Submitted to *Transportation Science*
manuscript TS-2014-0293

Capacity Uncertainty in Airline Revenue Management: Models, Algorithms, and Computations

Christina Büsing
Lehrstuhl II für Mathematik, RWTH Aachen University, buesing@math2.rwth-aachen.de

Daniel Kadatz
Freie Universität Berlin, daniel.kadatz@gmail.com

Catherine Cleophas
Department Management Science, Lancaster University Management School, c.cleophas@lancaster.ac.uk

Most airline revenue optimization models assume capacity to be fixed by fleet assignment, and thus treat it as deterministic. However, empirical data shows that on 40% of flights, capacity is updated at least once within the booking horizon. Capacity updates can be caused by fleet-assignment re-optimizations or by short-term operational problems. This paper proposes a first model to integrate the resulting capacity uncertainty in the leg-based airline revenue management process. While assuming deterministic demand, the proposed model includes stochastic scenarios to represent potential capacity updates. To derive optimal inventory controls, we provide both a mixed-integer-program and a combinatorial solution approach, and discuss efficient ways of optimizing the special case of a single capacity update. We also explore effects of denied boarding cost and the model's relationship to the static overbooking problem. We numerically evaluate the model on empirically calibrated demand instances and benchmark it on the established deterministic approach and an upper bound based on perfect hindsight. In addition, we show that the combinatorial solution approach reduces the computational effort. Finally, we compare the static overbooking approach derived from the capacity uncertainty model to existing EMSR-based approaches.

Key words: revenue management, aviation, capacity uncertainty, overbooking, stochastic optimization, combinatorial optimization

History: Fourth revised submission, January 4, 2018

---

1. Introduction

Classically, airline revenue management controls demand for capacitated, perishable products to maximize revenue from ticket sales. A comprehensive overview of mathematical models and methods is provided by Talluri and van Ryzin (2004). The idea that capacity is limited is crucial to the concept – one of the most basic restrictions of the revenue optimization problem is that one cannot sell more than there are units of capacity given. Most models consider the capacity restriction as constant over the booking horizon.

Yet, in practice, flights' capacity is anything but fixed. After assigning aircrafts to flights, airlines publish a schedule and offer ticket reservations from almost one year prior to departure. Throughout the booking horizon, aircraft assignments can change due to special sales events, adapted demand forecasts, or changes in crew planning. Even shortly before departure, technical complications can cause new aircrafts to be assigned.

To verify this observation, we analyze an empirical data set from a major European network carrier, documenting the number of seats available to the economy compartment throughout the booking horizon of 5,867 intercontinental flights departing in a single month of 2014. For 40% of these flights, aircraft changes lead to capacity updates of at least 10% of the previous value. For 35% of flights, capacity updates of at least 50% were reported. Domain experts indicate that more than eight weeks prior to departure, updates are primarily caused by fleet assignment (71%), whereas from two weeks prior to departure on, updates are primarily driven by operational difficulties (19%). Classically, revenue management considers capacity updates only after they are announced, via re-optimization. This approach is also applied at the airline that supplied the analyzed empirical data: Inventory controls are optimized to maximize revenue for the initial capacity. Any announced capacity update triggers a re-optimization. In a computational study, we benchmark the proposed scenario-based model on this approach.

To illustrate the effects of capacity updates, consider two examples from the empirical data: Between 201 and 172 days before departure, a re-optimization of the regular fleet assignment shrank the economy compartment of all departures of a particular flight from Düsseldorf to New York from 225 to 165 seats. This is unlikely to have caused denied boardings, as most bookings occur later in the booking horizon. Nevertheless, when assuming a small capacity, revenue management implements more restrictive inventory controls to reserve seats for valuable, late-booking customers. Thus, the initial inventory controls were suboptimal for the actual, larger capacity. In a more extreme example, one day before departure, the economy compartment for a flight from Munich to New York shrank from 270 to 161 seats. This was most likely caused by operational difficulties. For a fully booked flight, it could have caused 109 denied boardings, excluding effects from intentional overbooking.

