}'$ will not have any impact on the expected revenue and, consequently, $\mathcal{P} \cup \{\mathfrak{p}'\}$ is a local optimum of the pricing problem with $M + 1$ products.

On the other hand, a locally optimal solution $\mathcal{P}$ of the pricing problem with $|\mathcal{P}| = M$ will often contain inefficient products, which do not have any impact on the objective function value. Let always $\hat{\mathcal{P}} \subseteq \mathcal{P}$ be the set of efficient products w.r.t. the optimal control strategy for $\mathcal{P}$, and let $\hat{M} = |\hat{\mathcal{P}}|$ be its cardinality. The histogram Fig. 8.3 shows the distribution of $\hat{M}$ depending on M. For example, for the problem instance with six products slightly less than 50% of the 1000 computed local optima have six efficient products, while all other solutions contain at least one inefficient product and can therefore be regarded as solutions of the smaller problem instances with $M < 6$. As expected, with increasing number of products it becomes harder to find solutions where all products are efficient. However, even for the largest problem instance with $M = 9$, more than 13% of the computed local optima had no inefficient products.

Figures 8.4 to 8.11 show the local optima found, grouped by the number of efficient products. In all figures, a local optimum is represented by a vertical line and the horizontal axis represents the objective function value. The upper diagram is a histogram showing the frequency of occurrence of the local maximum relative to the total number of local optimal with the same number of efficient products. The lower diagram shows the local optima, each of which is a set of efficient products $\hat{\mathcal{P}}$. Every product is represented by a point on the vertical line, where price is represented on the vertical axis, while the binary product attribute—booking flexibility—is represented by the shape of the product. For all $\hat{M} = 2, \ldots, 9$ there were additional solutions that are not included in the plots. These solutions contain only flexible products and therefore—due to the lack of customer

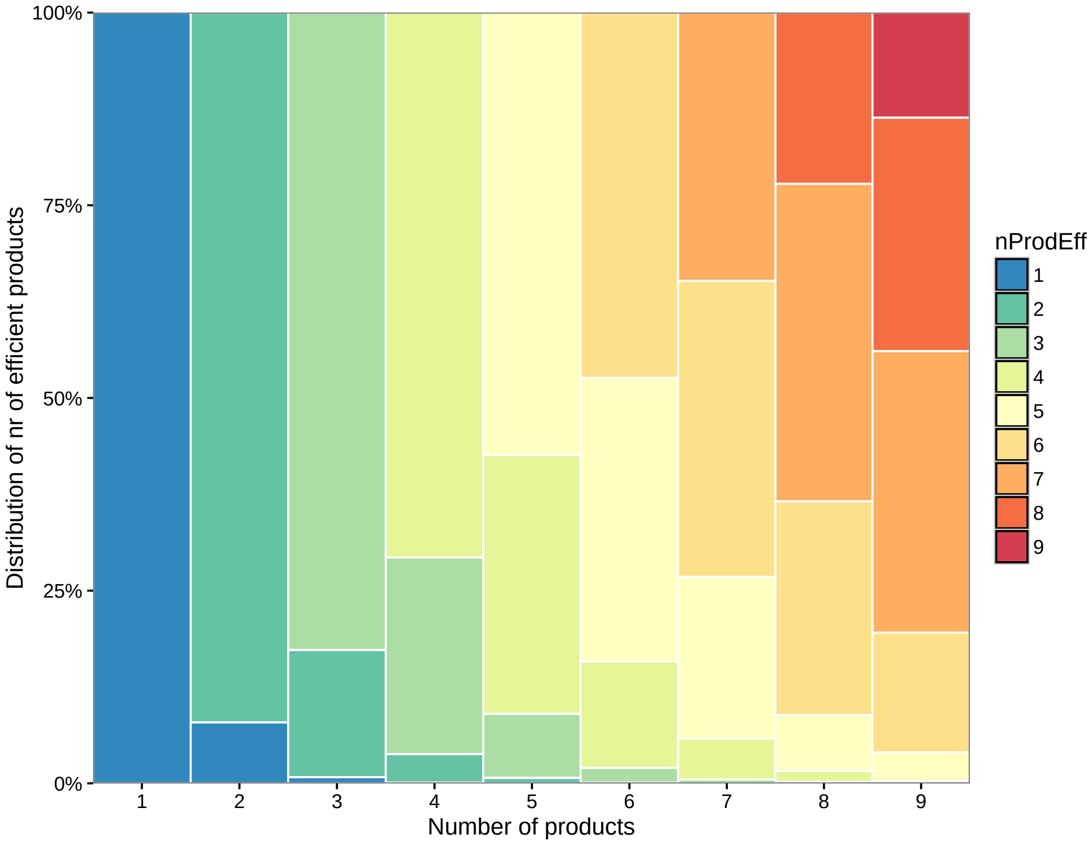


Figure 8.3: Distribution of number of efficient products depending on the number of products

segmentation—all have very low expected revenues. These solutions are omitted, because they would have lead to a large horizontal gap in the diagrams, making the relevant portion much harder to read.

Clearly the number of different local optima increases with the number of efficient products. Solutions with the same number of efficient produts often differ in the number of flexible products and in the pattern of flexible/non-flexible products when ordered by price. However, some solutions only differ in price, but not in the order of flexible/non-flexible products. For example with four efficient products (Fig. 8.6) we have four local optima with two flexible and two non-flexible products, and for the two of these with the highest objective function values the two most expensive products are flexible while the two cheapest products are not, and the only difference is the distribution of price points.

In summary, we see that even a small problem with very simple customer models and only two product attributes can be heavily non-convex. However, Fig. 8.3 and the histograms in Figs. 8.4 to 8.11 suggest that the chance of finding a good solution is fairly high when applying a multi-start approach with sufficiently many different starting values: even in the hardest instance of $M = 9$ the chance to find a solution with $\hat{M} = 9$ is higher than 13%. In addition, for every $\hat{M}$ the frequency of occurrence for the best local optimum that was found is higher or close to 5%.

