Demand-oriented Integrated Scheduling for Point-to-Point Airlines

February 22, 2016

Oliver Faust, Jochen Gönsch[2], Robert Klein[3]

[2]Mercator School of Management,
University of Duisburg-Essen, Lotharstraße 65, 47057 Duisburg, Germany
[3]Chair of Analytics & Optimization,
University of Augsburg, Universitätsstraße 16, 86159 Augsburg, Germany

Oliver Faust

Email: okfaust@gmail.com

Jochen Gönsch

Email: jochen.goensch@uni-due.de

Robert Klein (🖂)

Phone: +49 821 598 4150

Fax: +49 821 598 4226

Email: robert.klein@wiwi.uni-augsburg.de

Abstract

Optimizing an airline schedule usually comprises multiple planning stages. These are the choice of flights to offer (schedule design), the assignment of fleets to flight legs (fleet assignment), and the construction of rotations under consideration of maintenance constraints (aircraft maintenance routing). Moreover, the airline must assign crews to all flights (crew scheduling). Traditionally, either these scheduling stages are considered sequentially or an existing schedule is modified in order to cope with the arising complexity issue. More recently, some authors have developed models that integrate adjacent stages. In this paper, outcomes of a research project with airline IT provider Lufthansa Systems are presented. We consider the case of a small to medium-sized point-to-point airline with a homogeneous fleet. Hence, fleet assignment is omitted, which offers the possibility to solve schedule design and aircraft maintenance routing simultaneously. Our approach explicitly accounts for passengers' return flight demand and for marginal revenues declining with increasing seat capacity, hence, anticipating the effects of capacity control in revenue management systems. In order to solve the arising integrated mixed-integer problem, a branch-and-price approach and a column generation-based heuristic have been developed. An extensive numerical study, using data from a major European airline provided by Lufthansa Systems, shows that the presented approaches yield high quality solutions to real-world problem instances within reasonable time.

Keywords: Airline Schedule Design, Aircraft Maintenance Routing, Point-to-Point Airline

1 Introduction

During the last century, major airlines developed sophisticated hub-and-spoke networks to offer passengers connecting flights to travel between as many airports as possible. Then, slightly before the liberalization of 1978, a new type of airline emerged with the advent of Pacific South West and Southwest. These low cost carriers (LCCs) had a clear focus on keeping operations simple and keeping costs down, allowing them to offer cheaper fares than the traditional *network airlines* (see, e.g., Dobruszkes 2006). In the 1990s, the LCC concept spread to the EU (e.g. Ryanair, EasyJet) where they profited from liberalization as well as an abundance of cheap secondary airports created on converted cold-war airfields. In the last decade, LCCs also gained momentum in the Asia Pacific region. Besides issues like short turnaround times, a simple "no frills" product structure and cheap in-house ticketing over the internet, virtually all LCCs cut operating costs by using only one (68% of LCCs) or two (24%) aircraft types (see Groß, Lück, and Schröder 2013). Moreover, most LCCs abstain from sophisticated hub-and-spoke networks and offer simple point-to-point connections, often in niches and between secondary airports (see, e.g., Williams 2001, O'Connell and Williams 2005). As we focus on this key characteristic, we use the term *point-to-point airline* in the following.

The *schedule* is at the heart of each airline's product portfolio. At the same time, it defines a large part of the cost an airline incurs. In the highly competitive environment most airlines face today, it is very important to determine an attractive as well as cost-effective schedule.

At most network airlines, it is thus common practice to use sophisticated Operations Research (OR) approaches. Usually, the airline scheduling problem is decomposed into four planning stages: *schedule design*, *fleet assignment*, *aircraft maintenance routing,* and *crew scheduling*. At the first stage, *schedule design*, the airline decides which flights to offer and optimizes the departure times based on the expected passenger demand. The next stage, *fleet assignment*, specifies the type of aircraft to use on each flight leg. The third stage, *aircraft maintenance routing*, is done separately for each type of aircraft and defines which flights are successively flown by the same aircraft in a *rotation*. The goal is usually to find feasible or cost-minimal rotations such that each flight is covered by exactly one aircraft type. A rotation is feasible if it satisfies certain maintenance requirements. Finally, the last stage, *crew scheduling*, seeks a minimal cost assignment of crews to flights that adheres to labor rules. This planning process is performed by OR-experts who dispose of a vast amount of knowledge resulting from research done in academia as well as at network airlines. Airline IT providers offer scheduling software that can be used more or less "out of the box". Moreover, often only incremental changes to an existing schedule are allowed due to operational and marketing reasons.

However, airline schedule planning is still in its infancy at point-to-point airlines and no specific software solutions exist. Often, only very basic heuristics are employed. They usually combine only two simple flight sequences: flight pairs (also called ping-pong flights or out and back flights; a-b-a) and combined flight pairs (a-b-c-b-a). This simplifies and speeds-up the solution process, but it is also a severe restriction on the solution space leading to suboptimal schedules. The structure of these heuristics is explained in detail in Section 4.2. There, we introduce an extended version of such a heuristic with more flight sequences, which later serves as a benchmark. Moreover, instead of a sophisticated demand forecast, new destinations are usually simply tested. That is, flights are offered for one or half a year, and the airline observes whether there is enough demand. This is largely due to fundamental differences in point-to-point airlines' requirements compared to traditional network airlines (see, e.g., Williams 2001, Lawton 2002, O'Connell and Williams 2005, Groß, Lück, and Schröder 2013). (1) They dispose of a largely homogeneous fleet and fleet assignment can be omitted. (2) Connecting passengers are not considered and the determination of passenger flows does not need to include complex network effects. (3) Point-to-point airlines address a market with a high price-elasticity and marginal revenues decline quickly with increasing seat capacity, resulting in a highly non-linear revenue function. (4) Between most airport pairs, only very few weekly flights are offered. Thus, some customer segments include possible return flights in their purchase decision and this has to be considered in schedule design. (5) The schedule is allowed to change completely from period to period. While the homogeneous fleet (1) and the absence of connecting passengers (2) make the problem easier for point-to-point airlines, other aspects make it harder. Declining marginal revenues (3) require a non-linear revenue function. The consideration of return flights (4) only slightly increases complexity. The possibility to create a completely new schedule (5) requires a completely different approach than the widespread utilization of a mandatory flight list with some optional flights to choose from.

To the best of our knowledge, these aspects have not yet been addressed together in the literature.

In this context, airline IT provider Lufthansa Systems wanted to develop specific OR approaches for point-to-point carriers' schedule planning processes. This paper shares outcomes from the corresponding research project. The main contributions are as follows:

(1) Presentation of the – to the best of our knowledge – first airline schedule planning model for point-to-point airlines. It fully integrates schedule design and aircraft maintenance routing. Enhanced demand modeling allows the consideration of different customer segments with time of day specific return flight demand and nonlinear revenues. The schedule is built from scratch (greenfield approach).

(2) Development of the first exact solution approach for integrated schedule design and aircraft maintenance routing for point-to-point airlines. Our branch-and-price algorithm efficiently solves realistic problem instances with up to 15 airports and 10 aircraft.

(3) Provision of advanced heuristics to obtain high-quality solutions for big instances.

(4) Comparison of revenue improvements, solution times, and insights regarding the tradeoff between modeling accuracy and solution accuracy in a numerical study based on real-world data.

The remainder of this paper is organized as follows. We review the literature in Section 2. In Section 3, we present the model. Section 4 contains the solution approaches developed, i.e. an exact branch-and-price algorithm and heuristics. Section 5 summarizes the numerical study and Section 6 concludes.

2 Literature Review

Literature from three areas is relevant. (1) Throughout the last two decades, many researchers have sought to ameliorate some of the drawbacks of a sequential solution approach by integrating various stages of the planning process. (2) Simultaneously, research on the detailed consideration of demand characteristics emerged. (3) Finally, as the aforementioned areas only consider network airlines, the sparse work on integrated scheduling for charter airlines is most closely related to ours.

2.1 Integration of Planning Stages

Several approaches integrate *schedule design with fleet assignment*. Rexing et al. (2000) enhance a fleet assignment model by allowing the departure times of legs to vary within specified time windows, providing greater flexibility. Lohatepanont and Barnhart (2004) build upon an existing schedule and complement a list of mandatory flights (master flight list) that have to be covered with a list of optional flights. Yan and Tseng (2002) model demand on an origin and destination basis and build the schedule from scratch like we do. Cadarso and Marín (2011) consider robustness to ensure connections in case of delays. Cadarso and Marín (2013) extend their previous model and include, among others, mandatory flights and a minimum average fleet utilization. Pita, Barnhart, and Antunes (2013) develop a linear model that explicitly accounts for congested airports and solve it for a smaller network airline.

Barnhart et al. (1998a) present a model and solution procedure that fully integrates *fleet assignment with aircraft maintenance routing* and consider comfort added when passengers can stay in the plane on a through flight instead of connecting to a different aircraft. El Moudani and Mora-Camino (2000) aim at online decision support and blend a heuristic with Dynamic Programming. Haouari, Aissaoui, and Mansour (2009) as well as Haouari et al. (2011) consider aircraft in a disaggregate way and develop a model with heuristic and exact solution approaches, respectively, for a regional airline. Recently, Liang and Chaovalitwongse (2013) have proposed a new formulation that efficiently generates near-optimal solutions for a medium-sized airline.

Barnhart, Lu, and Shenoi (1998) partially integrate *fleet assignment and crew scheduling*. To maintain tractability, they include an approximation of the crew scheduling problem in the fleet assignment problem. Based on this problem's solution, the full crew scheduling problem is solved.

Cordeau et al. (2001) integrate *aircraft maintenance routing with crew scheduling*. Their model is further enhanced by Mercier, Cordeau, and Soumis (2005) who introduce penalties if a crew has only little time to change aircraft. Cohn and Barnhart (2003) propose a new modeling approach and use variables that represent complete solutions to the aircraft routing problem. Mercier (2008) analyzes different types of cuts that have been proposed in the literature for solving integrated aircraft routing and crew scheduling problems. Weide, Ryan, and Ehrgott (2010) also consider robustness. Weide (2009) additionally develops an exact solution approach. Dück et al. (2012) solve an integrated model heuristically by decomposing it into separate stochastic problems for crews and aircraft.

Some authors extend these considerations to aspects of *schedule design, aircraft maintenance routing, and crew scheduling* by allowing flight legs to be moved within departure time windows. Klabjan et al. (2002) provide more flexibility for crew scheduling while maintaining the feasibility of aircraft routing by adding plane-count constraints to the crew-scheduling problem. However, the crew scheduling and aircraft routing problems are solved sequentially. Mercier and Soumis (2007) present a fully integrated model allowing for flight retiming and develop an efficient solution algorithm.

Various studies on the integration of *fleet assignment, aircraft maintenance routing, and crew scheduling* have been published. Clarke et al. (1996) provide modeling devices for including maintenance and crew considerations into the basic fleet assignment model. Overall maintenance requirements for each type of aircraft and base constraints to fulfill crew rest time requirements are added. Gao, Johnson, and Smith (2009) also incorporate aspects of aircraft maintenance routing, crew scheduling, and operations into a fleet assignment model. Rushmeier and Kontogiorgis (1997) present a formulation of the fleet assignment model that allows aggregated aircraft maintenance and crew considerations. Sandhu and Klabjan (2007) propose a model that completely integrates fleet assignment and crew scheduling and partially integrates aircraft maintenance routing. However, some aircraft maintenance constraints are neglected, which is problematic especially for point-to-point and small hub-and-spoke networks. Papadakos (2009) as well as Sherali, Bae, and Haouari (2013) present models that fully integrate the three stages and solve problem instances relating to major airlines. Cacchiani and Salazar-González (2013) as well as Salazar-González (2014) also fully integrate the stages and heuristically solve their model for a regional airline.

Most relevant to us is work on the integration of *schedule design, fleet assignment, and aircraft maintenance routing* as there are no publications regarding only the integration of schedule design and aircraft maintenance routing. The first paper in this direction is from Desaulniers et al. (1997), who extend the model of Abara (1989) with time windows to include scheduling considerations and develop solution approaches to obtain a daily schedule. Ioachim et al. (1999) consider a planning horizon of one week. Flights are labeled with an identifier and flights with the same identifier must have the same departure time across all days. Both approaches need time windows, usually based on a previous schedule, as inputs and do not allow building a schedule from scratch. Grosche and Rothlauf (2008) develop problem-specific metaheuristics based on threshold acceptance and genetic algorithms.