Existing models that do relax the fixed capacity assumption predominantly aim to integrate fleet assignment and revenue management. For concepts such as 'demand driven dispatch' (Berge and Hopperstad 1993), revenue management triggers capacity updates to adjust to demand variation. We regard capacity updates that are controlled by revenue management as *endogenous*. In contrast, this paper considers *exogenous* updates. After an aircraft change, exogenous capacity updates are announced to the revenue management department, but the automated algorithms do not anticipate them. The idea that such updates can cause collateral damage when not anticipated in the optimization model motivates our work. To our knowledge, only Wang and Regan (2002, 2006) propose an earlier revenue management model to account for uncertain capacity. As that model is motivated by the idea of aircraft swaps, it assumes a single point in time when updates are announced and only two possible capacities. However, the empirical data shows that capacity updates can be announced at any time in the booking horizon, as their timing depends in part on the events that trigger capacity changes. Furthermore, more than two final capacities can result from aircraft changes in practice. We thus identify a research gap beyond endogenous capacity updates and revenue management models regarding capacity as a deterministic and fixed parameter.

This paper contributes to reducing this gap as follows:

- We propose a first leg-based revenue management model that explicitly considers exogenous capacity changes occuring at multiple times in the booking horizon and leading to an arbitrary number of potential final capacities. We term this model the *quantity-based revenue management under capacity uncertainty (RMCU)* problem, and numerically analyze its sensitivity to problem characteristics, such as the time and magnitude of update.
- To solve the RMCU problem, we provide a mixed-integer program (MIP) as well as a combinatorial solution approach. In a computational study, we show that the combinatorial approach solves the problem in less than 0.5 percent of the run time required to solve the MIP via CPLEX. Furthermore, we suggest exploiting problem characteristics to efficiently derive solutions for the special case of updates occurring only at a single point in time.
- Last but not least, we consider the effect of denied boarding cost and relate the RMCU to static overbooking by transforming a static formulation of overbooking into an RMCU problem. In the numerical study, we show that RMCU-based overbooking achieves comparable results to EMSR-based approaches reviewed in Aydin et al. (2012).

This paper is organized as follows: The next section reviews related work, both on revenue management under capacity uncertainty and on uncertain capacity utilization. Section 3 presents the RMCU problem, a leg-based revenue management model assuming deterministic demand and stochastic capacity. Next to a mixed-integer program formulation, this section introduces a combinatorial solution approach, analyzes the special case of a single capacity update, the effect of denied boarding cost, and the problem's relationship to static overbooking. To prepare the computational study, we also state an upper bound on the expected revenue and model the common approach of only re-optimizing revenue management when capacity updates are announced. Section 4 documents the results of benchmarking solution approaches and of analyzing their sensitivity. This section also includes results that numerically illustrate our remarks on overbooking. Finally, we summarize our findings and their managerial implications and point out future research opportunities.

2. Related Work

Two streams of revenue management research are closely related to the idea of accounting for capacity uncertainty. On the one hand, the concept of overbooking entails controlling a virtual capacity to account for uncertain capacity utilization. On the other hand, approaches aiming to integrate fleet assignment and revenue management systematically trigger capacity updates to compensate demand variation.

Overbooking was considered as early as 1958 (Beckmann and Bobkoski 1958); dynamic overbooking models exist since the 1970s (Rothstein 1971). By selling tickets beyond the physical capacity, airlines compensate for cancellations and no-shows. E.g., Rothstein (1971) and Subramanian et al. (1999) model cancellations as a Markov process. More recently, Aydin et al. (2012) consider static models of class-dependent cancellations and no-shows and a dynamic model that considers bookings and cancellations as streams of events. Topaloglu et al. (2012) propose open loop policies for joint overbooking and capacity controls on a single flight leg.