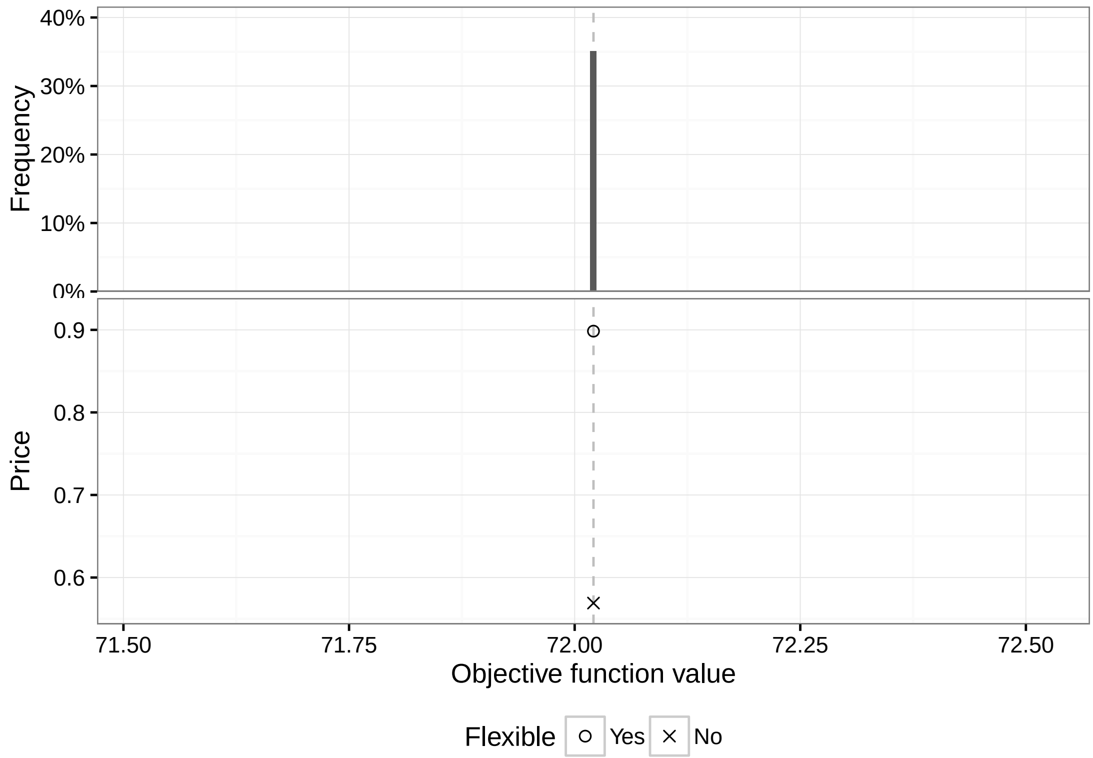


Figure 8.4: Local optima with 2 efficient products

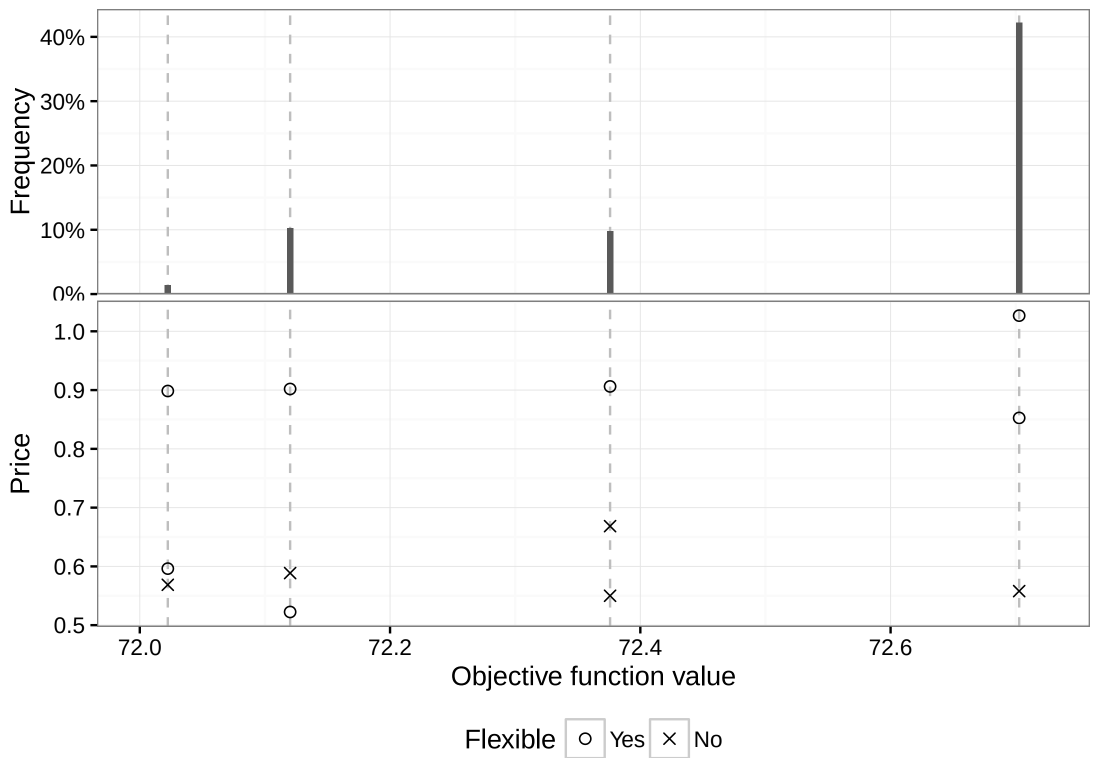


Figure 8.5: Local optima with 3 efficient products

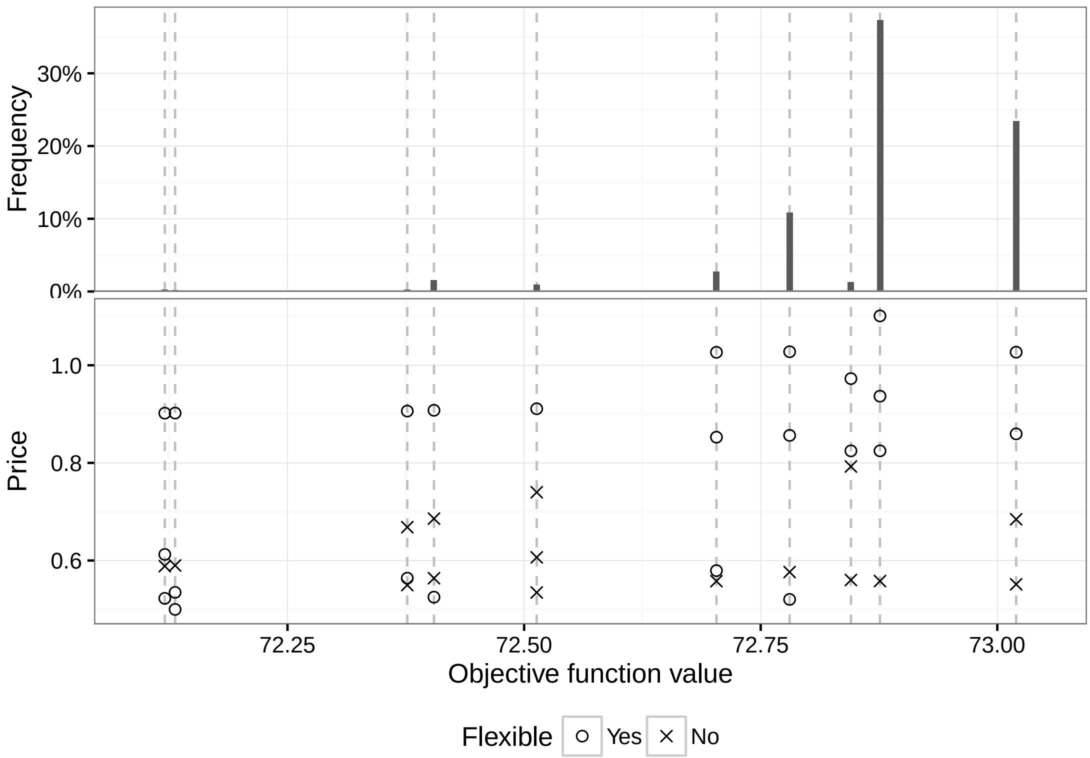


Figure 8.6: Local optima with 4 efficient products

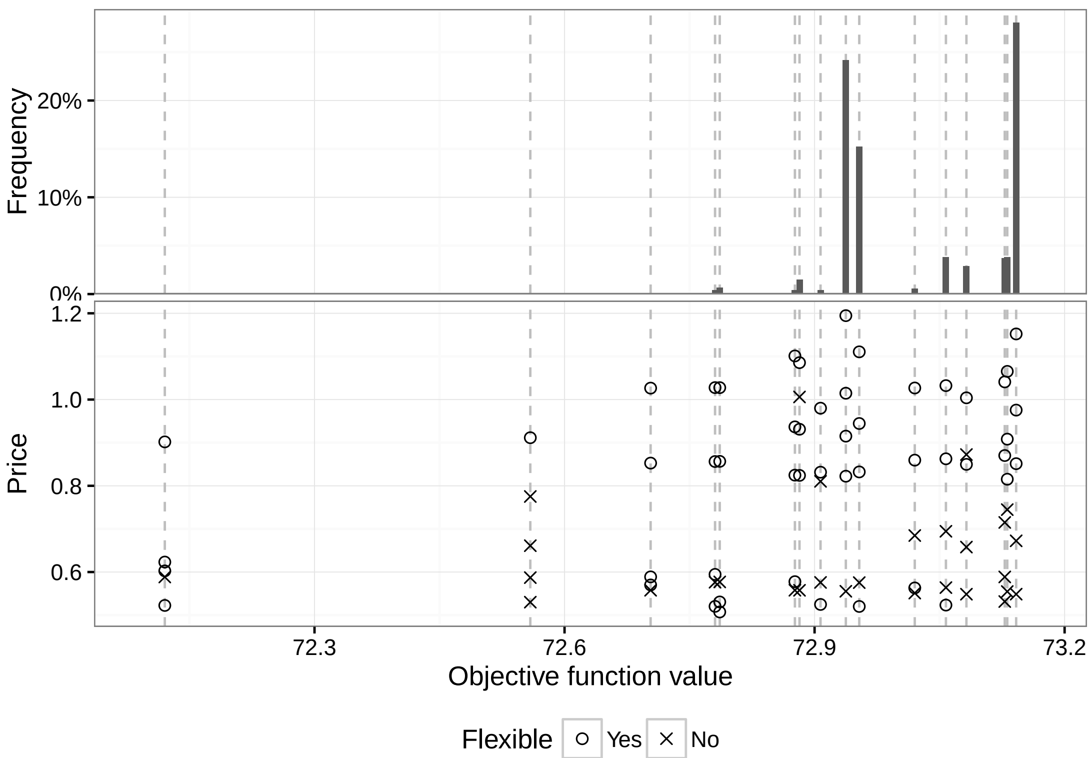


Figure 8.7: Local optima with 5 efficient products

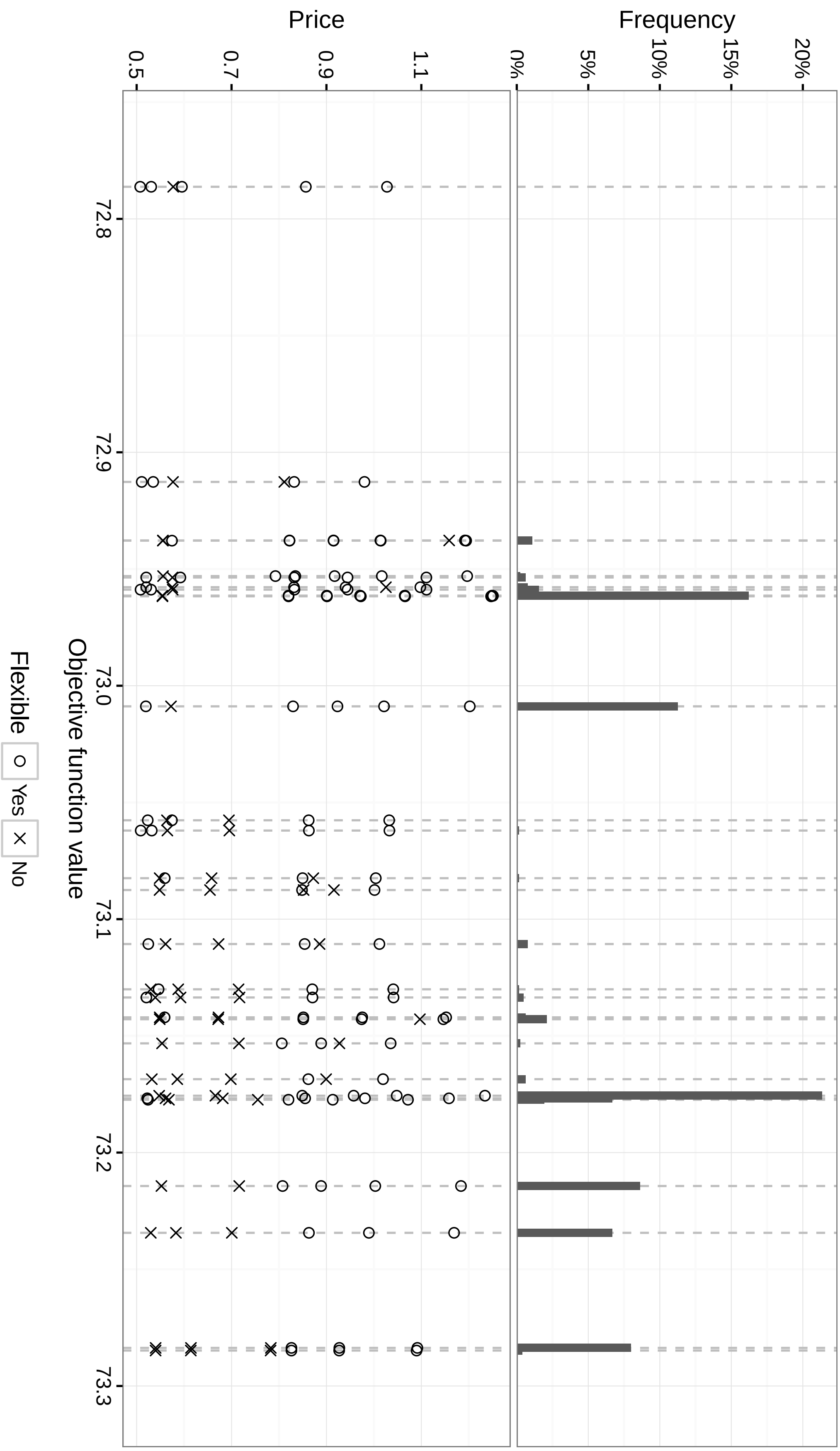


Figure 8.8: Local optima with 6 efficient products

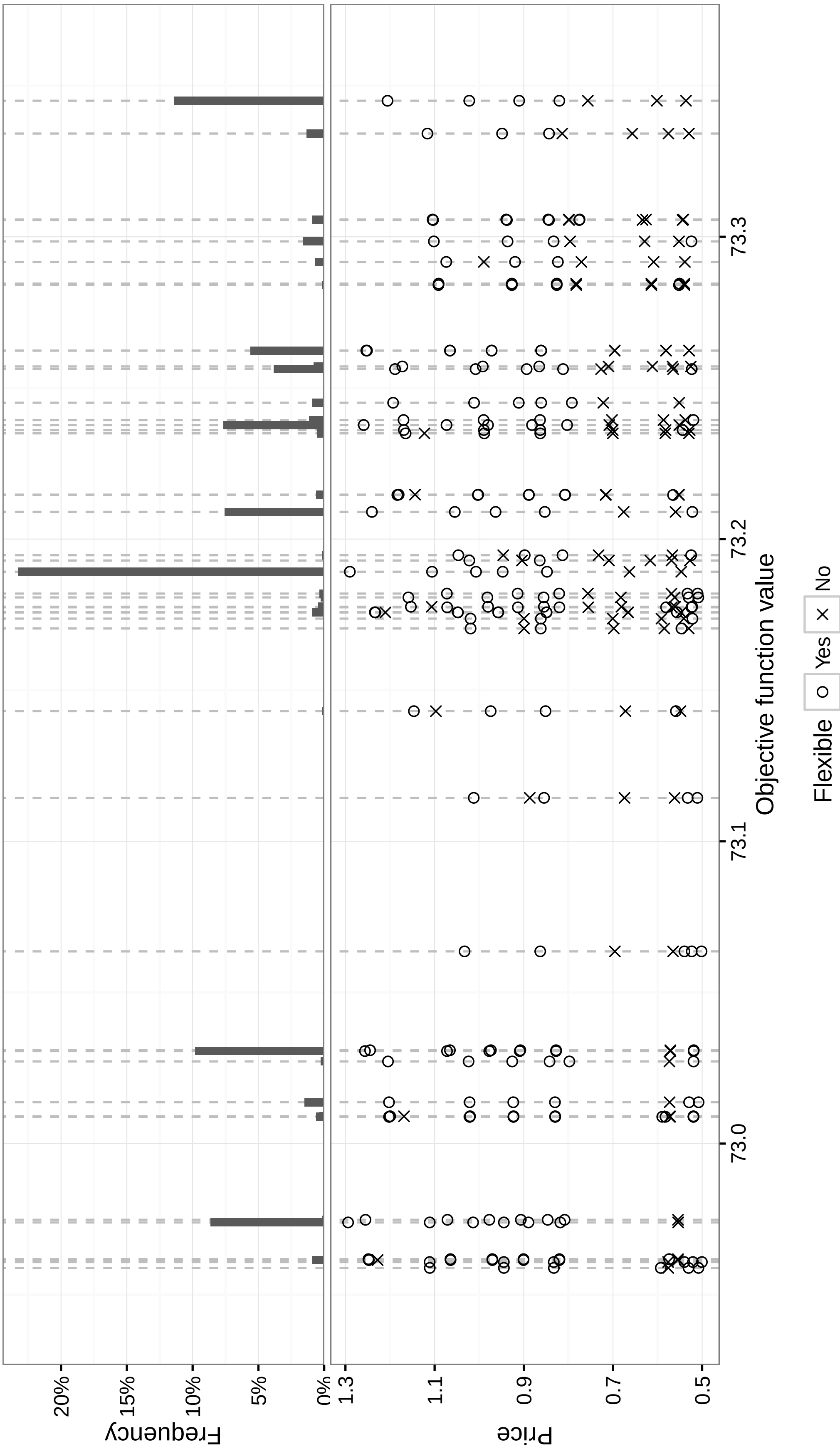


Figure 8.9: Local optima with 7 efficient products

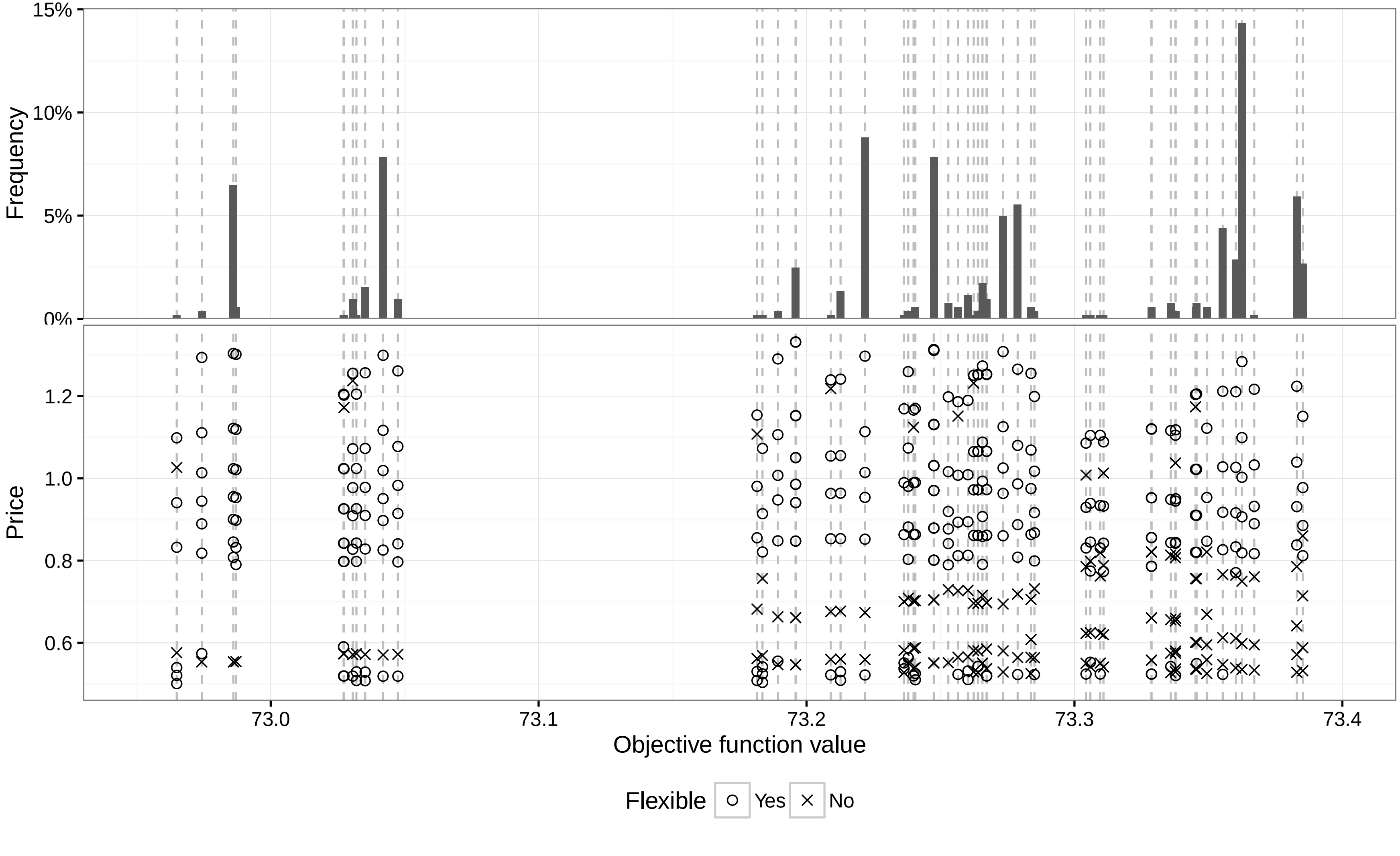


Figure 8.10: Local optima with 8 efficient products

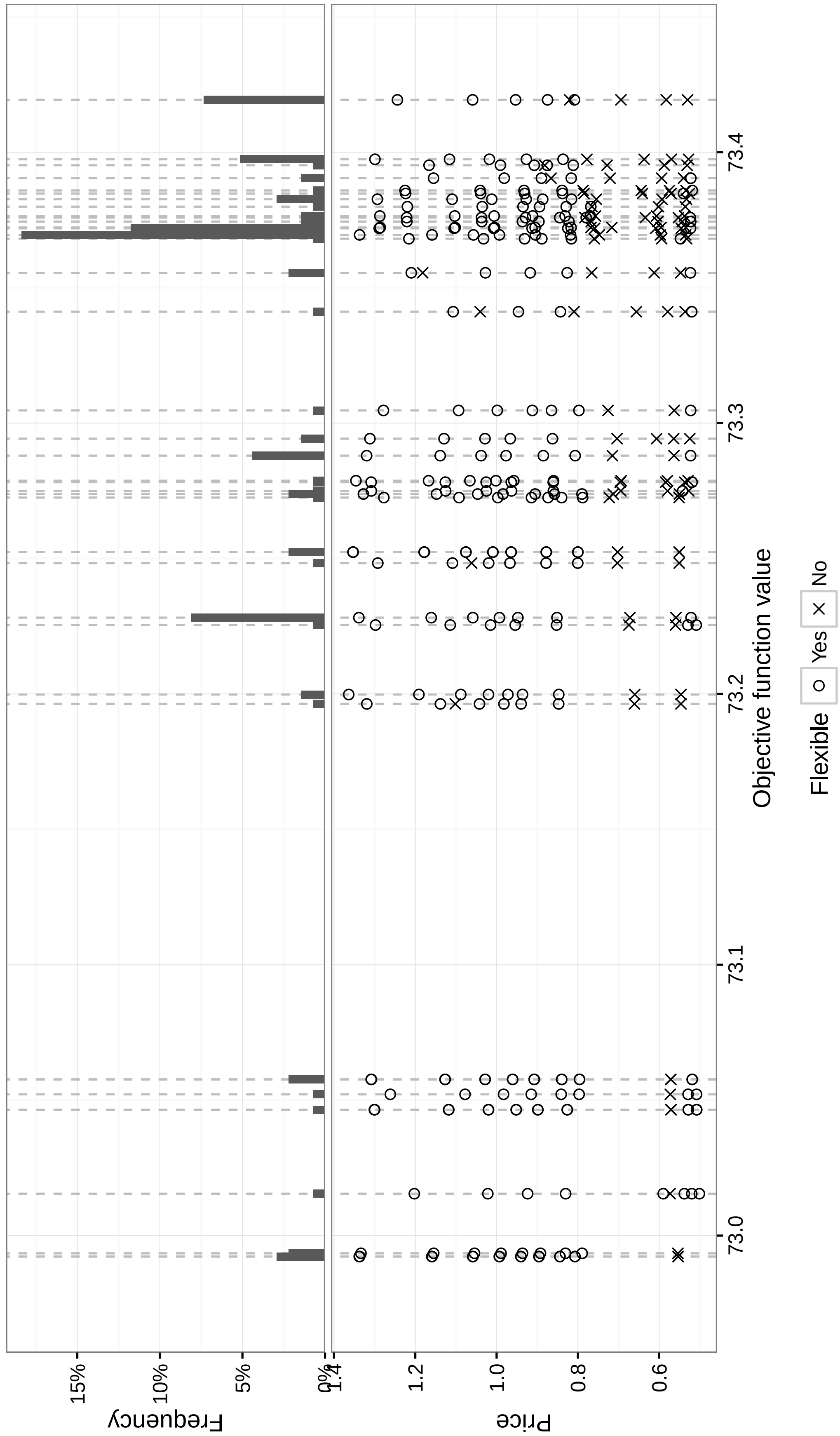


Figure 8.11: Local optima with 9 efficient products

Conclusion

Summary

The airline pricing problem deals with defining revenue-optimal prices and fare conditions for a limited number of booking classes. Because of its strong impact on airline revenue and profit, it is highly relevant in industry practice. Still, unlike the Revenue Management problem, pricing has not gotten a lot of attention in the scientific literature and is therefore usually done manually by analysts with little decision support. This work is the first comprehensive treatment of this problem in the scientific literature and describes a framework to formulate and numerically solve the pricing problem under only mild assumptions about customer choice behavior.

The pricing problem is strongly related to the classic airline Revenue Management problem, which, assuming given and fixed fares, aims to maximize revenue by dynamically controlling availability of each booking class over the course of the booking horizon. We formulate the pricing problem as a two-level optimization problem where the control variables of the outer problem — prices and fare conditions — determine the constant parameters of the RM problem. Consequently, evaluating the objective function of the pricing problem for a given set of pricing parameters requires solving the respective instance of the choice-based revenue management availability control problem, which we formulate as a parametric optimal control problem parameterized by prices and expected demand rates for each booking class. Between the two layers we have the customer choice model, which maps prices and fare conditions to expected aggregate demand.