2.2 Detailed Consideration of Demand

The interaction between demand and supply is a crucial element in the construction of an airline schedule, especially regarding the first two stages, *schedule design* and *fleet assignment*.

The common models make simplifying assumptions about revenues, passenger demands, and network flows to approximate the revenue obtained from each flight leg. Often, only aggregate demand and average fares for the different flight legs are considered. For example, the revenue for each fleet-type/flight leg combination is needed as input for many fleet assignment models, even though this value can be computed exactly for a given leg only after the fleeting of all other legs is determined. Instead, simplifying assumptions provide an estimate for the expected revenue of each leg. The problem with this approach is that it does not accurately incorporate the origin and destination nature of demand and the resulting passenger flows throughout the network. On any given flight leg, passengers with many different origins and destinations as well as tickets in several different fare classes compete for space. Thus, on an operational level, most airlines employ state-of-the-art revenue management (RM) systems to manage inventory and maximize revenues across the whole network by protecting seats for high value customers.

Farkas (1996) demonstrates that RM has a significant effect on the origin-destination passenger flows and the fare class mix in the network. His analyses show that both network flow and stochastic demand should be incorporated in fleet assignment to obtain optimal solutions, but the approaches presented are computationally very demanding.

Kniker (1998) develops the passenger mix model which routes passengers over a flight network. This linear program is the first that includes spill and (partial) recapture of passengers. Passengers are said to be spilled if they cannot be accommodated on their preferred itinerary due to capacity shortage. Partial recapture means that some of the spilled passengers choose another itinerary offered by the same airline. Kniker incorporates the passenger mix model into the classical fleet assignment model (FAM, see e.g., Hane et al. 1995). The resulting itinerary-based fleet assignment model (IFAM) is solvable, but much more difficult to solve than the corresponding fleet assignment model. By comparing models that capture network effects but assume deterministic demand versus stochastic models that ignore network effects, he shows that capturing network effects is often more important than capturing stochastic effects. A model incorporating both network effects and stochastic demand is not given by Kniker.

Lohatepanont (2002) continues the analysis of IFAM and investigates its sensitivity to several assumptions. He shows through empirical testing that (i) IFAM needs only rough estimates of recapture rates, (ii) IFAM outperforms the classical FAM despite only deterministic demand is considered, and (iii) despite its assumption that the airline has full control over the passenger mix, IFAM continues to perform well in a more realistic, less controlled environment. Moreover, both FAM and IFAM are sensitive to demand forecast errors. To strike a balance between FAM and IFAM, Lohatepanont introduces the subnetwork-based fleet assignment model (SFAM). SFAM captures partial network effects but is much more tractable. In addition, several approaches for the integration of fleet assignment and schedule design are proposed. They build upon IFAM and use master/optional flight lists. Major results of the aforementioned PhD theses of

Kniker (1998) and Lohatepanont (2002) have been published in several papers. Results regarding IFAM and SFAM are presented in Barnhart, Kniker, and Lohatepanont (2002) and Barnhart, Farahat, and Lohatepanont (2009), respectively. Lohatepanont and Barnhart (2004) publish results regarding the integration of schedule design and fleet assignment.

Sherali, Bae, and Haouari (2010) also use a list of optional flight legs. But contrary to Lohatepanont (2002), their itinerary-based fleet assignment model directly incorporates multiple fare classes and calculates the number of passengers to accept on each itinerary without considering spill and recapture.

Jacobs, Smith, and Johnson (2008) combine an itinerary-based fleet assignment model with a network flow formulation of a revenue management problem with stochastic itinerary demands in an iterative procedure. In each iteration, the revenue management problem is solved with current capacity and provides updated bid prices which reflect the marginal value of a seat and are used to add a new Benders' revenue cut in the fleet assignment model.

2.3 Charter Airlines

Charter airlines possess some similarities to point-to-point airlines. However, corresponding research is extremely sparse compared to the aforementioned work on network airlines, and highly depends on a number of charter-specific assumptions that reduce problem size and thus allow using standard software.

Erdmann et al. (2001) as well as Kim and Barnhart (2007) both consider airlines with heterogeneous fleets, symmetric, time-insensitive demand and allow at most one connection in an itinerary. Additional assumptions further restrict the number of feasible rotations. Moreover, in Erdmann et al. (2001), passengers are transported only between a set of home airports and a set of airports abroad. The authors do not consider different fare classes. Kim and Barnhart (2007) only consider the airports visited by a plane throughout the day but do not determine exact departure or arrival times. The scheduling problem is further simplified by considering a plane's total flying time via aggregate type classifications that group the legs flown. This allows a much smaller model formulation compared to the common usage of binary decision variables representing rotations.

Ronen (2000) addresses a combined fleet assignment and maintenance routing problem. Given a set of flights to cover, the charter airline seeks a feasible, cost-minimal schedule and flights not covered by its own aircraft are sourced from other airlines. Keskinocak and Tayur (1998) consider an almost identical problem faced by a company that manages time-shared jet aircraft and has to satisfy customer requests.

3 Model Description

The model captures the specific requirements of point-to-point carriers mentioned in Section 1 and was developed in cooperation with our industry partner to ensure a good fit with the real-world requirements and common management guidelines. In the following, we describe the flight network and outline how demand is modeled along with the necessary notation. Then, we present the model formulation.

3.1 Flight Network

The supply side of the airline is determined by its flight schedule. The integration of schedule design and aircraft maintenance routing enables us to guarantee feasibility regarding point-to-point airlines' most important restrictions.

Each aircraft of the homogeneous fleet is assigned to a fleet base, where it stays overnight to allow the performance of maintenance activities. Please note that maintenance requirements are similar for point-to-point airlines and network airlines because they are largely defined by legal requirements that apply to all airlines. We cover activities that are performed overnight on a daily time scale (A-checks, see e.g. Clarke et al. 1996 or Sarac, Batta, and Rump 2006). In the morning, the aircraft successively become available and perform their daily flights. Rotations start and end at the same fleet base and aircraft movements are restricted by operational and regulatory issues. Minimum ground times must be met, the environment may be protected by nightly curfews and congested airports often have slot requirements restricting the maximum number of departures and arrivals in a certain timeframe.

For now, we assume that we know the complete set of feasible rotations that respect the abovementioned requirements. In Section 4.1.1, we generate new rotations online in a branch-and-price framework. Accordingly, the following sets and parameters include data regarding individual rotations as well as information like maintenance time windows that govern the number of aircraft available, and, thus determine whether different rotations can be flown simultaneously.

Sets

N: set of airports.

$FB \subseteq N$: set of fleet bases.

$T \subseteq \mathbb{N}_0$: time. In our numerical study, we consider w.l.o.g. a daily schedule and measure time in minutes, thus $T = \{0, \dots, 1440\}$.

R: set of routes. Between airports A and B, there are two routes: A-B and B-A. For route $r = (r_d, r_a) \in R$, r_d denotes the departure airport and r_a is the arrival airport.

F: set of flight legs. Flight leg $f = (f_d, f_a, f_{dt}, f_{at}) \in F$ departs from airport $f_d \in N$ at time $f_{dt} \in T$ and arrives at $f_a \in N$ at time $f_{at} \in T$.

$T_n^F \subseteq T$: set of possible departure times of flights at airport $n \in N$, restricted e.g. because of curfews.

Ω: set of rotations. A rotation $\omega \in \Omega$ is a sequence of legs $f^i \in F$ that are subsequently flown by the same aircraft, departing from and finally arriving back at the same fleet base: $\omega = [f^1, \dots, f^i, \dots, f^{|\omega|}], f_d^1 = f_a^{|\omega|} \in FB, f_a^i = f_d^{i+1}, f_{at}^i \leq f_{dt}^{i+1}$ $(i \in \{1, 2, \dots, |\omega| - 1\})$.

TW_{fb}^m=$\{1,2, \dots, |TW_{fb}^m|\}$: index set of maintenance time windows at fleet base $fb \in FB$.

$T_{fb,tw^m}^{\Omega} \subseteq T_n^F$: set of departure times of rotations from fleet base $fb \in FB$ in maintenance time window $tw^m \in TW_{fb}^m$.

Parameters

$c_r \in \mathbb{R}_0^+$: cost of flying route $r \in R$.

$cap \in \mathbb{N}$: seating capacity of the single plane type, i.e. number of available seats. In line with industry practice, there is only one cabin type (i.e. economy class, see, e.g., Williams 2001).

$dep_\omega \in FB$: first departure airport in rotation $\omega \in \Omega$.

$start_{\omega} \in T$: departure time of the first flight in rotation $\omega \in \Omega$.

$end_{\omega} \in T$: arrival time of the last flight in rotation $\omega \in \Omega$.

$X_{\omega,f}$: 1 if rotation $\omega \in \Omega$ contains leg $f \in F$, 0 otherwise.

$start_{fb}^{m}(tw^{m}) \in T$: begin of tw^{m}-th maintenance time window ($tw^{m} \in TW_{fb}^{m} \cup \{|TW_{fb}^{m}| + 1\}$) at fleet base $fb \in FB$.

$q_{fb,tw^{m}} \in \mathbb{N}_0$: number of planes belonging to fleet base $fb \in FB$ available in tw^{m}-th maintenance time window ($tw^{m} \in TW_{fb}^{m}$).

$slotTime \in \mathbb{N}$: length of rolling time interval for slot restrictions.

$maxDep_{n} \in \mathbb{N}_0$: maximum number of departures from airport $n \in N$ during time interval $slotTime$.

Please note that the assumption that rotations start and end at the same fleet base follows current practice at point-to-point airlines. It simplifies planning and operations and saves cost. For example, the crews usually live near "their" fleet base and the airline thus does not need to provide hotel accommodation during stays at remote airports. From a theoretical point of view, this assumption is not crucial for the model, although it simplifies it. Relaxing the assumption would necessitate additional flow conservation constraints that ensure a constant number of aircraft at the fleet bases. Likewise, the solution methods presented in Section 4 can also be extended, but would become more computationally intensive as more feasible rotations exist.

3.2 Demand Modeling

On the demand side, the model captures the most important revenue effects of a point-to-point airline to provide a profit-maximizing flight schedule. The airline offers single-leg return flights to two types of customers, business and leisure customers. While leisure customers' demand is time-insensitive and only relates to a pair of departure and arrival airports, business customers ask for outgoing and return flights in specific time windows. Thus, for each airport pair, we have w.l.o.g. one leisure customer segment and multiple business customer segments, referring to different demand time windows for the flights. For example, think of business customers who want to depart from A to B in the morning and return in the afternoon (business customer segment 1), who depart in the morning and return in the evening (business customer segment 2), a.s.o. However, even in one segment, there are customers with different valuations for the flight. Airlines' operational RM systems exploit this heterogeneity and prefer customers with higher valuations (who are willing to pay more) if capacity is scarce. Thus, airlines typically observe a non-linear, concave revenue function with marginal revenues decreasing in the number of seats allocated to a customer segment. We capture this by a piecewise-linear approximation (see Section 5.1 for the specific revenue function used in the numerical study). Note that modeling time-sensitive leisure customers is only a matter of input data and possible via additional "business" customer segments instead of leisure customer segments.

The following notation is used to describe demand:

Sets

S^B: set of business customer segments.

S^L: set of leisure customer segments.

$TW^d = \{1,2,\ldots,|TW^d|\}$: index set of demand time windows.

$SP = \{0,1,\ldots,|SP|-1\}$: index set of sampling points for the piecewise-linear revenue approximation. To ease notation, we use the same number of sampling points for all segments. Hence, the interval between the sampling points $p-1$ and p ($p \in SP\backslash\{0\}$) is considered as the p-th interval of the approximated, piecewise-linear revenue function.

Parameters

$start^d(tw^d) \in T$: begin of tw^d-th demand time window ($tw^d \in TW^d \cup \{|TW^d|+1\}$).

$sp^B_{s,p} \in \mathbb{R}^+_0$: p-th sampling point for the linearization of business customer segment s's revenue function ($p \in SP, s \in S^B$).

$sp^L_{s,p} \in \mathbb{R}^+_0$: p-th sampling point for the linearization of leisure customer segment s's revenue function ($p \in SP, s \in S^L$).

$y^B_{s,p} \in \mathbb{R}^+_0$: average revenue in the p-th interval between sampling points $p-1$ and p ($p \in SP\backslash\{0\}$) of business customer segment s's revenue function ($s \in S^B$).

$y^L_{s,p} \in \mathbb{R}^+_0$: average revenue in the p-th interval between sampling points $p-1$ and p ($p \in SP\backslash\{0\}$) of leisure customer segment s's revenue function ($s \in S^L$).