Upgrading passengers when a compartment is depleted is a straight-forward approach to integrate overbooking and capacity adaptation. This idea is first proposed by Alstrup et al. (1986): The authors solve a dynamic overbooking problem with two segments by substitution. Karaesmen and van Ryzin (2004) describe the possibility of multiple substitutable inventory classes. Both approaches aim to compensate demand uncertainty by adjusting virtual capacity between the aircraft's compartments.

Insufficient overbooking and unanticipated capacity increases may cause spoilage, as seats remain unsold. Excessive overbooking and unanticipated capacity decreases may cause spill, where valuable demand is rejected and denied boardings occur (Belobaba and Farkas 1999). However, cancellations mostly happen in increments of one. Updates to the fleet assignment result in significantly increased or reduced numbers of seats. Therefore, overbooking approaches cannot be simply adapted to consider uncertain capacity. Our model incorporates this uncertainty in the form of stochastic scenarios, rather than as dynamic, incremental changes considered by most overbooking research. Nevertheless, in the last part of the next section, we formalize the relationship between the two problems. The respective remarks may serve as an inspiration for future research in this area.

Fleet assignment ideally pairs the largest aircraft with the flights that expect the highest and most valuable demand (Barnhart et al. 2009). When revenue management can adapt fleet assignment to accommodate demand variation, this is called i.a. 'dynamic capacity management' (Frank et al. 2006), 'demand driven swapping' (Bish et al. 2004), or 'demand driven dispatch' (Berge and

Hopperstad 1993). Berge and Hopperstad (1993) refer to a 1–5% revenue improvement caused by relaxing the assumption of a fixed capacity. De Boer (2004) proposes a dynamic version of EMSR called EMSRd. Numerical examples show that EMSRd can be very useful, however the best results are achieved for instances with few fare classes. Frank et al. (2006) show positive effects from continuously adjusted fleet assignments given dependent demand.

While organizational constraints mostly preclude automization in practice, such contributions motivate manual re-fleeting processes aiming to compensate demand variation. In practice, these are implemented via analyst communication across planning departments. As the resulting capacity updates are not anticipated by the revenue management model, this results in further capacity uncertainty – albeit of a well-meaning nature.

So far, capacity uncertainty in revenue management has been explicitly addressed only by Wang and Regan (2002, 2006). In a leg-based model with uncertain demand, Wang and Regan (2002) allow for a single possible capacity swap, which occurs at an a priori known point of time. Extending Liang (1999), the continous-time optimization model divides the time horizon to consider a period before and a period after the swap. Wang and Regan (2002) present an optimal policy for dealing with two potential capacities on a flight and mathematically prove the potential revenue improvement. Their focus is also on preventing overbooking when the revenue risk from a potential capacity decrease is too high.

In a second paper, Wang and Regan (2006) abandon the focus on overbooking and provide further numerical results from a simulation study considering different capacities, demand mixes and markets. The contribution focuses on endogenous capacity swaps: The optimal policy updates repetitively over the booking horizon and is compared with heuristics that allow for only one update on a particular time.

The model presented in this paper takes a perspective of discrete-time and deterministic demand. Rather than allowing for a single capacity swap at a single point of time, we consider the possibility of more than two potential capacities and allow for capacity updates at any point of time in the booking horizon. This is motivated by our empirical analysis of capacity changes in the airline industry, which are neither limited to a certain new capacity nor to a single time in the booking horizon.

In the next section, we introduce the quantity-based revenue management under capacity uncertainty problem.

3. Model and Solution Approaches

The model proposed here represents an alternative to that considered in Wang and Regan (2002, 2006): It allows for potential updates to occur at any time of the booking horizon and to result in an arbitrary number of new capacities. Our model considers quantity- and leg-based revenue maximization and assumes independent, deterministic demand. It anticipates only a single capacity update per departure – however, this could be remedied by resolving the model after each expected time of update. As the revenue management problem is frequently solved independently for physical flight compartments, we consider only a single compartment per flight. To avoid overlapping effects, this model does not account for no-shows and cancellations; all bookings require one unit of capacity. The resulting *revenue management under capacity uncertainty (RMCU) problem* anticipates future capacity updates when optimizing the number of tickets to offer. The goal is to define a *global strategy* of fare class availability, which applies until the capacity is updated. After such an update, the global strategy is abandoned for a *scenario-based strategy*, which optimizes inventory controls for the new capacity.