For the solution we combine methods from revenue management, customer choice theory, numerics, optimization, and optimal control theory. A key for efficiently solving the outer problem with gradient-based optimization methods is a sensitivity analysis of both the customer choice model and the revenue management problem. To this end we present a sensitivity analysis of expected customer demand as a function of the pricing parameters for a general class of customer choice models. In addition, viewing the single-leg revenue management dynamic program as an optimal control problem, we derive an adjoint equation that can be used to efficiently compute the sensitivity of optimal expected revenue as a function of prices and expected demand. Combined with suitable numerical methods this allows us to use gradient-based numerical optimization methods to efficiently solve the single leg pricing problem. We thoroughly analyze a number of instances of an example problem and show that the pricing problem can be highly non-convex even for cases with only a single resource and simple fare structures.

For the extension to the network case with multiple flight legs connected by transfer traffic we require a sensitivity analysis of the dynamic network revenue management problem. This problem cannot be solved to optimality even for a few resources due to the curse of dimensionality and in practice is solved heuristically. Building on the well-known LP-DP decomposition heuristic for the network RM problem, we heuristically extend the sensitivity analysis of the RM problem to the multi-resource case. Combined with the sensitivity analysis of the customer choice model this allows us to heuristically solve the network pricing problem.

As a side result our sensitivity heuristic gives rise to an improved version of the LP-DP decomposition that uses probabilistic instead of deterministic displacement costs. We show in a simulation study that this new method for solving the dynamic network RM problem is computationally feasible and on average significantly improves overall network revenue.

Directions for further research

In this thesis we develop a framework to formulate the airline pricing problem as a mathematical optimization problem and present computationally feasible solution methods. However, due to the complexity of the problem a number of questions remain open and should be subjected to further research.

A particularly important and somewhat natural extension is a decomposition of the network pricing problem along itineraries. The objective function of the network pricing problem is separable. It is the sum of expected revenue over all products, and expected revenue for each product is independent of all products that belong to a different itinerary. The objective function can therefore be written as the sum of expected revenue over all itineraries, with disjoint sets of control variables for the summands. The itineraries are coupled via the capacity constraints in the underlying network RM problem. This structure lends itself to a dual decomposition approach, where pricing for each itinerary is optimized separately given the current estimate of the dual variables, which for the pricing problem are the (distributions of) the bid prices for each leg. A dual decomposition would be strongly related to our heuristic sensitivity analysis and roughly work as follows:

(1) Generate an initial guess for all pricing variables.

(2) Given the pricing variables, compute bid price vectors for all resources using the LP-DP decomposition for the network RM problem. Compute the adjoint states, which represent the distribution of remaining inventory, as described for the sensitivity analysis.

(3) Given the bid price distributions, optimize pricing on each itinerary to maximize its network contribution.

(4) iterate (2) and (3) until convergence.

This would allow significant improvements in computation time, because both the primal and the dual problem can be parallelized along itineraries and resources respectively. In addition, this would greatly help implementing pricing optimization in industry practice, where it is usually not feasible to simultaneously update prices and fare conditions for the whole network, but where such decisions are taken on a market-by-market basis.

Evaluating the customer choice model to compute for a given set of pricing parameters expected demand and its gradient w.r.t. the controls requires evaluating all possible offer sets and is therefore exponentially complex in the number of products (within one itinerary). The same problem arises in choice-based revenue management and several solution approaches are known (for example the column-generation procedure by [19]), which could potentially be adapted to the pricing problem.

As illustrated by an example the pricing problem can be highly non-convex. It is therefore natural to evaluate whether methods for global optimization can be applied.

Lastly, it would of course be valuable to test the method in a real-world scenario. As a prerequisite this requires choosing and estimating a suitable customer choice model on real data.

Trends in airline distribution

During the time this thesis was written airlines have started working towards modern, internet-based distribution technology to replace the legacy world of GDS distribution that was the dominating technology for the past half decade [39]. The IATA NDC initiative described in IATA resolution 787 [62] sets a standard for airline shopping and booking that in principle allows true dynamic pricing as outlined in Section 3.6. In particular, once this standard is widely adopted by airlines, travel agents and other distribution partners worldwide, airlines will not have the need any more to file static fares into a fixed number of booking classes. In terms of the pricing problem analyzed in this thesis, this means that the price itself would move from the outer problem into the real-time decision in the dynamic program.

However, our results remain relevant and applicable for several reasons. Firstly, due to the magnitude of required change and the high costs of replacing existing infrastructure, NDC will be implemented rather slowly in a step-by-step fashion, while existing GDS channels are still strongly relevant. Secondly, even with dynamic pricing capabilities there will still be a number of static pricing parameters that set the framework for real-time price optimization. For example, airlines will still need to publish a fixed lowest (or promotional) fare that can be used for marketing purposes. Furthermore, fees such as rebooking fees or the cost for extra luggage will usually be static to ensure transparency for customers and satisfy local and international regulations. Thirdly, the sensitivity analyses of both the inventory control dynamic program and of the customer choice probabilities have potential applications outside of the pricing problem, such as the improved network decomposition heuristic described in Section 6.3.

List of Figures

List of Tables

Glossary

Bid Price	Expected marginal value of one unit of capacity or, equivalently, opportunity cost associated with selling a seat.. 70
Booking Class	Also called *fare class.* Abstract classification of a product, used for availability control. Usually identified by one capital letter.. 55
BVP	boundary value problem. 29
CAB	Civil Aeronautics Board. 56
CDF	cumulative distribution function. 51
CQ	Constraint Qualifications. 9
CRCQ	Constant Rank Constraint Qualification. 9
DAE	differential algebraic equation. 23
DAVN	Displacement adjusted virtual Nesting. 77
DCP	Data Collection Point. 57
DLP	Deterministic Linear Program. 74–76
DP	dynamic program. 71–76, 83, 84, 97, 98, 104, 105, 110–113, 115, 118, 120, 121, 127, 132, 133, 135, 137, 138, 140, 145, 150, 151, 157
EMSRa/EMSRb	Heuristic based on the concept of Expected Marginal Seat Revenue. 56, 70, 71, 77
GDS	*Global Distribution System.* Computerized system for booking products related to travel, such as airline seats, hotel rooms or rental cars. Interfaces for providers, e.g. airlines or hotels, and agents, e.g. travel agencies or web based search and booking services. 55
HJB-equation	Hamilton-Jacobi-Bellman equation. 104, 107, 108, 111–114, 129
IIA	Independence of Irrelevant Alternatives. 33, 34, 38, 43, 44, 51, 52
IID	*Independent and Identically Distributed.* Each random variable has the same distribution and all variables are mutually independent. 49, 52, 99
IVP	initial value problem. 14, 16, 17, 19, 22, 23, 104, 110–112, 118, 119, 140

KKT	First order Karush-Kuhn-Tucker necessary conditions for a local optimum. See Theorem 1.1.6. 8–11, 13, 25
LICQ	Linear Independence Constraint Qualification. 8–11, 13
LP	Linear Program. 74, 75, 142, 157
MDP	Markov Decision Process. 106
MFCQ	Mangasarian-Fromovitz Constraint Qualification. 9
MNL	multinomial logit. 51, 52, 99
MNP	multinomial probit. 51
NLP	Nonlinear Program. 6, 7, 10, 28, 29, 86, 87, 90, 94–96
No-Show	A passenger who does not cancel a booking beforehand and does not claim their seat at departure. 54, 63
O&D	*Origin/Destination.* Pair of cities describing from where to where someone intends to travel, independent of the path one uses to get there. 55, 56, 89, 91, 92, 131, 149
OCP	Optimal Control Problem. 21, 23, 24, 26–29, 104, 109–111, 117, 118, 128, 129, 138–140
ODE	ordinary differential equation. 14, 18–20, 23, 24, 27, 72, 73, 88, 104, 108, 114, 118, 151, 173
ODI	*Origin Destination Itinerary* or *Origin Destination Information.* Travel path on a network, described by a collection of flight segments.. 61, 127, 129
Overbooking	The decision to allow more bookings on a flight than there is physical capacity. 54
PNR	*Passenger Name Record.* Record stored in the database of a Computer Reservation System (CRS) containing information about one or more passengers and their travel itinerary. 54
POS	*Point of sale.* Geographical location, sales channel or other criteria that define a specific submarket.. 56, 64, 92, 176
RHS	right-hand side. 14, 17, 19, 21–24, 26, 27, 49, 51, 72, 73, 98, 107, 108, 115, 118, 119, 124, 151, 173
RM	Revenue Management. 2–5, 31, 53–56, 59, 63, 64, 66, 69, 71, 76, 78, 84, 89, 90, 96, 97, 99, 104, 106, 118, 174
SCS	Strict complementary slackness. See Definition 1.1.8. 9, 13
SONCs	Second order necessary conditions for a local optimum. See Theorem 1.1.13. 9
SOS1	Special Ordered Set Type 1. 94, 96

SOSCs	Second order sufficient conditions for a local optimum. See Theorem 1.1.14. 10, 11, 13
Spill	Passengers that are denied boarding at departure as a result of overbooking a flight. 54
Spoilage	Seats left at the time of departure are called *spoiled* seats, because the opportunity to sell them is lost and the seats become worthless. 54
Virtual Capacity	Modified capacity used for optimization purposes in order to allow for a decoupled optimization of overbooking and availability control. 54
Willingness-to-Pay	The maximum amount of money a customer is willing to pay for a certain product.. 42, 43, 54, 56, 60, 86, 97, 99, 100, 175–177
Yield	Amount of revenue the airline expects to get from a customer buying a certain product. This differs from the price the customers pays, because the total price from the customers perspective includes additional taxes and fees that are not received by the airline.. 55

Bibliography

[1] D. Adelman. Dynamic bid-prices in revenue management. *Operations Research*, 55(4):647–661, July/August 2007.