ϕ^B_{s,r,tw^d}: 1 if business customer segment $s \in S^B$ requires route $r \in R$ in demand time window $tw^d \in TW^d$, 0 otherwise.

$\phi^{L}_{s,r}$: 1, if leisure customer segment $s \in S^L$ requires route $r \in R$ (not time-specific), 0 otherwise.

The parameters described above are used in the model to specify a piecewise linear revenue function. This can be the real revenue function or an approximation of an arbitrary revenue function. There is no further requirement besides nonincreasing marginal revenues. In our computational experiments in Section 5, this piecewise linear function approximates the real logit demand function. Figure 1 illustrates the piecewise linear approximation of total revenue for business customer segment $s = 1$ with $|SP| - 1 = 3$ passenger intervals. Note that we approximate total revenue only until its maximum because selling more seats would lead to a lower total revenue.

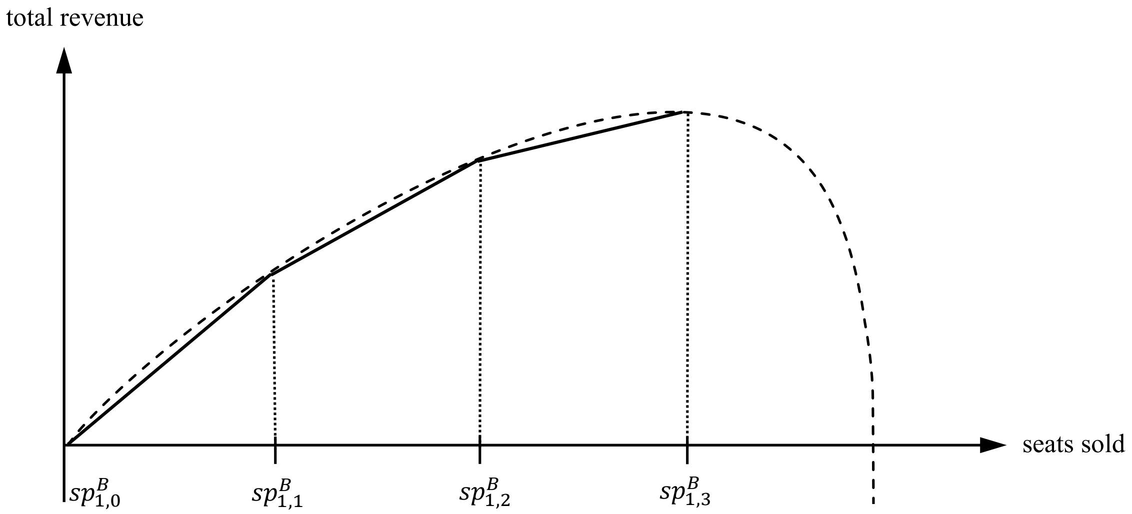

Figure 1: Approximation of total revenue obtained from business customer segment $s = 1$ with $|SP| = 4$ sampling points (illustration)

3.3 Model formulation

Next, we define the decision variables of the model:

$z^B_{s,p} \in \mathbb{R}^+_0$: seating capacity allocated to business customer segment $s \in S^B$ in the p-th interval between sampling points $p - 1$ and p ($p \in SP\backslash\{0\}$).

$z^L_{s,p} \in \mathbb{R}^+_0$: seating capacity allocated to leisure customer segment $s \in S^L$ in the p-th interval between sampling points $p - 1$ and p ($p \in SP\backslash\{0\}$).

x_ω: 1 if rotation $\omega \in \Omega$ is flown, 0 otherwise.

Abbreviations

To ease notation and improve readability, we use the following sets of flights:

$F_r = \{f \in F: f_d = r_d \wedge f_a = r_a\}$: set of flights on route $r = (r_d, r_a) \in R$.

$F_{tw^d} = \{f \in F: f_{dt} \geq start^d(tw^d) \wedge f_{dt} < start^d(tw^d + 1)\}$: set of flights in demand time window $tw_d \in TW^d$.

$F_{n,t^F,slotTime} = \{f \in F: f_d = n \wedge f_{dt} \geq t^F \wedge f_{dt} \leq t^F + slotTime\}$: set of flights that depart from airport $n \in N$ in the time interval with length $slotTime$ beginning at $t^F \in T_n^F$.

The Demand-Oriented Integrated Scheduling Model for a point-to-point airline (DOISM) reads

maximize

$$F(\mathbf{x}, \mathbf{z}) = \sum_{p \in SP\setminus\{0\}} \left(\sum_{s \in S^B} y_{s,p}^B z_{s,p}^B + \sum_{s \in S^L} y_{s,p}^L z_{s,p}^L \right) - \sum_{\omega \in \Omega} \sum_{f \in F} X_{\omega,f} x_\omega c_{(f_d, f_a)} \quad (1)$$

subject to

$$\sum_{p \in SP\setminus\{0\}} \left(\sum_{s \in S^B} \sum_{tw^d \in TW^d} \phi_{s,r,tw^d}^B z_{s,p}^B + \sum_{s \in S^L} \phi_{s,r}^L z_{s,p}^L \right) \leq \sum_{\omega \in \Omega} \sum_{f \in F_r} X_{\omega,f} x_\omega cap \quad \forall r \in R \quad (2)$$

$$\sum_{p \in SP\setminus\{0\}} \sum_{s \in S^B} \phi_{s,r,tw^d}^B z_{s,p}^B \leq \sum_{\omega \in \Omega} \sum_{f \in F_r \cap F_{tw^d}} X_{\omega,f} x_\omega cap \quad \forall r \in R, tw^d \in TW^d \quad (3)$$

$$\sum_{\substack{\omega \in \Omega: dep_\omega = fb \\ \wedge\, start_\omega \leq t^\Omega \\ \wedge\, end_\omega > t^\Omega}} x_\omega \leq q_{fb,tw^m} \quad \forall fb \in FB, tw^m \in TW_{fb}^m, t^\Omega \in T_{fb,tw^m}^\Omega \quad (4)$$

$$\sum_{\omega \in \Omega} \sum_{f \in F_{n,t^F,slotTime}} X_{\omega,f} x_\omega \leq maxDep_n \quad \forall n \in N, t^F \in T_n^F \quad (5)$$

$$0 \leq z_{s,p}^B \leq sp_{s,p}^B - sp_{s,p-1}^B \quad \forall s \in S^B, p \in SP\setminus\{0\} \quad (6)$$

$$0 \leq z_{s,p}^L \leq sp_{s,p}^L - sp_{s,p-1}^L \quad \forall s \in S^L, p \in SP\setminus\{0\} \quad (7)$$

$$x_\omega \in \{0,1\} \quad \forall \omega \in \Omega \quad (8)$$

The objective function (1) reflects profit. The first part is the sum of the revenues obtained from business and leisure customers. For each customer type and segment s, total revenue is obtained by summing up the product of marginal revenue $y_{s,p}^B, y_{s,p}^L$ and the number of passengers $z_{s,p}^B, z_{s,p}^L$ in every interval p between the sampling points $p-1$ and p ($p \in SP\setminus\{0\}$) of the piecewise linear revenue function. The second part subtracts the cost of flying the selected rotations. This cost is calculated by summing over all rotations ω and flight legs f. Remember that the binary variable x_ω is 1 if rotation ω is flown and the binary parameter $X_{\omega,f}$ indicates whether rotation ω contains flight leg f. Constraints (2) ensure that total allocated capacity on each route does not exceed available capacity. On the left hand side, the number of passengers travelling on route r is calculated. Therefore, we sum over all intervals, customer segments and (for business customers) time windows. Remember that the binary parameters ϕ^B_{s,r,tw^d} and $\phi^L_{s,r}$ indicate whether a segment requires a route and the decision variables $z^B_{s,p}$ and $z^L_{s,p}$ denote allocated capacity. Note that demand time windows are not considered here. This is necessary only for business customer segments. Accordingly, constraints (3) have the same structure as (2) but now ensure that the capacity allocated to business customers on each route in each time window does not exceed the respective capacity. Together, these two groups of constraints ensure that every business customer gets a flight in his demand time window and every customer gets a flight. By using this so-called surrogate formulation, we avoid to explicitly consider the demand time windows for leisure segments, which would require additional distribution variables (see, e.g., Gönsch and Steinhardt 2015). Constraints (4) ensure for every fleet base, maintenance time window, and departure time that the number of aircraft used does not exceed the number of aircraft available. Similarly, constraints (5) express the slot restrictions in every timeframe of length $slotTime$ for every airport and departure time. Constraints (6) and (7) ensure for business and leisure customer segments, respectively, that capacity allocations are in line with the intervals of the piecewise linear revenue function. Finally, w.l.o.g., binary requirements on the variables x_ω are imposed (8).

In our solution methods (Section 4), we use the LP relaxation of DOISM obtained by substituting the integrality constraint (8) with

$$0 \leq x_\omega \leq 1 \qquad \forall \omega \in \Omega \quad (8')$$

Now, the dual of the LP (1) – (8') is given by (9) – (13), where the dual variables correspond to the constraints (2) – (8') as follows:

Constraint (2): u^1_r $\quad \forall\, r \in R.$

Constraint (3): u^2_{r,tw^d} $\quad \forall\, r \in R, tw^d \in TW^d.$

Constraint (4): u^3_{fb,tw^m,t^Ω} $\quad \forall fb \in FB, tw^m \in TW^m_{fb}, t^\Omega \in T^\Omega_{fb,tw^m}.$

Constraint (5): u^4_{n,t^F} $\quad \forall n \in N, t^F \in T^F_n.$

Constraint (6): $u^5_{s,p}$ $\quad \forall s \in S^B, p \in SP\backslash\{0\}.$

Constraint (7): $u^6_{s,p}$ $\quad\forall s \in S^L, p \in SP\backslash\{0\}$.

Constraint (8'): u^7_ω $\quad\forall \omega \in \Omega$.

minimize

$$FD(\mathbf{u}) = \sum_{fb\in FB}\sum_{tw^m\in TW^m_{fb}}\sum_{t^\Omega\in T^\Omega_{fb,tw^m}} q_{fb,tw^m}u^3_{fb,tw^m,t^\Omega} + \sum_{n\in N}\sum_{t^F\in T^F_n} maxDep_n u^4_{n,t^F} + \sum_{p\in SP\backslash\{0\}}\sum_{s\in S^B}\left(sp^B_{s,p} - sp^B_{s,p-1}\right)u^5_{s,p} + \sum_{p\in SP\backslash\{0\}}\sum_{s\in S^L}\left(sp^L_{s,p} - sp^L_{s,p-1}\right)u^6_{s,p} + \sum_{\omega\in\Omega} u^7_\omega \quad (9)$$

subject to

$$-\sum_{f\in F} X_{\omega,f} cap\, u^1_{(f_d,f_a)} - \sum_{tw^d\in TW^d}\sum_{f\in F_{tw^d}} X_{\omega,f} cap\, u^2_{(f_d,f_a),tw^d} + \sum_{tw^m\in TW^m_{dep_\omega}}\sum_{\substack{t^\Omega\in T^\Omega_{dep_\omega,tw^m}: \\ start_\omega\le t^\Omega \\ \wedge\, end_\omega > t^\Omega}} u^3_{dep_\omega,tw^m,t^\Omega} + \sum_{n\in N}\sum_{t^F\in T^F_n}\sum_{f\in F_{n,t^F,slotTime}} X_{\omega,f}u^4_{n,t^F} + u^7_\omega \ge -\sum_{f\in F} X_{\omega,f}c_{(f_d,f_a)} \quad \forall\omega\in\Omega \quad (10)$$

$$\sum_{r\in R}\sum_{tw^d\in TW^d}\phi^B_{s,r,tw^d}u^1_r + \sum_{r\in R}\sum_{tw^d\in TW^d}\phi^B_{s,r,tw^d}u^2_{r,tw^d} + u^5_{s,p} \ge y^B_{s,p} \quad \forall s\in S^B, p\in SP\backslash\{0\} \quad (11)$$

$$\sum_{r\in R}\phi^L_{s,r}u^1_r + u^6_{s,p} \ge y^L_{s,p} \quad \forall s\in S^L, p\in SP\backslash\{0\} \quad (12)$$

$$u^1_r,\ u^2_{r,tw^d},\ u^3_{fb,tw^m,t^\Omega},\ u^4_{n,t^F},\ u^5_{s^B,p},\ u^6_{s^L,p},\ u^7_\omega \ge 0 \quad \forall n\in N, r\in R, fb\in FB, tw^d\in TW^d, tw^m\in TW^m_{fb}, t^\Omega\in T^\Omega_{fb,tw^m}, t^F\in T^F_n, s^B\in S^B, s^L\in S^L, p\in SP\backslash\{0\}, \omega\in\Omega \quad (13)$$

4 Solution Methods

The model given in the previous section contains a very large number of binary variables. For example, in one of our networks with only 5 airports and one fleet base, already 79,975,740 possible rotations exist. To overcome this drawback, such models are typically solved using branch-and-price algorithms. In Subsection 4.1, we develop a suitable branch-and-price algorithm and describe the corresponding subproblem with its solution approach as well as the branching rule used. In Subsection 4.2, we develop two heuristic approaches.