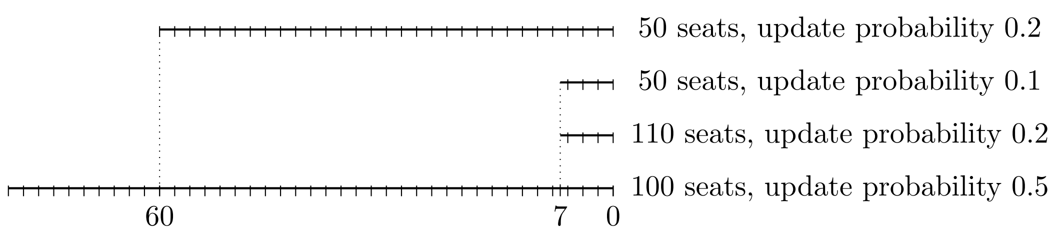

Figure 1 **Exemplary time line**

For a small example, consider Figure 1. At the time of optimization, the global strategy considers four possible scenarios: With probability 0.2, capacity will be updated to 50 seats at 60 days before departure; with probability 0.1, capacity will be updated to 50 seats at 7 days before departure; with probability 0.2, capacity will be updated to 110 seats at 7 days before departure; finally, with probability 0.5, the initially announced capacity of 100 seats will never be updated. Whenever one of the scenarios realizes, the corresponding scenario-based strategy is implemented.

3.1. Model Description

Let the booking horizon start at time $\hat{t} \in \mathbb{N}$ and end with departure at time 0. The set F contains all fare classes that can be offered. Revenue r_f is fixed per fare class $f \in F$. For each fare class $f \in F$ and every time $t \in T := \{\hat{t}, \ldots, 0\}$, expected demand is indicated by $D_{ft} \in \mathbb{N}$. The number of acceptable denied boardings is bounded by K; $b_i \in \mathbb{N}$ denotes the cost of the ith denied boarding, $i \in \{1, \ldots, K\}$. We assume that the denied boarding cost increases, i.e., $b_1 \leq \ldots \leq b_K$. The increase may be linear or, as in the computational study, exponential.

We model capacity updates via a set of scenarios S. These scenarios describe all relevant combinations of update time and resulting capacity. Every scenario $s \in S$ defines an update time

$t^s \in T$ and a resulting capacity $c^s \in \mathbb{N}$. The probability of scenario $s \in S$ is denoted by p^s, where $\sum_{s \in S} p^s = 1$.

A feasible solution of the RMCU is to define both a global strategy and for each scenario a scenario-based strategy. The *global strategy* x_{ft} defines the number of tickets to offer in fare class $f \in F$ at time $t \in T$. This strategy is executed until scenario $s \in S$ updates the capacity to c^s at time t^s. Scenario $s \in S$ triggers the *scenario-based strategy* x^s_{ft}, $f \in F$, $t \in T$, which defines the number of tickets offered in fare class f at time $t \leq t^s$. If the number of tickets offered between $\hat{t}$ and $t^s + 1$ exceeds capacity c^s, no further tickets can be offered, i.e., $x^s_{ft} = 0$, $f \in F$, $t \in T$, $t \leq t^s$. In addition, denied boardings result. In the following, the variable $\alpha^s_i \in \{0, 1\}$ indicates whether the ith boarding is denied in scenario $s \in S$, causing denied boarding cost b_i. Thus, the resulting revenue $R^s(x, (x^s, \alpha^s))$ in scenario $s \in S$ equals

$$R^s(x, (x^s, \alpha^s)) = \sum_{f \in F} r_f \left(\sum_{t=\hat{t}}^{t^s+1} x_{ft} + \sum_{t=t^s}^{0} x^s_{ft} \right) - \sum_{i=1}^{K} b_i \alpha^s_i .$$