[2] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Computational Geometry*, 7(5-6):265–301, 1997.

[3] P. P. Belobaba. *Air travel demand and airline seat inventory management.* PhD thesis, Massachusetts Institute of Technology, 1987.

[4] P. P. Belobaba. Optimal vs. heuristic methods for nested seat allocation. In *ORSA/TIMS Joint National Meeting*, November 1992.

[5] P. P. Belobaba. Q investigations - algorithms for unrestricted fare classes. PODS Presentation, Amsterdam, 2004.

[6] P. P. Belobaba and C. A. Hopperstad. Algorithms for revenue management in unrestricted fare markets. Presented at the Meeting of the INFORMS Section on Revenue Management, Cambridge, MA, 2004.

[7] M. Ben-Akiva and S. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand.* MIT Press, 1985.

[8] M. Berkelaar, K. Eikland, and P. Notebaert. *lp_solve 5.5, Open source (Mixed-Integer) Linear Programming system*, May 1 2004. Available at ¡http://lpsolve.sourceforge.net/5.5/¿. Last accessed Dec, 18 2009.

[9] M. Berkelaar and others. *lpSolve: Interface to 'Lp_solve' v. 5.5 to Solve Linear/Integer Programs*, 2015. R package version 5.6.13.

[10] D. Bertsekas. *Nonlinear programming.* Athena Scientific, 2003.

[11] D. Bertsekas. *Dynamic programming and optimal control, Volume 2.* Athena Scientific, Belmont, Mass., 3. ed. edition, 2007.

[12] H. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, volume 183 of *Bonner Mathematische Schriften.* Universität Bonn, Bonn, 1987.

[13] J. F. Bonnans and A. Shapiro. Optimization problems with perturbations, a guided tour. *SIAM REVIEW*, 40:228–264, 1996.

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

[15] S. Boyd and L. Vandenberghe. *Convex Optimization.* University Press, Cambridge, 2004.

[16] B. Bradie. *A Friendly Introduction to Numerical Analysis.* Pearson Education. Pearson Prentice Hall, 2005.

[17] U. Brandt-Pollmann. *Numerical solution of optimal control problems with implicitly defined discontinuities with applications in engineering.* PhD thesis, Universität Heidelberg, 2004.

[18] J. W. Brehm and A. R. Cohen. Re-evaluation of choice alternatives as a function of their number and qualitative similarity. *Journal of Abnormal Psychology*, 58(3):373–378, 1959.

[19] J. J. M. Bront, I. Mndez-Daz, and G. Vulcano. A column generation algorithm for choice-based network revenue management. *Operations Research*, 57(3):769–784, 2009.

[20] L. Brouwer. Über Abbildung von Mannigfaltigkeiten. *Mathematische Annalen*, 71(1):97–115, 1911.

[21] A. Bryson and Y.-C. Ho. *Applied Optimal Control*. Wiley, New York, 1975.

[22] C. Büskens and H. Maurer. SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control. *Journal of Computational and Applied Mathematics*, 120:85–108, 2000.

[23] J. Butcher. *The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta and General Linear Methods*. Wiley, 1987. ISBN 0-471-91046-5 (paperback).

[24] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, September 1995.

[25] J. M. Chaneton and G. Vulcano. Computing bid-prices for revenue management under customer choice behavior. *System*, 13(4):452–470, 2009.

[26] R. W. Chaney. Piecewise functions in nonsmooth analysis. *Nonlinear Analysis: Theory, Methods & Applications*, 15(7):649 – 660, 1990.

[27] I. Charpentier. Checkpointing schemes or adjoint codes: Application to the meteorological model Meso-NH. *SIAM Journal on Scientific Computing*, 22(6):2135–2151, 2001.

[28] M. Chen and Z.-L. Chen. Recent developments in dynamic pricing research: Multiple products, competition, and limited demand information. *Production and Operations Management*, 24(5):704–731, 2015.

[29] C. Cizaire and P. Belobaba. Joint optimization of airline pricing and fare class seat allocation. *Journal of Revenue and Pricing Management*, 12(1):83–93, Jan 2013.

[30] R. Cools and A. Haegemans. Algorithm 824: Cubpack: a package for automatic cubature; framework description. *ACM Transactions on Mathematical Software*, 29(3):287–296, 2003.

[31] W. L. Cooper. Asymptotic behavior of an allocation poliy for revenue management. *Operations Research*, 50(4):720–722, 2002.

[32] J.-P. Cote, P. Marcotte, and G. Savard. A bilevel modelling approach to pricing and fare optimisation in the airline industry. *Journal of Revenue and Pricing Management*, 2(1):23–36, 2003.

[33] D. P. De Farias and B. V. Roy. Approximate linear programming for average-cost dynamic programming. *Operations Research*, 51(6):850–856, 2003.

[34] T. Domencich and D. McFadden. *Urban Travel Demand — A behavioral Analysis*. North-Holland Publishing Co., Amstardam, 1975.

[35] D. Eddelbuettel. *Seamless R and C++ Integration with Rcpp*. Springer, New York, 2013. ISBN 978-1-4614-6867-7.

[36] A. Fiacco and J. Kyparisis. Sensitivity analysis in nonlinear programming under second order assumptions. In A. Bagchi and H. T. Jongen, editors, *Systems and Optimization*, volume 66 of *Lecture Notes in Control and Information Sciences*, pages 74–97. Springer Berlin Heidelberg, 1985.

[37] A. Fiacco and G. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, New York, 1968.

[38] A. V. Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10(1):287–311, 1976.

[39] T. Fiig, U. Cholak, M. Gauchet, and B. Cany. What is the role of distribution in revenue management? – past and future. *Journal of Revenue and Pricing Management*, 14(2):127–133, Mar 2015.

[40] T. Fiig, O. Goyons, R. Adelving, and B. Smith. Dynamic pricing – the next revolution in RM? *Journal of Revenue and Pricing Management*, 15(5):360–379, Oct 2016.

[41] T. Fiig, K. Isler, C. Hopperstad, and P. P. Belobaba. Optimization of mixed fare structures: Theory and applications. *Journal of Revenue and Pricing Management*, 9(1/2):152–170, 2010.

[42] T. Fiig, K. Isler, C. Hopperstad, and R. Cleaz-Savoyen. DAVN-MR: A unified theory of O&D optimization in a mixed network with restricted and unrestricted fare products. AGIFORS Reservations and Yield Management Study Group Meeting, Cape Town, South Africa, May 2005.

[43] A. Filippov. Differential equations with discontinuous right hand side. *AMS Transl.*, 42:199–231, 1964.

[44] K. Fukuda. *cddlib*. Institute for Operations Research and Institute of Theoretical Computer Science, ETH Zurich, 2005.

[45] L. Gallaher. A multidimensional Monte Carlo quadrature with adaptive stratified sampling. *Communications of the ACM*, 16(1):4950, 1973.

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

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

[48] G. Gallego and R. Wang. Multiproduct price optimization and competition under the nested logit model with product-differentiated price sensitivities. *Operations Research*, 62(2):450–461, 2014.

[49] M. Gerdts. *Optimal Control of Ordinary Differential Equations and Differential-Algebraic Equations*. Habilitation, University of Bayreuth, 2006.

[50] I. Gladwell. Initial value routines in the NAG library. *ACM Trans. Math. Softw.*, 5(4):386–400, December 1979.