4.1 Branch-and-Price

Branch-and-price is an extension of the well-known branch-and-bound approach that works without explicitly enumerating a large number of variables. As in branch-and-bound methods, an LP relaxation is solved at each node to generate dual bounds on the optimal solution value and determine the problems considered at potential child nodes. In branch-and-price methods, these LPs are solved with a special solution approach called column generation.

Column generation (see e.g. Desrosiers and Lübbecke 2005 for an introduction) is an iterative method that alternately solves a restricted master problem (RMP) and a corresponding subproblem. The RMP is a restriction of the master problem that considers only a subset of the variables. The RMP is solved and its dual solution is used to find variables (columns) with positive reduced profit (as we are maximizing profit). If such variables are found, they are added to the RMP and a new iteration starts. If not, the current solution is optimal for the (unrestricted) master problem. Checking for positive reduced profits can also be seen as verifying dual feasibility and adding violated dual constraints along with the corresponding primal variables. Although branch-and-price and column generation have been successfully applied to areas like transportation and scheduling for decades (see, e.g., Desrosiers, Soumis, and Desrochers 1984, Barnhart and Cohn 2004, as well as, for a survey, Barnhart et al. 1998b), Lübbecke and Desrosiers (2005) note that the implementation is still difficult because of the vast possibilities to tune the components.

We consider the LP relaxation of DOISM the master problem, i.e. problem (1) – (8'). In order to solve it, we start with a subset of the feasible rotations in the RMP. In our numerical experiments in Section 5, these initial rotations are obtained by using the heuristic *IH(4,15)* and then *CGH* described in Section 4.2. Then, we use column generation to dynamically generate additional rotations, that is, variables x_{ω}. This lends itself to the problem structure as the overall schedule and demand implications are considered in the RMP and feasibility of rotations is captured in the subproblem.

4.1.1. Subproblem: Generating new Rotations

The reduced profit $\bar{c}_\omega$ of a rotation $\omega \in \Omega$ is then given by

$$\bar{c}_\omega = -\sum_{f\in F} X_{\omega,f} c_{(f_d,f_a)} + \sum_{f\in F} X_{\omega,f} cap\, u^1_{(f_d,f_a)} + \sum_{tw^d \in TW^d} \sum_{f\in F_{tw^d}} X_{\omega,f} cap\, u^2_{(f_d,f_a),tw^d} - \sum_{tw^m \in TW^m_{dep_\omega}} \sum_{\substack{t^\Omega \in T^\Omega_{dep_\omega,tw^m}: \\ start_\omega \le t^\Omega \\ \wedge\, end_\omega > t^\Omega}} u^3_{dep_\omega,tw^m,t^\Omega} - \sum_{n\in N} \sum_{t^F \in T^F_n} \sum_{f\in F_{n,t^F,slotTime}} X_{\omega,f} u^4_{n,t^F} - u^7_\omega \quad (14)$$

As we consider a maximization problem here, a positive $\bar{c}_\omega$ implies nonoptimality of the current solution and rotation ω can be added to the primal RMP. Analogously, this can be seen as starting with a reduced dual problem (9) – (13) with only a subset of constraints (10). Given the current solution, we now search for a constraint (10) for some rotation ω that is violated. If we find one, it is added. If not, the solution is optimal. Finding a rotation with positive reduced profit (a constraint (10) that is violated for a rotation) corresponds to a longest-path problem (LPP) without resource constraints over a time line network that aims at finding a rotation with the largest reduced profit.

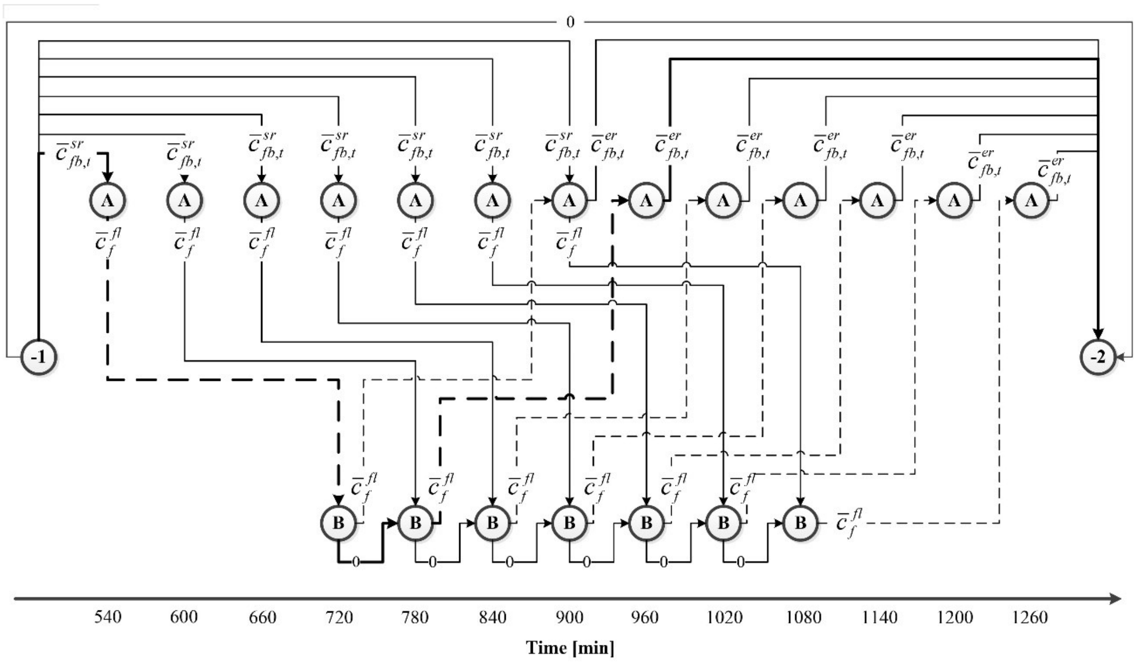

Figure 2: Subproblem network (time discretization: 60 minutes, curfew: 9pm-9am for illustration)

For each fleet base fb, the underlying acyclic network is an implicit representation of all rotations an aircraft belonging to fb can fly. Therefore, it must incorporate all restrictions that determine whether an isolated rotation is feasible, for example minimum ground times and curfews. Such a network is illustrated in Figure 2. To keep the example small, there is only the fleet base (A) and a second (non-fleet base) airport (B). Moreover, we use a time discretization interval of 60 minutes and a curfew from 9pm (1260 min) – 9am (540 min). This network contains four node types: *source, sink*, the *fleet base* fb, and *regular airports* (which may be fleet bases for other aircraft). There is a single source node (-1) and a single sink node (-2) to represent the start and the end of the rotation, respectively. For all airports including fb, there is a node for each point in time when a flight can start or end at this airport. Every feasible rotation $\omega \in \Omega$ of an aircraft belonging to fleet base fb (that is, $dep_\omega = fb$) corresponds to a path from source to sink in the network for fb. The length of the path is the reduced profit $\bar{c}_\omega$ of rotation ω.

The network contains five arc types: *empty rotation, start-of-rotation, end-of-rotation, flight*, and *waiting*:

There is a single empty rotation arc that directly links the source node to the sink. Its profit is 0.

Moreover, there are multiple start-of-rotation arcs that each link the source node to a fleet base node at a point in time $t \in T$ when a rotation can start. Their profit is $\bar{c}^{sr}_{fb,t} = -\sum_{tw^m \in TW^m_{fb}} \sum_{t^\Omega \in T^\Omega_{fb,tw^m}: t \le t^\Omega} u^3_{fb,tw^m,t^\Omega}$. Likewise, there are end-of-rotation arcs from fleet base nodes to the sink with profits $\bar{c}^{er}_{fb,t} = \sum_{tw^m \in TW^m_{fb}} \sum_{t^\Omega \in T^\Omega_{fb,tw^m}: t \le t^\Omega} u^3_{fb,tw^m,t^\Omega}$.

Flight arcs $f = (f_d, f_a, f_{dt}, f_{at})$ correspond to legs and connect two airports. They start at a departure airport f_d at some point in time f_{dt} and arrive at a destination airport f_a at a later point in time f_{at}. Note that in our model, the arrival time f_{at} includes the minimum ground time at the airport f_a. This allows for a combination of two adjacent flight arcs $f_1 = (f_{d_1}, f_{a_1}, f_{dt_1}, f_{at_1})$ and $f_2 = (f_{d_2} = f_{a_1}, f_{a_2}, f_{dt_2} = f_{at_1}, f_{at_2})$ without considering minimum ground times while solving the LPP. The profit of a flight arc f is $\bar{c}^{fl}_f = -c_{(f_d,f_a)} + cap \cdot u^1_{(f_d,f_a)} + cap \cdot u^2_{(f_d,f_a), \max_{tw^d \in TW^d} start^d(tw^d) \le f_{dt}} - \sum_{t^F \in T^F_{f_d}: f_{dt} \ge t^F \wedge f_{dt} \le t^F + slotTime} u^4_{f_d,t^F}$.

Finally, waiting arcs correspond to the idle time between two flights and allow an aircraft to remain at an airport until later. The profit of a waiting arc is 0.

Thus, the reduced profit $\bar{c}_\omega$ of rotation ω is given by

$$\bar{c}_\omega = \bar{c}^{sr}_{dep_\omega, start_\omega} + \sum_{f \in F} X_{\omega,f} \cdot \bar{c}^{fl}_f + \bar{c}^{er}_{dep_\omega, end_\omega} \tag{15}$$

To accelerate the column generation process, we seek to find good feasible solutions early on. Therefore, we start with a rough discretization of time which successively becomes finer. More precisely, when performing column generation, we first use a rough discretization of time in the subproblem's acyclic network. Flight times etc. are adjusted such that each path in this network still represents a feasible rotation in the original problem. For example, assume that a flight from A to B needs 35 minutes. With a discretization interval of 60 minutes, there are flight arcs leaving A at 8am and arriving at B at 9am, leaving A at 9am and arriving at B at 10am, etc. With an interval of 20 minutes, there are arcs from 8am to 8:40am, from 8:20am to 9:00am, etc. When no additional rotations can be generated, the discretization becomes finer and we again try to generate new columns. Only when the finest level is reached and no more columns can be generated, the RMP is solved to optimality and we continue with the next node. In our numerical experiments (Section 5), we start with an interval of 60 minutes, which is successively reduced to 20, 10, and 5 minutes.

4.1.2. Solution Algorithms for the Subproblem

As the network described above can contain positive as well as negative profits, label setting approaches like the popular Dijkstra algorithm (Dijkstra 1959) cannot be applied. Instead, label correcting approaches such as the FIFO algorithm (see, e.g., Pape 1974) may be used. However, to guarantee optimality, every node must be visited. Thus, we primarily use a variant of the FIFO algorithm that terminates as soon as a rotation with positive reduced profit is found (*FIFO-term*). In the numerical experiments (see Section 5), we also briefly evaluate solving the subproblem to optimality with the FIFO algorithm (*FIFO-opt*) and with the standard solver IBM ILOG CPLEX (*CPLEX*).

4.1.3. Branching: Integer Solutions

Compared to branch-and-bound, obtaining integer solutions regarding the binary rotation variables (x_ω) poses some additional challenges. To generate new rotations at the nodes of the search tree, a branching rule that is compatible with the subproblem and its solution procedure is necessary. The standard approach would be to branch on the x_ω

and either fix a rotation into the solution ($x_\omega = 1$) or forbid the use of a rotation ($x_\omega = 0$). It is easy to fix a rotation into the solution. However, forbidding its use is difficult since setting $x_\omega = 0$ only prevents the use of the *already generated* rotation with index ω. But most probably, the original subproblem returns an identical rotation with another index. To avoid this, we would have to modify the subproblem to prevent the generation of forbidden rotations. However, we can usually not simply delete any of the arcs involved since they might be needed by other rotations. As Barnhart et al. (2003) point out in the context of airline crew scheduling, this could require finding the $(k+1)$th shortest path if k pairings have been forbidden by previous branching decisions. Moreover, the number of possible rotations is extremely large.