The model's objective is to maximize the *expected revenue* $R(x, (x^s, \alpha^s)_{s \in S})$ for such a strategy set $x \in \mathbb{N}^{\hat{t} \times |F|}$, $(x^s, \alpha^s) \in \mathbb{N}^{t^s \times |F|}$, $s \in S$, given by

$$R(x, (x^s, \alpha^s)_{s \in S}) = \sum_{s \in S} p^s R^s(x, (x^s, \alpha^s)) = \sum_{s \in S} p^s \left(\sum_{f \in F} r_f \left(\sum_{t=\hat{t}}^{t^s+1} x_{ft} + \sum_{t=t^s}^{0} x^s_{ft} \right) - \sum_{i=1}^{K} b_i \alpha^s_i \right).$$

A global strategy $x \in \mathbb{N}^{\hat{t} \times |F|}$ is called *optimal*, if for each scenario $s \in S$ there exists a scenario-based strategy x^s or denied boardings α^s such that the expected revenue $R(x, (x^s, \alpha^s)_{s \in S})$ is maximal. Let $x \in \mathbb{N}^{\hat{t} \times |F|}$ be a global strategy that is not necessarily optimal and let $c^s_r = c^s - \sum_{t=T}^{t^s+1} \sum_{f \in F} x_{ft}$ be the capacity remaining after time $t^s + 1$, $s \in S$. A scenario-based strategy x^s and denied boardings α^s are *optimal* according to x, if (x^s, α^s) maximizes the revenue for the remaining capacity c^s_r.

3.2. Mixed-Integer Program Formulation

An optimal solution to the RMCU problem can be derived via a mixed-integer program (MIP). In addition to the already introduced variables x for the global strategy, x^s, $s \in S$ for the scenario-based strategy, and α^s for the denied boarding in $s \in S$, we introduce decision variable $z^s \in \{0, 1\}$ to indicate the necessity of denied boardings starting at time t^s in scenario s. The following RMCU-MIP models the RMCU problem:

$$\text{(RMCU-MIP)} \qquad \max \sum_{s\in S} p^s \left(\sum_{f\in F} r_f \left(\sum_{t=\hat{t}}^{t^s+1} x_{ft} + \sum_{t=t^s}^{0} x^s_{ft} \right) - \sum_{i=1}^{K} b_i \alpha^s_i \right)$$

$$\sum_{f\in F} \left(\sum_{t=\hat{t}}^{t^s+1} x_{ft} + \sum_{t=t^s}^{0} x^s_{ft} \right) \le c^s + \sum_{i=1}^{K} \alpha^s_i \qquad \forall s\in S \tag{1}$$

$$\sum_{i=1}^{K} \alpha^s_i \le K z^s \qquad \forall s \in S \tag{2}$$

$$x_{ft} \le D_{ft} \qquad \forall t\in T,\ f\in F \tag{3}$$

$$x^s_{ft} \le D_{ft}(1-z^s) \qquad \forall t\in T,\ f\in F, s\in S \tag{4}$$

$$x_{ft}, x^s_{ft} \ge 0 \qquad \forall t\in T,\ f\in F,\ s\in S$$

$$\alpha^s_i, z^s \in \{0,1\} \qquad \forall i \in \{1,\dots,K\},\ s\in S.$$

Constraint (1) guarantees that the number of sold tickets, adjusted by potential denied boardings, does not exceed the capacity in each scenario. Constraint (2) guarantees that $z^s = 1$ if denied boardings occur in scenario s. Constraint (3) restricts the number of sold tickets for the global strategy to the expected demand D_{ft} for each fare class $f \in F$ at each point in time $t \in T$. Constraint (4) guarantees that no tickets can be offered in scenario $s \in S$ if any denied boardings occur, i.e. $z^s = 1$. Otherwise, the number of tickets offered is bounded by the expected demand D_{ft}. Solving the RMCU-MIP produces a global strategy and scenario-based strategies maximizing the expected revenue.