[51] J. Grammig, R. Hujer, and M. Scheidler. Discrete choice modelling in airline network management. *Journal of Applied Econometrics*, 20(4):467–486, 2005.

[52] W. H. Greene. *Econometric Analysis*. Prentice Hall, Upper Saddle River, NJ, 5. edition, 2003.

[53] A. Griewank. Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation. *Optimization Methods and Software*, 1:35–54, 1992.

[54] A. Griewank and A. Walther. Algorithm 799: revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software*, 26(1):19–45, 2000.

[55] J. C. Guo. Estimation of sell-up potential in airline revenue management systems. Master's thesis, Massachusetts Institute of Technology, Operations Research Center, 2008.

[56] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Monographs on Applied Probability and Statistics. Springer Netherlands, 1964.

[57] J. M. Hammersley and D. C. Handscomb. Monte Carlo methods. *Annals of Statistics*, 44(247):335–341, 1964.

[58] R. Hartl, S. Sethi, and R. Vickson. A survey of the maximum principles for optimal control problems with state constraints. *SIAM Review*, 37:181–218, 1995.

[59] H.-Z. He. Analysis of a joint pricing and seat allocation model in a hub-to-hub airline network. *Journal of the Operations Research Society of China*, 4(3):309–333, Sep 2016.

[60] M. K. Horn. Scaled Runge-Kutta algorithms for handling dense output. Technical report, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt, Oberpfaffenhofen, West Germany, 1981.

[61] M. K. Horn. Fourth- and fifth-order, scaled Runge–Kutta algorithms for treating dense output. *SIAM Journal on Numerical Analysis*, 20:558–568, 1983.

[62] IATA. Enhanced airline distribution. Technical report, IATA, https://www.iata.org/whatwedo/airline-distribution/ndc/Documents/ndc-resolution-787.pdf, 2012.

[63] K. Isler, H. Imhof, and M. Reifenberg. From seamless availability to seamless quote. AGIFORS Reservations and Yield Management Study Group Meeting, Cape Town, South Africa, May 2005.

[64] S. S. Iyengar and M. R. Lepper. When choice is demotivating: Can one desire too much of a good thing? *Journal of Personality and Social Psychology*, 79(6):995–1006, 2000.

[65] R. Janin. Directional derivative of the marginal function in nonlinear programming. *Mathematical Programming*, 21:110–126, 1984.

[66] K. Jittorntrum. Solution point differentiability without strict complementarity in nonlinear programming. In A. Fiacco, editor, *Sensitivity, Stability and Parametric Analysis*, volume 21 of *Mathematical Programming Studies*, pages 127–138. Springer Berlin Heidelberg, 1984.

[67] W. Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, Department of Mathematics, University of Chicago, 1939.

[68] A. Kocabıyıkouglu, I. Popescu, and C. Stefanescu. Pricing and revenue management: The value of coordination. *Management Science*, 60(3):730–752, 2014.

[69] M. Kojima. Strongly stable stationary solutions in nonlinear programs. In S. M. Robinson, editor, *Analysis and computation of fixed points*, pages 93–138. Academic Press, 1980.

[70] H. Kuhn and A. Tucker. Nonlinear programming. In J. Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, 1951. University of California Press.

[71] D. Kuksov and J. M. Villas-Boas. When more alternatives lead to less choice. *Marketing Science*, 29(3):507–524, 2009.

[72] M. Kutta. Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. *Zeitschrift für Mathematik und Physik*, 46:435–453, 1901.

[73] A. Kuyumcu and A. Garcia-Diaz. A polyhedral graph theory approach to revenue management in the airline industry. *Computers & Industrial Engineering*, 38(3):375 – 395, 2000.

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

[75] G. P. Lepage. A new algorithm for adaptive multidimensional integration. *Journal of Computational Physics*, 27:192–203, 1978.

[76] G. P. Lepage. Vegas: An adaptive multidimensional integration program. *Cornell Preprint CLNS*, 1980.

[77] G. Li, P. Rusmevichientong, and H. Topaloglu. The d-level nested logit model: Assortment and price optimization problems. *Operations Research*, 63(2):325–342, 2015.

[78] K. Littlewood. Forecasting and control of passenger bookings. *Proceedings of the AGIFORS Symposium*, pages 95–128, 1972.

[79] R. D. Luce. *Individual Choice Behavior: A theoretical analysis*. Wiley, New York, reprinted by dover publications edition, 1959.

[80] R. D. Luce. The choice axiom after twenty years. *Journal of Mathematical Psychology*, 15(3):215 – 233, 1977.

[81] R. D. Luce and P. Suppes. Preference, utility, and subjective probability. In R. D. Luce, R. Bush, and E. Galanter, editors, *Handbook of Mathematical Psychology*, volume 3, chapter 19, pages 249–410. John Wiley and Sons, Inc., New York, 1965.

[82] K. Malanowski and H. Maurer. Sensitivity analysis for parametric control problems with control-state constraints. *Computational Optimization and Applications*, 5(3):253–283, 1996.

[83] O. Mangasarian and S. Fromovitz. Fritz John necessary optimality conditions in the presence of equality and inequality constraints. *Journal of Mathematical Analysis and Applications*, 17:37–47, 1967.

[84] C. F. Manski and D. McFadden. *Structural analysis of discrete data and econometric applications*. The MIT Press, 1981.

[85] A. Marshall. *Principles of economics*. Macmillan, 1895.

[86] A. W. Marshall. The use of multistage sampling schemes in Monte Carlo computations. In *Symposium on Monte Carlo methods, University of Florida, 1954*, pages 123–140. John Wiley and Sons, Inc., New York, 1956.

[87] H. Maurer, C. Büskens, J. Kim, and Y. Kaya. Optimization methods for the verification of second-order sufficient conditions for bang-bang controls. *Optimal Control Methods and Applications*, 26:129–156, 2005.

[88] H. Maurer and N. Osmolovskii. Second order sufficient conditions for time-optimal bang-bang control. *SIAM Journal on Control and Optimization*, 42:2239–2263, 2004.

[89] H. Maurer and H. Pesch. Solution differentiability for nonlinear parametric control problems. *SIAM Journal on Control and Optimization*, 32(6):1542–1554, 1994.

[90] H. Maurer and H. Pesch. Solution differentiability for parametric nonlinear control-problems with control-state constraints. *Journal of Optimization Theory and Applications*, 86:285–309, 1995.

[91] H. Maurer and G. Vossen. Sufficient conditions and sensitivity analysis for optimal bang-bang control problems with state constraints. In *Proceedings of the 23rd IFIP Conference on System Modeling and Optimization, Cracow, Poland*, pages 82–99. Springer Verlag, July 2007.

[92] D. McFadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, *Frontiers in Econometrics*, chapter 4, pages 105–142. Academic Press, 1974.

[93] D. McFadden. Consumer behavior econometric models for probabilistic choice among products. *Interfaces*, 53(3):1329, 2009.

[94] P. McMullen. The maximal number of faces of a convex polytope. *Mathematika*, 17:179–184, 1970.

[95] T. S. Motzkin, H. Raiffa, G. L. Thompson, R. M. Thrall, H. W. Kuhn, and A. W. Tucker. The double description method. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the theory of games*, volume 2, pages 81–103. Princeton University Press, 1953.

[96] D. F. Otero and R. Akhavan-Tabatabaei. A stochastic dynamic pricing model for the multi-class problems in the airline industry. *European Journal of Operational Research*, 242(1):188 – 200, 2015.

[97] Z. Pang, O. Berman, and M. Hu. Up then down: Bid-price trends in revenue management. *Production and Operations Management*, 24(7):1135–1147, 2015.

[98] J. Pinnell and S. Englert. The number of choice alternatives in discrete choice modeling. In *Proceedings of the Sawtooth Software Conference*, pages 121–153. Sawtooth Software, IncEditor, 1997.

[99] S. Pölt. Back to the roots: New results on leg optimization. In *AGIFORS Reservations and Yield Management Study Group Symposium*, London, 1999.

[100] L. Pontryagin, V. Boltyanski, R. Gamkrelidze, and E. Miscenko. *The Mathematical Theory of Optimal Processes*. Wiley, Chichester, 1962.

[101] W. H. Press and G. R. Farrar. Recursive stratified sampling for multidimensional Monte Carlo integration. *Computers in Physics*, 4:190–195, 1990.

[102] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[103] D. Ralph and S. Dempe. Directional derivatives of the solution of a parametric nonlinear program. *Mathematical Programming*, 70(1-3):159–172, 1995.

[104] S. A. Raza. An integrated approach to price differentiation and inventory decisions with demand leakage. *International Journal of Production Economics*, 164:105 – 117, 2015.