To overcome these issues, we branch on the flight arcs' flow, that is, the total number of rotations using a flight arc. This guarantees integrality of the arc flows. From the flow decomposition theorem follows that the rotation variables x_ω are integral as well or an equivalent integer solution can easily be constructed (see, e.g., Vanderbeck 2000).

Suppose the arc flow of flight arc $f' = (f'_d, f'_a, f'_{dt}, f'_{at})$ is fractional with value a. Then, we generate two child nodes with the following additional constraints in their RMPs:

Left child: $\sum_{\omega \in \Omega} X_{\omega,f'} x_\omega + y_j^{art} \geq \lceil a \rceil$ (16)

Right child: $\sum_{\omega \in \Omega} X_{\omega,f'} x_\omega - y_j^{art} \leq \lfloor a \rfloor$ (17)

Whereas in branch-and-bound a node can be immediately pruned if the corresponding problem is infeasible, in branch-and-price infeasibility can simply indicate that additional columns must be generated, which in turn requires dual values from the optimal solution with the current columns. Thus, we guarantee feasibility at the child nodes using an artificial variable y_j^{art} in conditions (16) and (17), where j refers to the level in the tree where the condition was added. In the objective function, y_j^{art} is penalized by a "big M cost" (see also Lübbecke and Desrosiers 2005, who discuss adding an artificial variable to all constraints, which is not necessary here).

Regarding the generation of new rotations, the structure of the subproblem remains unchanged. However, we have to take the new dual variable $u^j_{arc\,flows}$ associated with constraint (16) or (17) into account by adding $u^j_{arc\,flows}$ to the profit of the corresponding flight arc.

We now have all building blocks to use branch-and-price. Unfortunately, branching directly on arc flows performed very poor (see also the computational experiments in Section 5) because restricting a flow (condition (17)) added little to the structure of the solution and often the next solution was basically equivalent with just that specific flight starting one point in time earlier or later.

To arrive faster at the structure of an optimal solution, we investigated several alternative branching rules and combinations thereof. The best performance was obtained by first ensuring integrality of the cumulative number of flights on each route $r = (r_d, r_a) \in R$ in each demand time window $tw^d \in TW^d$. If the number of flights on route $r' = (r'_d, r'_a)$ departing in demand time window $tw^{d'}$ is a fraction, we add the following child nodes:

Left child: $\sum_{\omega\in\Omega} \sum_{f\in F_{r'} \cap F_{tw^{d'}}} X_{\omega,f}\, x_\omega + y_j^{art} \geq \lceil a \rceil$ (18)

Right child: $\sum_{\omega\in\Omega} \sum_{f\in F_{r'} \cap F_{tw^{d'}}} X_{\omega,f}\, x_\omega - y_j^{art} \leq \lfloor a \rfloor$ (19)

The corresponding dual variable is denoted by $u^j_{sum\, routes\, tw}$. Regarding the subproblem, this variable is added to the profit of all flight arcs on route r' departing in demand time window $tw^{d'}$. When integrality regarding the cumulative number of flights on each route in each demand time window is attained at a node, we continue pursuing integrality of the arc flows as described above if not all arc flows are integer yet.

In addition to the branching rule described above, we also evaluated eight branching rules with clustering of arc flows ranging from one cluster containing all arc flows over clusters containing only routes/time windows to clusters for each point in time. Moreover, we considered a dozen combinations thereof, each first using a higher aggregation and then successively becoming more granular. The results (not given here) were mostly poor. The second best rule was branching on clusters of time windows before using the rule described above, but was two to five times slower.

4.2 Heuristics

Finally, we present two heuristic approaches. They both belong to a class where column generation is performed offline. That is, a subset of all columns is generated upfront and the RMP is solved to optimality using branch-and-bound over this subset, without generating new rotations.

- The *Column Generation Heuristic (CGH)* builds on the branch-and-price approach and performs column generation only at the root node. It solves the integer program over the columns obtained to optimality with CPLEX. This approach is widely used in academia (see, e.g., Barnhart, Kniker, and Lohatepanont 2002 or Lohatepanont and Barnhart 2004 for applications to related problems).
- The *Industry Heuristic (IH)* mimics current practice and allows incorporating suggestions and know-how from industry. In a first step, intuitive flight sequences are generated. Often only flight pairs (a-b-a) and combined flight pairs (a-b-c-b-a) are used in industry. We additionally consider triangle (a-b-c-a), square (a-b-c-d-a) and pentagonal (a-b-c-d-e-a) flights, where a, b, c, d, and e correspond to different airports. Then, additional flight sequences are generated by combining these initial flight sequences up to a total of τ^{max} legs. For example, we might obtain the sequence a-b-a-b-c-d-a by combining a flight pair and a square flight. Subsequently, rotations are generated by considering the start of every flight sequence at the beginning of each demand time window and each TD^{IH} minutes later. In the example above, we might consider the start of the sequence at 7:00am (the beginning of the first demand time window) and 12:00am (the beginning of the second demand time window), obtaining two rotations. With $TD^{IH} = 120$, we would additionally obtain the rotations starting at 9:00am, 11:00am, 2:00pm, 4:00pm, etc. Finally, DOISM is solved using CPLEX over the rotations generated. Note that the combination of initial flight sequences is done primarily to speed up the heuristic; DOISM can also combine rotations if short ones can be flown one after the other using the same aircraft. We denote this heuristic together with its parameter values as $IH(\tau^{max}, TD^{IH})$. As the length τ^{max} of the initial flight sequences increases and the offset TD^{IH} decreases, more rotations are generated and the heuristic tends to yield better solutions, albeit at the cost of an increased runtime.

5 Computational Experiments

In this section, we describe a series of computational experiments using real-world data. The goal of our experiments is twofold. First, we investigate the performance of the branch-and-price procedure and the two heuristics with regard to solving the linear DOISM model. Second, we consider the influence of the demand function's linearization and analyze the revenues obtained using the underlying non-linear revenue functions.

All experiments were performed on a workstation with two Intel Xeon E5-2690 CPUs at 2.9 GHz and 128 GB of RAM running Windows Server 2008 R2 Enterprise (SP1). We implemented the algorithms in C# with Microsoft .NET Framework Version 5.5.50938 SP1 and used IBM ILOG CPLEX 12.5 to solve linear programs.

5.1 Test Instances

Airline IT provider Lufthansa Systems provided data from a major European point-to-point airline for our experiments. Because of confidentiality issues, we are restricted to reporting only some key parameters. The base setting contains 52 airports, 2 fleet bases, 4 demand time windows, 3 maintenance time windows, a homogeneous fleet with seat capacity of 156 per aircraft, predefined ground times, curfews, and the maximum number of departures per airport, as well as costs and block times (think of flight times) for each pair of two airports. Test instances were generated by sampling from this base setting and varying the number of airports $|N|$, fleet bases $|FB|$, and the total number of aircraft q considering the following parameter values:

- $|N| \in \{5, 10, 15, 20, 30, 40, 52\}$
- $|FB| = 1$ for $|N| = 5, 10$
- $|FB| \in \{1, 2\}$ for $|N| = 15$
- $|FB| = 2$ for $|N| = 20, 30, 40, 52$
- $q \in \{2, 5\}$ for $|N| = 5, 10$
- $q \in \{5, 10\}$ for $|N| = 15, 20, 30$
- $q \in \{10, 15\}$ for $|N| = 40, 52$

We used the popular logit demand function (Phillips 2005, p. 53), which can also be interpreted as a binary logit choice model (Ben-Akiva and Lerman 1985, p. 71). Thus, the total revenue of selling a tickets to customer segment $s \in S^B$ ($s \in S^L$) is given by

$$TR(a) = -a \cdot \frac{-R_s b^s + \ln\left(\frac{a}{MA_s - a}\right)}{b^s} \quad (20)$$

where R_s is usually interpreted as the (overall) market price and MA_s as the potential market size. The price sensitivities of business and leisure customers were $b^B = 0.02$ and $b^L = 0.1$, respectively for $B \in S^B$ and $L \in S^L$.

From (20), we obtain non-linear, concave total revenue curves (see also Figure 1). We linearized these curves between 0 and their maximum using $|SP| \in \{2, 3, 4, 5\}$ equidistantly spaced sampling points. Thus, we obtained 64 test instances to evaluate the exact approaches. For the heuristic approaches, we used $|SP| \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and obtained 144 test instances.

Although the model addresses strategic/tactical planning, in practice there is only limited solution time, because the planning experts usually prefer to work iteratively and are used to calculate several scenarios. We mimicked this by using a time limit of 12 hours and report instances not solved to optimality with the parameter values when the time limit was reached.

5.2 Numerical Results

In this section, we describe the computational results for our solution algorithms. We first examine the performance of the implemented branch-and-price procedures in Subsection 5.2.1. Subsection 5.2.2 contains an analysis of the heuristic solution methods. Finally, the exact and heuristic approaches are compared in Subsection 5.2.3. In Subsection 5.2.4, we investigate the scalability of the heuristics to settings with three fleet bases.

5.2.1. Branch-and-Price

In the following, we investigate the performance of the branch-and-price approach with regard to solving the DOISM model that uses the linearized revenue function. An initial feasible solution is obtained with *IH(4,15)* (Section 4.2) at the root node. The choice of the solution method for the subproblems (Section 4.1.2) had a comparatively small influence. Finding the best rotations to add with *CPLEX* and *FIFO-opt* showed very similar runtimes. Although this exact solution of the subproblem reduced the number of column generation iterations necessary, it was about 20% – 50% slower than using *FIFO-term* to find an arbitrary rotation with positive reduced profit. Thus, we only report results obtained with *FIFO-term*.

Table 1 and Table 2 contain key performance indicators obtained with branch-and-price and the two branching rules described in Section 4.1.3, that is directly branching on arc flows (rule 1) in comparison to clustering the arc flows according to routes and time windows (rule 2). We only show results for $|SP| = 2$ sampling points (Table 1) and $|SP| = 5$ (Table 2) here. To save space, the results are averaged over two underlying problem instances with different numbers of aircraft q. Regarding instances solved to optimality, we observe that the number of nodes in the complete B&P tree is quite small with rule 2. In these nodes, only a comparably small number of rotations is generated. By contrast, using rule 1, the number of nodes explodes. For example, in the first column of Table 1 ($|FB| = 1$, $|N| = 5$), both underlying instances are solved to optimality with rule 2 (instances solved to optimality = 2 out of 2), requiring on average only one node and 126 rotations. Rule 1 solves only one instance (instances solved to optimality = 1 out of 2) while the second is terminated after 12 hours, leading to an average runtime slightly above 6 hours. On average, 132,511 nodes and 576 rotations were generated. A detailed investigation (not shown here) proves that this is because branching according to rule 1 often adds little to the solution structure. If using a flight leg is restricted, usually basically the same solution with just that one specific flight a little bit earlier or later is obtained in the child node. Instances that are aborted due to the time limit must be treated with care. Here, opposite values are observed. Rule 2 still generates fewer rotations, but more nodes in the tree. However, this again confirms rule 2's ability to quickly structure the solution in the nodes through branching. Because of their stronger structure, nodes can be processed quicker generating less (unnecessary) rotations.

There is only a small revenue advantage of rule 2, indicating that also rule 1 usually finds good solutions within the time limit, but fails at proving optimality. Even in problem instances solved to optimality, often only one feasible solution is evaluated. This shows that already the initialization with *CGH* usually finds very good or optimal solutions.