In the following, we introduce an alternative solution approach based on combinatorial optimization. In general, combinatorial algorithms have the advantage of being more computationally efficient, easy to implement, and easily adaptable as starting heuristics for more complex settings. With regard to the RMCU, more complex settings may include extensions to a network-model or to one of stochastic, dependent demand.

3.3. Combinatorial Solution Approach

The main idea of the combinatorial solution approach relies on the following properties of an optimal global strategy and scenario-based strategies, which will be proven later:

1. If an optimal global strategy is known, then after the time t^s, optimal scenario-based strategies maximize the revenue for the remaining capacity.

2. If an optimal global strategy is not known, but the number of tickets offered until any time t^s is known, then maximizing the revenue for this number of tickets is an optimal global strategy.

Combining these properties reduces the problem of computing an optimal global strategy and scenario-based strategies to computing the number of tickets to offer until any point t^s, $s \in S$.

Subsequently, we formally show the two properties and design a combinatorial algorithm based on a longest path problem to compute an optimal number of tickets to offer.

The two properties call for solving subproblems of the original revenue management problem, considering a subinterval in time and a new capacity available during that time. More formally, we define the *restricted deterministic revenue management (RDRM) problem* $\mathcal{R}(\bar{t}, \underline{t}, \bar{c})$ by computing the number of tickets x_{ft} to offer per fare class $f \in F$ and time $\bar{t} \geq t \geq \underline{t}$ respecting the expected demand D_{ft} and the capacity $\bar{c} \geq 0$ available at $\bar{t}$ so as to maximize the revenue $R_{[\bar{t},\underline{t},\bar{c}]}(x) := \sum_{f \in F} \sum_{t=\bar{t}}^{\underline{t}} r_f \cdot x_{ft}$. We denote the optimal value by $R(\bar{t}, \underline{t}, \bar{c})$. Note that the RDRM problem does not anticipate any capacity updates. It can easily be solved by a greedy-algorithm, which sells tickets in the most expensive fare class until either the demand or the capacity is exhausted. If there is capacity left, it sells tickets in the next cheaper fare class and so on.

Formalizing the first property calls for additional notation: Let T_b denote the set of points in time where an update may occur, i.e., $T_b := \{t \in T \mid \exists s \in S, t^s = t\} \cup \{\hat{t}\} = \{t_0, t_1, \ldots, t_N\}$ with $t_0 = \hat{t} \geq t_1 \geq \ldots \geq t_N$. Let x be a global strategy. We then define $c_i(x)$ as the number of tickets to offer in the time period $[t_{i-1}, t_i - 1]$, i.e., $c_i(x) = \sum_{f \in F} \sum_{t=t_{i-1}}^{t_i - 1} x_{ft}$, $i = 1, \ldots, N$.

Lemma 1. *Let $\overline{x}$ be an optimal global strategy. If $c^s - c_i(\overline{x}) \geq 0$, let $\overline{x}^s$ be an optimal solution for the RDRM problem $\mathcal{R}(t^s, 0, c^s - c_i(\overline{x}))$ with $t_i = t^s$, $s \in S$. If $c^s - c_i(\overline{x}) < 0$, define $\overline{\alpha}^s$ as the necessary number of denied boardings. Then, $\overline{x}$ and $(\overline{x}^s, \overline{\alpha}^s)$, $s \in S$, maximize the expected revenue for the quantity-based revenue management under capacity uncertainty (RMCU) problem.*