[105] E. Reutskaja and R. M. Hogarth. Satisfaction in choice as a function of the number of alternatives : When goods satiate. *Psychology*, 26(March 2009):197–203, 2009.

[106] S. M. Robinson. Generalized equations and their solutions, part II: Applications to nonlinear programming. In M. Guignard, editor, *Optimality and Stability in Mathematical Programming*, volume 19 of *Mathematical Programming Studies*, pages 200–221. Springer Berlin Heidelberg, 1982.

[107] C. D. T. Runge. Über die numerische Auflösung von Differentialgleichungen. *Mathematische Annalen*, 46(2):167–178, 1895.

[108] P. Schweitzer and A. Seidmann. Generalized polynomial approximations in markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, 1985.

[109] D. Sierag, G. Koole, R. van der Mei, J. van der Rest, and B. Zwart. Revenue management under customer choice behaviour with cancellations and overbooking. *European Journal of Operational Research*, 246(1):170 – 185, 2015.

[110] H. A. Simon. *Models of man: social and rational: mathematical essays on rational human behavior in a social setting*. Wiley, 1957.

[111] B. C. Smith, J. F. Leimkuhler, and R. M. Darrow. Yield management at American Airlines. *Interfaces*, 22(1):8–31, 1992.

[112] B. So. An implementable optimization of airfare structure. *Journal of Revenue and Pricing Management*, Mar 2017.

[113] J. Stoer and R. Bulirsch. *Numerische Mathematik 2*. Springer, Berlin Heidelberg New York, 4th edition, 2000. ISBN 3-540-67644-9.

[114] P. Stumm and A. Walther. New algorithms for optimal online checkpointing. *SIAM Journal on Scientific Computing*, 32(2):836, 2010.

[115] K. Talluri and G. Van Ryzin. A discrete choice model of yield management. *Social Science Research Network*, pages 1–20, 2001.

[116] K. Talluri and G. Van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33, 2004.

[117] K. T. Talluri and G. J. van Ryzin. *The theory and practice of revenue management.* Kluwer Academic Publishers, 2004.

[118] L. L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927.

[119] G. van Ryzin and Q. Liu. On the choice-based linear programming model for network revenue management. Technical report, Columbia Business School, 2004.

[120] B. Vinod. A set partitioning algorithm for virtual nesting indexing using dynamic programming. Technical report, SABRE Decision Technologies, 1989.

[121] B. Vinod. Origin-and-destination yield management. In D. Jenkins, editor, *The Handbook of Airline Economics*, pages 459–468. McGraw-Hill, New York, NY, 1995.

[122] D. Williams. *Probability with martingales.* Cambridge mathematical textbooks. Cambridge University Press, 1991.

[123] E. L. Williamson. *Airline network seat inventory control: Methodologies and revenue impacts.* PhD thesis, Massachusetts Institute of Technology, 1992.

[124] M. Yu. Joint pricing and inventory decisions under resource constraints and price-based substitution. *Journal of Revenue and Pricing Management*, 14(1):19–27, Feb 2015.

[125] D. Zhang. An improved dynamic programming decomposition approach for network revenue management. *Manufacturing & Service Operations Management*, 13(1):35–52, 2010.

[126] D. Zhang and D. Adelman. An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science*, 43(3):381–394, aug 2009.

[127] M. Zhang and P. Bell. Price fencing in the practice of revenue management: An overview and taxonomy. *Journal of Revenue and Pricing Management*, 11(2):146–159, Mar 2012.

Appendix A

Software

To solve the numerical example problems, the algorithms described in this thesis were implemented in software. Because of its convenience in handling data, vector arithmetic and generating visualizations, the statistical programming language `R` [102] was used for the overall framework. Various subroutines with high computational cost were implemented in other, more performance efficient programming languages and integrated using the `Rcpp` package [35], which exposes `C`, `C++` and `Fortran` functions from shared libraries to `R`, making them available to be called in the same way as a regular `R` function.

A number of `R` packages played a critical role in solving the optimization problems as described in the following sections. In addition the following utility packages were used to simplify processing or create plots: `data.table`, `digest`, `foreach`, `ggplot2`, `logging`, `plyr`, `reshape2`, `R.utils`. These packages can be found in the package repository `CRAN` (`https://CRAN.R-project.org`).

A.1 Availability control optimization

A.1.1 Single resource

The solver for the single leg availability control dynamic program described in Chapter 5 is a custom `C` implementation of three explicit ODE solvers of different orders — the explicit Euler method of order one, the second order Heun method, and the fourth order Runge-Kutta method (see Fig. 1.1) — specialized to the problem at hand. The `C` algorithm is wrapped in the custom `R` package `singleLegDP` together with a few helper functions for easier use. It supports solution of the dynamic program itself to compute overall expected and outputs bid prices that can be used in a control scheme for simulation. In addition it can compute a solution to the adjoint equation and use this to provide a sensitivity analysis of the results w.r.t. expected demand and yield per booking class. It uses a checkpointing scheme to store the primal solution (value function) for use in the adjoint computation.

A.1.2 Network

The network linear program (see Eq. (3.53)) is used to generate upper bounds on network revenue and to compute the vector of displacement costs used in the LP-DP decomposition. It is solved using the linear program solver from the `R` package `lpSolve` [9], which interfaces the open source solver `lp_solve` [8].

The LP-DP decomposition (Section 6.2) and the probabilistic decomposition (Section 6.3) are implemented in R and utilize the dynamic program and linear program solvers described in the previous sections.

A.2 Simulation

Random numbers for test scenarios and demand streams as described in Sections 6.4.1 and 6.4.3 were generated using the built-in random number generator of `R` with fixed random seeds.

Simulation of the booking process is an iterative procedure (availability for each request depends on the previous history of the booking process) and therefore cannot be formulated in terms of vector operations. Because explicit loops are relatively slow in `R`, this part was implemented in `C++` to improve performance and made available to call from `R` using `Rcpp`.

A.3 Customer choice

The parameters of the customer choice model which define the distribution of customer preferences and the coefficients in the linear utility function are defined in R. To compute the booking probability for each product in an offer set with given prices and product characteristics, we first determine the linear inequalities that define the relevant set of customers (see Eq. (7.3)) via simple linear algebra in R.

Then a triangulation of this polytope is computed using the package `rcdd`, which is an `R` interface around the `C`-library `CDDLIB` [44] that implements a version of the double description method by Motzkin et al. [95]. Compared to other algorithms the double description method has poor asymptotic worst-case behavior, but it generally performs very well in low dimensions even for highly degenerate inputs.

The booking probability is given by the integral of the joint density function of the customer preferences over this customer set. This integral is evaluated with the `FORTRAN90` library `CUBPACK` [30], which provides integration schemes for higher dimensional integrals over a collection of simplexes of arbitrary dimension. Analogously to ODE-solvers with adaptive step size control, an error estimate is derived from results computed with integration rules of different order and the local relative error is controlled using an adaptive refinement of the integration region. To simplify using `CUBPACK` from `R` an interface was implemented in the custom `R` package `Rcubpack`.

As described in Section 7.1, the gradient of the booking probability w.r.t. product attributes can be computed as an integral over the boundary of the customer set. To do so we again use `rcdd` to compute a triangulation of the boundary and evaluate the integral via `CUBPACK`.

A.4 Pricing optimization

To solve the pricing problem for a given scenario we combine the algorithms described in the previous sections. Our implementation currently is only able to handle single leg problem instances, so that each scenario is defined by a combination of capacity of the flight, the general structure of products (i.e. an abstract definition of product characteristics), and one ore more customer types. Each customer type is defined by a utility function, the distribution of customer preferences, and the (potentially non-constant) arrival rate over the booking horizon.

To evaluate expected revenue and its gradient for a given set of product attributes we first compute in `R` the set of feasible offer sets. Then we compute booking probabilities and their gradient for every combination of offer set, product and customer type as described in Appendix A.3 We then compute demand rates for each product and offer set using a linear combination of the results for the different customer types, weighted with their respective arrival rates.

This input is fed into the single leg dynamic program solver described in Appendix A.1 to compute expected overall revenue and its gradient w.r.t. prices and demand rates.

The optimization problem itself is solved using the `L-BFGS-B` method of Byrd et al. [24] implemented in the `R` function `optim`. It allows box constraints on the control variables and works with a limited memory approximation of the Hessian that is computed using `BFGS` updates. To account for non-convexity of the problem we use a multi-start approach with randomly generated initial values uniformly distributed within the admissible region defined in the scenario.