Table 1: Branch-and-price – comparison of branching directly on arc flows (rule 1) and on routes in time windows (rule 2) for $|SP| = 2$, averaged over q

		|FB| = 1			|FB| = 2				
	rule	**|N| = 5**	**|N| = 10**	**|N| = 15**	**|N| = 15**	**|N| = 20**	**|N| = 30**	**|N| = 40**	**|N| = 52**
linearized profit	1	20,760.1	21,715.0	37,214.3	105,895.3	107,853.8	109,131.1	166,371.0	166,763.1
	2	20,760.1	21,715.0	37,214.3	105,896.9	108,030.1	109,131.1	166,371.0	166,763.1
runtime	1	06:00:03	06:00:03	06:00:22	12:00:00	12:00:00	12:00:00	12:00:00	12:00:00
[hh:mm:ss]	2	00:00:04	00:00:04	00:00:12	00:53:12	04:07:46	06:32:43	12:00:00	12:00:00
feasible solutions	1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1
(1st / 2nd instance)	2	1 / 1	1 / 1	1 / 1	1 / 2	1 / 2	1 / 1	1 / 1	1 / 1
instances solved to	1	1 (2)	1 (2)	1 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)
optimality (total)	2	2 (2)	2 (2)	2 (2)	2 (2)	2 (2)	1 (2)	0 (2)	0 (2)
nodes in B&P tree	1	132,511.0	111,947.5	66,882.5	53,329.0	15,598.0	5,835.0	2,542.5	641.5
	2	1.0	1.0	4.0	5,240.0	11,977.0	7,663.0	11,347.0	4,653.0
generated	1	576.0	898.0	1,877.5	4,846.0	6,520.0	10,099.5	13,353.5	16,491.5
rotations	2	126.0	175.5	425.5	2,054.0	3,891.5	7,146.5	8,443.5	11,758.5
max. depth B&P	1	49.0	38.5	41.5	61.5	78.0	90.0	89.0	63.5
	2	0.5	0.5	2.0	27.5	31.0	36.0	43.0	31.5

Table 2: Branch-and-price – comparison of branching directly on arc flows (rule 1) and on routes in time windows (rule 2) for $|SP| = 5$, averaged over q

		|FB| = 1			|FB| = 2				
	rule	**|N| = 5**	**|N| = 10**	**|N| = 15**	**|N| = 15**	**|N| = 20**	**|N| = 30**	**|N| = 40**	**|N| = 52**
linearized profit	1	61,560.6	70,844.1	111,149.5	275,170.8	291,951.2	310,131.0	460,742.5	481,633.2
	2	61,560.6	70,844.1	111,149.5	275,170.8	291,951.2	310,131.0	460,742.5	481,633.2
runtime	1	08:15:24	07:03:11	12:00:00	12:00:00	12:00:00	12:00:00	12:00:00	12:00:00
[hh:mm:ss]	2	00:00:05	00:00:30	00:01:53	12:00:00	12:00:00	12:00:00	12:00:00	12:00:00
feasible solutions	1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1
(1st / 2nd instance)	2	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1
instances solved to	1	1 (2)	1 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)
optimality (total)	2	2 (2)	2 (2)	2 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)
nodes in B&P tree	1	221,544.5	95,963.0	59,637.0	40,440.5	11,761.5	2,553.5	1,161.5	487.0
	2	20.0	60.0	134.0	71,087.0	29,656.5	10,978.5	4,905.5	2,718.0
generated	1	1,557.0	2,349.5	3,007.0	7,462.0	5,741.0	8,093.5	10,190.0	14,204.0
rotations	2	184.5	647.0	1,278.5	3,595.0	4,705.0	5,802.0	7,753.5	8,288.0
max. depth B&P	1	126.5	101.0	115.5	161.0	153.5	75.0	98.0	86.5
	2	6.0	8.5	16.0	53.0	41.0	31.5	43.5	49.5

To illustrate the benefit of branching rule 2 over rule 1, Figure 3 depicts the solution process for a problem instance with $|N| = 10$, $|FB| = 1$, $q = 5$, and $|SP| = 5$. Solving the instance to optimality took more than 2 hours when branching on arc flows (rule 1), and 25,132 nodes with 1,750 columns were generated. The maximum depth of the tree was 87. When branching on routes and time windows (rule 2), the solution time was only 11 seconds, and 36 nodes with 380 columns were generated. The maximum depth was 7.

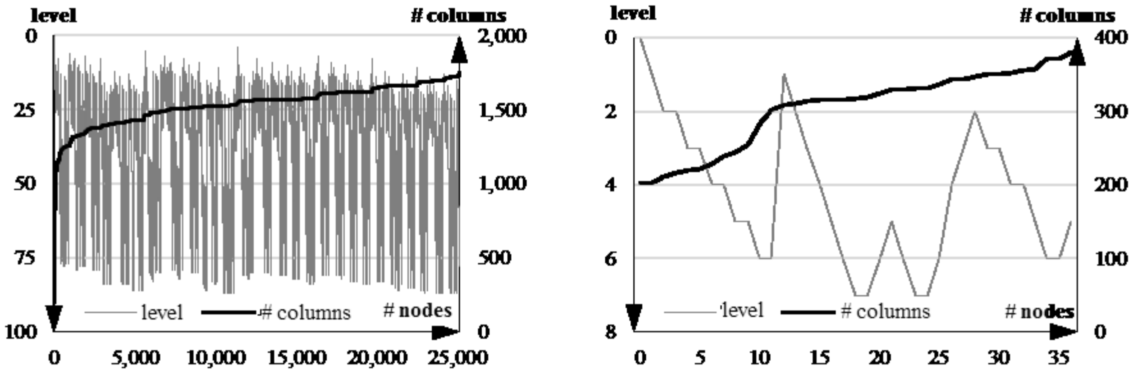

Figure 3: Solution process of an instance with $|N| = 10$, $|FB| = 1$, $q = 5$, $|SP| = 5$: Number of columns generated and level in the branch-and-price tree with branching directly on arc flows (rule 1; left) and on routes in time windows (rule 2; right)

5.2.2. Heuristics

In this subsection, we investigate the heuristics' performance with regard to DOISM with its linearized revenue function. In addition to the *Column Generation Heuristic (CGH)* we also consider four variants of the *Industry Heuristic (IH)*, obtained by combining two values $\tau^{max} = 4, 8$ for the initial flight sequences' maximum length with two values $TD^{IH} = 15, 120$ for the offset to create additional rotations.

Table 3 (Table 4) shows the heuristics' performance for $|SP| = 2$ ($|SP| = 5$) sampling points, averaged over the number of aircraft q. The heuristics *CGH* and *IH(8,15)* yield comparably high revenues, whereas *IH(4,15)*, which combines only two flight pairs, obtains slightly less revenue. When less rotations at later times are generated by increasing the parameter TD^{IH}, *IH(4,120)* and *IH(8,120)* perform much worse. Regarding runtime, *IH(8,15)* is quite slow compared to *CGH* and other variants of *IH*. This is due to the very high number of rotations generated by *IH(8,15)*, which outnumbers *CGH's* by a factor of about 10 to 15. For $|SP| = 2$, $|FB| = 2$, $|N| = 52$, there are two outliers with considerably higher runtimes: *IH(4,15)* and *IH(8,15)*. In both cases, an instance could not be solved by CPLEX within the time limit of 12 hours and was aborted. However, the underlying rotations were generated in a few minutes.

Summing up, we can state that using predefined flight sequences can lead to good results, but needs a very high number of rotations. By contrast, *CGH* obviously succeeds in focusing on relevant rotations as it obtains comparable revenues with considerably less rotations and, thus, is much faster than *IH(8,15)*.

Please note that, although IH mimics current practice, it is considerably more sophisticated. In industry, usually only flight pairs (see Section 4.2) and combined flight pairs are used and only a small number of rotations is generated. Thus, we think that *IH(4,120)* is closest to practice, but already more elaborated as it also considers triangle, square and pentagonal flights.

Table 3: Comparison of heuristics CGH and IH for $|SP| = 2$, averaged over q

		|FB| = 1			|FB| = 2				
	heuristic	**|N| = 5**	**|N| = 10**	**|N| = 15**	**|N| = 15**	**|N| = 20**	**|N| = 30**	**|N| = 40**	**|N| = 52**
linearized profit	*CGH*	20,760.1	21,715.0	37,214.3	105,895.3	107,853.8	109,131.1	166,371.0	166,763.1
	IH(4,120)	18,897.7	21,231.9	28,896.8	87,902.7	92,706.0	93,003.4	133,745.1	138,457.4
	IH(8,120)	18,897.7	21,231.9	28,896.8	87,902.7	92,706.0	93,003.4	134,028.5	138,772.5
	IH(4,15)	20,634.5	21,595.5	34,447.4	104,329.2	106,161.4	106,883.2	158,838.4	159,764.2
	IH(8,15)	20,634.5	21,595.5	34,447.4	104,820.5	106,400.1	107,086.5	158,991.4	159,917.3
runtime [mm:ss]	*CGH*	00:03	00:03	00:07	00:11	00:35	01:37	04:56	11:31
	IH(4,120)	00:01	00:01	00:02	00:03	00:06	00:16	00:29	01:04
	IH(8,120)	00:01	00:01	00:04	00:11	00:17	01:08	01:48	04:47
	IH(4,15)	00:01	00:03	00:09	00:25	00:40	02:15	04:24	373:07
	IH(8,15)	00:02	00:05	00:20	01:23	02:10	10:01	16:22	433:56
generated rotations	*CGH*	116.5	142.5	322.5	648.5	979.5	1,687.5	3,085.0	4,744.0
	IH(4,120)	109.0	321.0	603.0	1,104.0	1,464.0	3,075.0	4,438.0	7,346.0
	IH(8,120)	238.0	518.0	1,248.0	2,990.0	3,745.0	9,005.0	11,296.0	20,530.0
	IH(4,15)	569.0	1,530.0	2,964.0	5,434.0	7,199.0	15,119.0	20,916.0	34,997.0
	IH(8,15)	984.0	2,183.0	4,911.0	11,547.0	14,353.0	33,275.0	41,462.0	73,130.0

Table 4: Comparison of heuristics CGH and IH for $|SP| = 5$, averaged over q

		|FB| = 1			|FB| = 2				
	heuristic	**|N| = 5**	**|N| = 10**	**|N| = 15**	**|N| = 15**	**|N| = 20**	**|N| = 30**	**|N| = 40**	**|N| = 52**
linearized profit	*CGH*	61,560.6	70,844.1	111,149.5	275,170.8	291,951.2	310,131.0	460,742.5	481,633.2
	IH(4,120)	45,694.7	49,870.6	75,009.4	234,398.1	254,315.1	265,652.4	381,120.7	397,928.6
	IH(8,120)	45,694.7	49,870.6	75,009.4	241,648.8	263,503.1	268,862.8	390,756.6	407,882.6
	IH(4,15)	58,094.5	64,732.6	104,279.8	273,779.7	291,354.8	305,397.7	448,699.3	472,711.5
	IH(8,15)	58,094.5	64,732.6	104,279.8	282,738.9	304,628.6	317,029.3	464,359.0	481,488.6
runtime [hh:mm:ss]	*CGH*	00:02	00:04	00:07	00:16	00:34	01:40	03:11	08:33
	IH(4,120)	00:01	00:01	00:02	00:04	00:06	00:16	00:32	01:04
	IH(8,120)	00:01	00:01	00:04	00:12	00:18	01:12	01:51	05:11
	IH(4,15)	00:01	00:04	00:10	00:28	00:42	02:30	04:42	12:31
	IH(8,15)	00:02	00:06	00:21	01:27	02:12	11:04	17:26	106:18
generated rotations	*CGH*	105.0	193.0	352.5	841.0	1,173.5	2,157.0	2,843.5	4,327.0
	IH(4,120)	109.0	321.0	603.0	1,104.0	1,464.0	3,075.0	4,438.0	7,346.0
	IH(8,120)	238.0	518.0	1,248.0	2,990.0	3,745.0	9,005.0	11,296.0	20,530.0
	IH(4,15)	569.0	1,530.0	2,964.0	5,434.0	7,199.0	15,119.0	20,916.0	34,997.0
	IH(8,15)	984.0	2,183.0	4,911.0	11,547.0	14,353.0	33,275.0	41,462.0	73,130.0

5.2.3. Comparison of Exact and Heuristic Solution Approaches

In this subsection, we investigate the performance of the three solution algorithms branch-and-price ($B\&P$), CGH and IH when applying the corresponding solution to the real, non-linearized revenue functions and analyze how it depends on the number of sampling points.

Table 5: Comparison of the exact B&P approach (rule 2) and the heuristics for different numbers of sampling points, averaged over $|N|$, $|FB|$, and q

		# sampling points								
	approach	**|SP|=2**	**|SP|=3**	**|SP|=4**	**|SP|=5**	**|SP|=6**	**|SP|=7**	**|SP|=8**	**|SP|=9**	**|SP|=10**
linearized profit	*B&P rule 2*	91,985	230,911	250,139	257,898	-	-	-	-	-
	CGH	91,963	230,911	250,139	257,898	261,431	262,377	263,738	264,854	265,200
	IH(4,120)	76,855	189,748	205,999	212,999	216,104	217,281	218,192	218,849	219,142
	IH(8,120)	76,930	192,878	210,271	217,904	220,754	222,185	223,137	223,775	224,039
	IH(4,15)	89,082	224,674	245,215	252,381	256,198	257,399	258,394	259,388	259,573
	IH(8,15)	89,237	227,875	251,886	259,669	263,707	265,077	266,392	267,139	267,439
non-linearized profit	*B&P rule 2*	141,964	253,864	257,946	262,585	-	-	-	-	-
	CGH	141,789	253,864	257,946	262,585	264,464	264,818	265,421	266,094	266,292
	IH(4,120)	120,139	206,952	213,004	217,842	218,859	219,403	219,727	220,066	220,094
	IH(8,120)	119,521	210,902	218,134	222,611	223,582	224,423	224,514	224,953	225,014
	IH(4,15)	139,465	245,830	253,162	257,154	259,141	259,811	259,998	260,645	260,608
	IH(8,15)	141,490	250,236	260,693	265,254	266,789	267,636	267,883	268,485	268,481
linearization gap [%]	*B&P rule 2*	37.94%	9.36%	2.78%	1.74%	-	-	-	-	-
	CGH	37.89%	9.36%	2.78%	1.74%	1.17%	0.80%	0.54%	0.47%	0.39%
	IH(4,120)	36.53%	7.92%	3.33%	2.16%	1.26%	0.87%	0.61%	0.51%	0.42%
	IH(8,120)	36.25%	8.15%	3.56%	2.07%	1.26%	0.89%	0.55%	0.49%	0.42%
	IH(4,15)	38.53%	9.09%	3.00%	1.78%	1.13%	0.81%	0.57%	0.51%	0.39%
	IH(8,15)	39.09%	9.41%	3.22%	1.97%	1.14%	0.84%	0.53%	0.53%	0.39%
runtime [hh:mm:ss]	*B&P rule 2*	04:26:55	07:30:37	07:30:31	07:30:31	-	-	-	-	-
	CGH	00:02:23	00:02:19	00:01:46	00:01:48	00:01:59	00:01:47	00:01:38	00:01:55	00:01:47
	IH(4,120)	00:00:15	00:00:16	00:00:16	00:00:16	00:00:16	00:00:16	00:00:17	00:00:17	00:00:17
	IH(8,120)	00:01:02	00:01:06	00:01:04	00:01:06	00:01:08	00:01:09	00:01:08	00:01:10	00:01:11
	IH(4,15)	00:47:38	00:02:46	00:02:49	00:02:39	00:03:55	00:03:33	00:03:24	00:03:27	00:03:56
	IH(8,15)	00:58:02	00:12:57	00:15:25	00:17:22	00:12:55	00:11:19	00:11:16	00:12:16	00:11:20
feasible solutions (avg)	*rule 2*	1.13	1.00	1.00	1.00	-	-	-	-	-
instances solved to optimality (total)	*rule 2*	11 (16)	6 (16)	6 (16)	6 (16)	-	-	-	-	-

The table and figures in this section show data averaged over the number of airports $|N|$ and fleet bases $|FB|$. Table 5 shows linearized/non-linearized profit, the linearization gap, and runtime for $B\&P$ with branching rule 2 and the 5 heuristics with 2 to 10 sampling points. In addition, it contains the number of feasible solutions found and the number of instances with proven optimality for $B\&P$. Values for $B\&P$ are only available up to $|SP| = 5$ because of the high runtime.

Note that even if an instance was solved to optimality with *B&P*, the profit based on the real, non-linearized revenue functions is not necessarily maximal because the linearized revenue function was optimized. However, the table clearly shows that profit increases if more sampling points are used to better approximate the real non-linear revenue function. At the same time, the runtime of *B&P* increases considerably as more and more instances cannot be solved within the time limit. The heuristics' runtimes show no clear trend and seldom exceed a few minutes. An exception are *IH(4, 15)* and *IH(8, 15)* for $|SP| = 2$, where a high average runtime is caused by one instance reaching the time limit.

Figure 4 displays the linearization gap, that is, the difference between the value of the piecewise linear objective function of DOISM and the profit obtained when applying the corresponding solution to the real, non-linearized revenue functions (20). Note that these values cannot be inferred by comparing Table 1 with Table 2 because the figure relates the values in each table to the corresponding non-linear revenue (not given here). The results are strikingly similar for all algorithms. There is a huge gap of up to 40% when using only 2 sampling points, which quickly declines and is close to zero with 5 to 6 sampling points. Moreover, there is not much variation between the test instances and similar results are observed for individual test instances (not shown here).

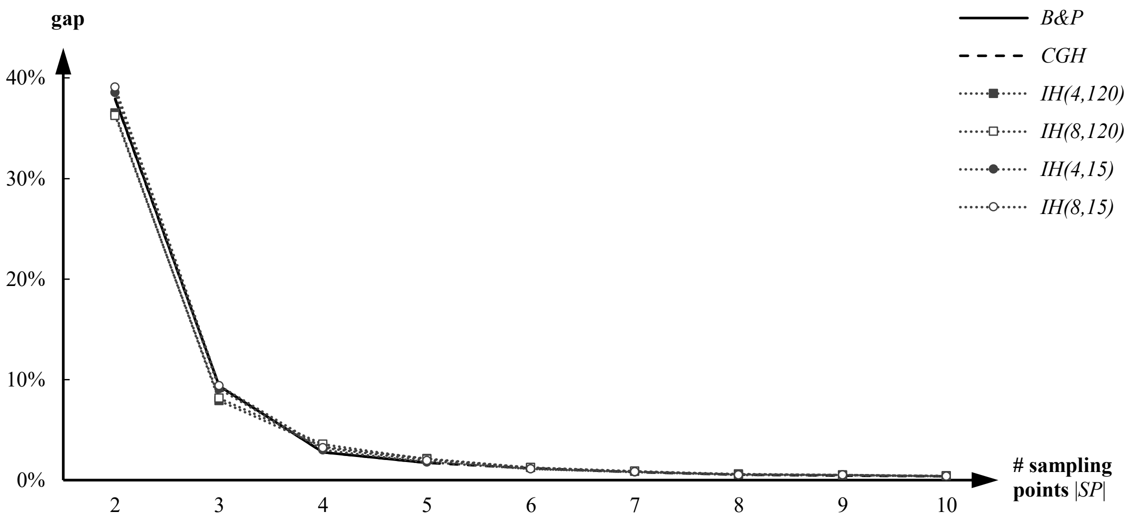

Figure 4: Effect of the number of sampling points on linearization gap, averaged over $|N|$ and $|FB|$

As a complement, Figure 5 displays the profit obtained when applying the algorithms' solutions with the real, non-linearized revenue functions. All mechanisms have in common that there is a huge profit increase when moving from 2 to 3 sampling points, but there is almost no further increase beyond 4 or 5 sampling points. Again, all test instances (not shown here) exhibit this behavior. Regarding the heuristics' performance, the figure illustrates the superior performance of *CGH*, *IH(4,15)*, and *IH(8,15)* already observed in the last subsection. Similar to Subsection 5.2.1, we report data for $|SP| \in \{2, 3, 4, 5\}$ regarding the *B&P* algorithm, although some instances could not be solved to optimality. *B&P* performs similar to the three best heuristics mentioned above, and even slightly worse than *IH(8,15)* for $|SP| \in \{4, 5\}$. This is partly due to instances not solved to optimality and partly due to the linearized objective function used in the optimization.

Thus, it seems more important to adequately capture demand's nonlinearity than to solve the problem to optimality. This observation is especially relevant for larger problem instances where a trade-off between using a heuristic with a good demand approximation and using an exact approach with only a few sampling points (or even linear demand) exists.

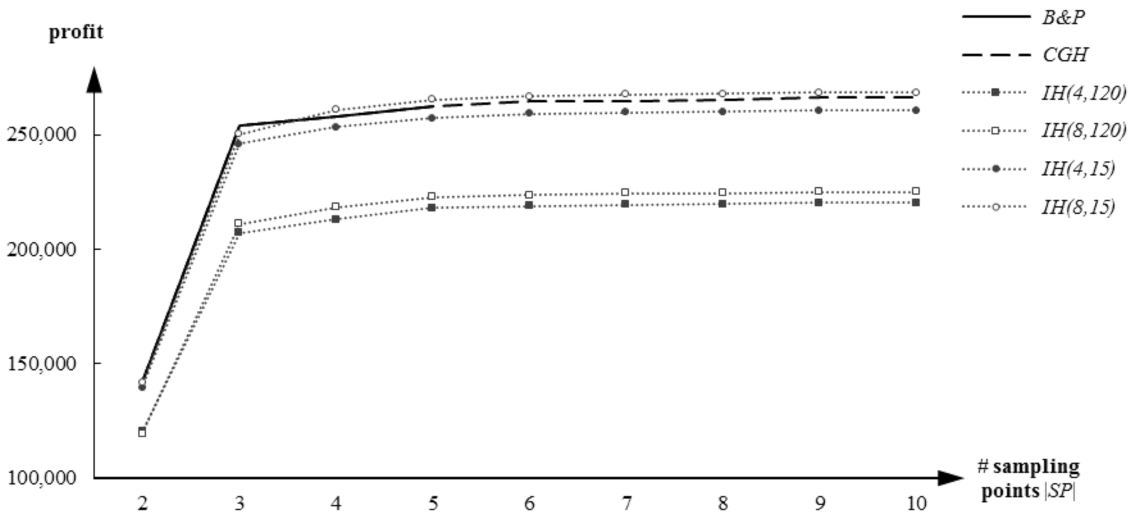

Figure 5: Effect of the number of sampling points on profit (non-linearized), averaged over $|N|$ and $|FB|$

5.2.4. Scalability of the Heuristics

In the previous subsections, numerical experiments with real-world data showed that the heuristics are scalable from 1 to 2 fleet bases. To investigate this further, we now also consider $|FB| = 3$ fleet bases. In the previous subsections, $|N| = 15$ airports were considered for both $|FB| = 1$ and $|FB| = 2$ fleet bases. Thus, we now focus on instances with $|N| = 15$. As our base setting obtained from industry (see Section 5.1) contains only two fleet bases, we generated a new setting for these tests. More specifically, we used an arbitrary instance with $|N| = 15$ obtained from the real-world data and modified it such that a third airport can be meaningfully considered as a fleet base. From this setting, we derived a total of 6 new test instances by combining $|FB| = \{1, 2, 3\}$ fleet bases with $q = \{5, 10\}$ aircraft. This allows meaningful comparisons among different numbers of fleet bases, however, the values obtained cannot be directly compared with our previous results for $|FB| = \{1, 2\}$ because other test instances were used.

Table 6 shows the results for $|SP| = 2$ and $|SP| = 5$, averaged over q. Obviously, a higher number of possible fleet bases offers more flexibility and more profit is obtained. In line with our previous results, *IH(4,120)* provides the lowest revenue, followed by *IH(8,120)*. Regarding the results for the Industry Heuristic, *IH(4,15)* and *IH(8,15)* obtain the highest revenues. As in the previous numerical experiments, the Column Generation Heuristic (*CGH*) outperforms *IH(8,15)*, obtaining higher revenues in less runtime, and generating less rotations.

Table 6: Effect of the number of fleet bases on the heuristics' performance ($|N| = 15$) averaged over q, artificial data

		|SP| = 2			**|SP| = 5**		
	heuristic	**|FB| = 1**	**|FB| = 2**	**|FB| = 3**	**|FB| = 1**	**|FB| = 2**	**|FB| = 3**
linearized profit	*CGH*	43,337.5	72,995.7	123,356.1	119,266.2	251,775.5	495,572.0
	IH(4, 120)	30,173.5	50,040.8	96,075.7	70,530.8	182,675.8	404,247.1
	IH(8, 120)	30,173.5	50,946.5	97,682.5	78,081.7	198,761.8	416,171.2
	IH(4, 15)	40,526.0	67,493.9	120,544.3	113,731.7	244,074.6	492,631.3
	IH(8, 15)	40,526.0	67,596.0	120,544.3	113,731.7	244,074.6	492,631.3
runtime [mm:ss]	*CGH*	00:07	00:14	00:26	00:11	00:19	01:02
	IH(4, 120)	00:02	00:05	00:08	00:03	00:05	00:09
	IH(8, 120)	00:05	00:12	00:31	00:05	00:14	00:30
	IH(4, 15)	00:11	00:30	01:08	00:12	00:33	01:35
	IH(8, 15)	00:22	01:21	04:25	00:25	01:30	04:29
generated rotations	*CGH*	316.0	613.0	922.5	415.5	874.0	1,451.5
	IH(4, 120)	728.0	1,331.0	2,104.0	728.0	1,331.0	2,104.0
	IH(8, 120)	1,502.0	3,193.0	5,972.0	1,502.0	3,193.0	5,972.0
	IH(4, 15)	3,349.0	6,333.0	10,337.0	3,349.0	6,333.0	10,337.0
	IH(8, 15)	5,428.0	11,920.0	22,818.0	5,428.0	11,920.0	22,818.0

In general, the number of rotations generated is similar to the previous tests. It roughly doubles from $|FB| = 1$ to $|FB| = 2$ and increases by a factor of about 1.7 from $|FB| = 2$ to $|FB| = 3$. This leads to a 2x – 3x runtime for each additional fleet base. Thus, the heuristics still provide fast solutions with all runtimes below 5 minutes and many less than a minute.

6 Conclusion

In this paper, we presented a novel model which integrates scheduling and maintenance routing for point-to-point airlines that dispose of a homogeneous fleet. Particular attention is paid to demand modeling. For each departure and arrival airport pair, we consider demand from multiple customer segments, which may require certain time windows for their outgoing as well as return flights. Moreover, not least due to the industry standard revenue management systems in place today, marginal revenues decline in the number of seats available for a customer segment. The resulting nonlinear total revenue function is captured by a piecewise linear approximation.

Three solution approaches are developed to solve the model. Our exact branch-and-price based procedure generates necessary rotations online, during the branching process, seeking integer solutions. By contrast, two heuristics generate some rotations offline in advance and then use a standard mixed integer linear programming solver (CPLEX) to obtain integer solutions. The first heuristic, *CGH*, is based on the branch-and-price procedure but uses column generation to solve the LP relaxation only in the root node to obtain a set of rotations. The second heuristic, *IH*, augments a standard approach in industry where airlines often consider a few ‘intuitive’ or ‘desirable’ types of rotations.

The numerical experiments with real-world data from a major European point-to-point airline document that the branch-and-price approach can solve problem instances with up to 15 airports and 10 aircraft to optimality in a reasonable amount of time. For larger problem instances, the heuristics can be used. Here, *CGH* clearly outperforms the other heuristics. Its revenue is comparable to that of *IH*’s best variants, but its runtime is only a fraction. The comparison with the results from our branch-and-price approach shows that it always yields near-optimal solutions and often even finds an optimal solution. Moreover, the experiments have shown that the quality of the revenue function approximation is crucial, much more important than the solution approach. It is more important to adequately capture demand’s nonlinearity than to solve the problem to optimality. We observed considerable revenue losses when too few sampling points were used, especially when we mimicked approaches with linear revenue functions with only two sampling points.

References

Abara J (1989) Applying integer linear programming to the fleet assignment problem. *Interfaces* 19(4):20-28.

Barnhart C, Boland NL, Clarke LW, Johnson EL, Nemhauser GL, Shenoi RG (1998a) Flight string models for aircraft fleeting and routing. *Transportation Sci.* 32(3):208-220.

Barnhart C, Cohn AM (2004) Airline schedule planning: Accomplishments and opportunities. *Manuf. & Service Oper. Manag.* 6(1):3-22.

Barnhart C, Cohn AM, Johnson EL, Klabjan D, Nemhauser GL, Vance PH (2003) Airline Crew Scheduling. In: Hall RW (editor): *Handbook of Transportation Science*, 2nd edition, chap. 14, 517-560. Kluwer Academic Publishers, Boston, MA.

Barnhart C, Farahat A, Lohatepanont M (2009) Airline fleet assignment with enhanced revenue modeling. *Oper. Res.* 57(1):231-244.

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998b) Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* 46(3):316–329.

Barnhart C, Kniker TS, Lohatepanont M (2002) Itinerary-based airline fleet assignment. *Transportation Sci.* 36(2):199-217.

Barnhart C, Lu F, Shenoi RG (1998) Integrated airline schedule planning. In: Yu G (editor): *Operations Research in the Airline Industry*, chap. 13, 384-403. Kluwer Academic Publishers, Boston, MA.

Ben-Akiva M, Lerman SR (1985) *Discrete choice analysis – Theory and application to travel demand.* Cambridge, London, UK.

Cacchiani V, Salazar-González J-J (2013) A heuristic approach for an integrated fleet-assignment, aircraft-routing and crew-pairing problem. *Electron. Notes in Discrete Math.* 41:391-398.

Cadarso L, Marín Á (2011) Integrated robust airline schedule development. *Procedia Soc and Behav. Sci.* 20:1041-1050.

Cadarso L, Marín Á (2013) Robust passenger oriented timetable and fleet assignment integration in airline schedule planning. *J. Air Transp. Manag.* 26:44-49.

Clarke LW, Hane CA, Johnson EL, Nemhauser GL (1996) Maintenance and crew considerations in fleet assignment. *Transportation Sci.* 30(3):249-260.

Cohn AM, Barnhart C (2003) Improving crew scheduling by incorporating key maintenance routing decisions. *Oper. Res.* 51(3):387-396.

Cordeau J-F, Stojković G, Soumis F, Desrosiers J (2001) Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Sci.* 35(4):375-388.

Desaulniers G, Desrosiers J, Dumas Y, Solomon MM, Soumis F (1997) Daily aircraft routing and scheduling. *Manag. Sci.* 43(6):841-855.

Desrosiers J, Lübbecke ME (2005) A primer in column generation. In: Desaulniers G, Desrosiers J, Solomon MM (editors): *Column Generation,* chap. 1, 1–32. Springer, New York, NY.

Desrosiers J, Soumis F, Desrochers M (1984) Routing with time windows by column generation. *Networks* 14(4):545-565.

Dobruszkes F (2006) An analysis of European low-cost airlines and their networks. *J. Transp. Geogr.* 14(3):249-264.

Dück V, Ionescu L, Kliewer N, Suhl L (2012) Increasing stability of crew and aircraft schedules. *Transp. Res. Part C: Emerg. Technologies* 20(1):47-61.

Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1) 269-271.

El Moudani W, Mora-Camino F (2000) A dynamic approach for aircraft assignment and maintenance scheduling by airlines. *J. Air Transp. Manag.* 6(4):233-237.

Erdmann A, Nolte A, Noltemeier A, Schrader R (2001) Modeling and solving an airline schedule generation problem. *Ann. Oper. Res* 107(1):117-142.

Farkas A (1996) *The influence of network effects and yield management on airline fleet assignment decisions.* Ph.D. thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Gao C, Johnson EL, Smith BC (2009) Integrated airline fleet and crew robust planning. *Transportation Sci.* 43(1):2-16.

Gönsch J, Steinhardt C (2015) On the incorporation of upgrades into airline network revenue management. *Rev. Manag. Sci.* 9:635–660.

Grosche T, Rothlauf F (2008) Simultaneous airline scheduling. In: Fink A, Rothlauf F (editors): *Advances in Computational Intelligence*, 81-108. Springer, Berlin, Germany.

Groß S, Lück M, Schröder A (2013) The low cost carrier – a worldwide phenomenon?! In: Groß S, Lück M (editors): *The Low Cost Carrier Worldwide*, chap. 1, 3-15. Ashgate, Farnham.

Hane CA, Barnhart C, Johnson EL, Marsten RE, Nemhauser GL, Sigismondi G (1995) The fleet assignment problem: Solving a large-scale integer program. *Math. Programming* 70:211-232.

Haouari M, Aissaoui N, Mansour FZ (2009) Network flow-based approaches for integrated aircraft fleeting and routing. *Europ. J. Oper. Res.* 193(2):591-599.

Haouari M, Sherali HD, Mansour FZ, Aissaoui N (2011) Exact approaches for integrated aircraft fleeting and routing at TunisAir. *Comput. Optim. Appl.* 49(2):213-239.

Ioachim I, Desrosiers J, Soumis F, Bélanger N (1999) Fleet assignment and routing with schedule synchronization constraints. *Europ. J. Oper. Res.* 119(1):75-90.

Jacobs TL, Smith BC, Johnson EL (2008) Incorporating network flow effects into the airline fleet assignment process. *Transportation Sci.* 42(4):514-529.

Keskinocak P, Tayur S (1998) Scheduling of time-shared jet aircraft. *Transportation Sci.* 32(3):277-294.

Kim D, Barnhart C (2007) Flight schedule design for a charter airline. *Comp. Oper. Res.* 34(6):1516-1531.

Klabjan D, Johnson EL, Nemhauser GL, Gelman E, Ramaswamy S (2002) Airline crew scheduling with time windows and plane-count constraints. *Transportation Sci.* 36(3):337-348.

Kniker TS (1998) Itinerary-based airline fleet assignment. Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

Lawton TC (2002) *Cleared for take-off – Structure and strategy in the low fare airline business*. Ashgate, Burlington, VT.

Liang Z, Chaovalitwongse WA (2013) A network-based model for the integrated weekly aircraft maintenance routing and fleet assignment problem. *Transportation Sci.* 47(4):493-507.

Lohatepanont M (2002) Airline fleet assignment and schedule design: Integrated models and algorithms. Ph.D. thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Lohatepanont M, Barnhart C (2004) Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Sci.* 38(1):19-32.

Lübbecke ME, Desrosiers J (2005) Selected topics in column generation. *Oper. Res.* 53(6):1007-1023.

Mercier A, Cordeau J-F, Soumis F (2005) A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Comp. Oper. Res.* 32(6):1451-1476.

Mercier A, Soumis F (2007) An integrated aircraft routing, crew scheduling and flight retiming model. *Comp. Oper. Res.* 34(8):2251-2265.

Mercier A (2008) A theoretical comparison of feasibility cuts for the integrated aircraft-routing and crew-pairing problem. *Transportation Sci.* 42(1):87-104.

O'Connell JF, Williams G (2005) Passengers' perceptions of low cost airlines and full service carriers: A case study involving Ryanair, Aer Lingus, Air Asia and Malaysia Airlines. *J. Air Transp. Manag.* 11(4):259-272.

Papadakos N (2009) Integrated airline scheduling. *Comp. Oper. Res.* 36(1):176-195.

Pape U (1974) Implementation and efficiency of Moore algorithms for the shortest route problem. *Math. Programming* 7(1):212-222.

Phillips RL (2005) *Pricing and revenue optimization*. Stanford University Press, Stanford, CA.

Pita JP, Barnhart C, Antunes AP (2013) Integrated flight scheduling and fleet assignment under airport congestion. *Transportation Sci.* 47(4):477-492.

Rexing B, Barnhart C, Kniker TS, Jarrah AI, Krishnamurthy N (2000) Airline fleet assignment with time windows. *Transportation Sci.* 34(1):1-20.

Ronen D (2000) Scheduling charter aircraft. *J. Oper. Res. Society* 51(3):258-262.

Rushmeier RA, Kontogiorgis SA (1997) Advances in the optimization of airline fleet assignment. *Transportation Sci.* 31(2):159-169.

Salazar-González J-J (2014) Approaches to solve the fleet-assignment, aircraft-routing, crew-pairing and crew-rostering problem. *Omega* 43:71-82.

Sandhu R, Klabjan D (2007) Integrated airline fleeting and crew-pairing decisions. *Oper. Res.* 55(3):439-456.

Sarac A, Batta R, Rump CM (2006) A branch-and-price approach for operational aircraft maintenance routing. *Europ. J. Oper. Res.* 175(3):1850-1869.

Sherali HD, Bae K-H, Haouari M (2010) Integrated airline schedule design and fleet assignment: Polyhedral analysis and Benders' decomposition approach. *INFORMS J. Comput.* 22(4):500-513.

Sherali HD, Bae K-H, Haouari M (2013) An integrated approach for airline flight selection and timing, fleet assignment, and aircraft routing. *Transportation Sci.* 47(4):455-476.

Vanderbeck F (2000) On Dantzig-Wolfe Decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Oper. Res.* 48(1):111-128.

Weide O (2009) Robust and integrated airline scheduling. Ph.D. thesis, Department of Engineering Science, University of Auckland, New Zealand.

Weide O, Ryan D, Ehrgott M (2010) An iterative approach to robust and integrated aircraft routing and crew scheduling. *Comp. Oper. Res.* 37(5):833-844.

Williams G (2001) Will Europe's charter carriers be replaced by "no-frills" scheduled airlines? *J. Air Transp. Manag.* 7:277-286.

Yan S, Tseng C-H (2002) A passenger demand model for airline flight scheduling and fleet routing. *Comp. Oper. Res.* 29(11):1559-1581.