Proof. For simplicity, only consider the case of $c^s - c_i(\overline{x}) \geq 0$ for all $s \in S$. Start by rewriting the expected revenue

$$\begin{aligned} R(x, (x^s, \alpha^s)_{s \in S}) &= \sum_{s \in S} p^s \left(\sum_{f \in F} r_f \left(\sum_{t=\hat{t}}^{t^s+1} x_{ft} + \sum_{t=t^s}^{0} x^s_{ft} \right) - \sum_{i=1}^{K} b_i \alpha^s_i \right) \\ &= \sum_{s \in S} p^s \sum_{f \in F} r_f \sum_{t=\hat{t}}^{t^s+1} x_{ft} + \sum_{s \in S} p^s R_{[t^s, 0, c^s - c_i(x)]}(x^s), \end{aligned}$$

since $\alpha^s_i = 0$ for all $s \in S$. Assume that $\overline{x}$ in combination with $\overline{x}^s$ as defined in the lemma is not an optimal solution. On the other hand, let $\tilde{x}^s$ be scenario-based strategies such that $\overline{x}$ in combination with $\tilde{x}^s$ maximizes the expected revenue. Hence, there exists a scenario $s^* \in S$ such that $R_{[t^{s^*}, 0, c^{s^*} - c_i(\overline{x})]}(\tilde{x}^{s^*}) > R_{[t^{s^*}, 0, c^{s^*} - c_i(\overline{x})]}(\overline{x}^{s^*})$. This contradicts the definition of $\overline{x}^s$ as an optimal solution to $\mathcal{R}(t^s, 0, c^s - c_i(\overline{x}))$. Q.E.D.

According to this lemma, solving a deterministic revenue management problem for each scenario generates the best scenario-based strategies for any global strategy. Now, assume that the optimal global strategy of the RMCU is not known but that the number of tickets to offer in the periods $[t_{i-1}, t_i - 1]$, $i \in \{1, \ldots, N\}$, $t_i \in T_b$, $t_0 = \hat{t}$, is known. Lemma 2 provides a way of computing an optimal global strategy using this information.

Lemma 2. *Let $\overline{x}$ be an optimal global strategy. Let $\tilde{x}$ be a global strategy, such that the restricted strategy for the time period $[t_{i-1}, t_i - 1]$, denoted by $\tilde{x}_{[t_{i-1}, t_i - 1]}$, is an optimal solution to $\mathcal{R}(t_{i-1}, t_i - 1, c_i(\overline{x}))$, $i = 1, \ldots, N$. Then $\tilde{x}$ is an optimal global strategy.*

Proof. Since $c_i(\overline{x}) = c_i(\tilde{x}) := c_i$ for all $i = 1, \ldots, N$, both induce the same optimal scenario-based strategies $\overline{x}^s$ (see Lemma 1). We can now re-formulate the expected revenue as

$$\begin{aligned} R(x, (x^s, \alpha^s)_{s\in S}) &= \sum_{s\in S} p^s \left(\sum_{f\in F} r_f \left(\sum_{t=\hat{t}}^{t^s+1} x_{ft} + \sum_{t=t^s}^{0} x^s_{ft} \right) - \sum_{i=1}^{K} b_i \alpha^s_i \right) \\ &= \sum_{s\in S} p^s R_{[t_{i-1}, t_i - 1, c_i]}(x_{[t_{i-1}, t_i - 1]}) + \sum_{s\in S} p^s (\sum_{f\in F} r_f \sum_{t=t^s}^{0} x^s_{ft} - \sum_{i=1}^{K} b_i \alpha^s_i). \end{aligned}$$

If $\tilde{x}$ is not an optimal solution, there exists a period in time $[t_{i-1}, t_i - 1]$, such that

$$R_{[t_{i-1}, t_i - 1, c_i]}(\overline{x}_{[t_{i-1}, t_i - 1]}) > R_{[t_{i-1}, t_i - 1, c_i]}(\tilde{x}_{[t_{i-1}, t_i - 1]}).$$

This is a contradiction to the choice of $\tilde{x}$. Q.E.D.

Thus, solving the RMCU problem requires only finding the optimal number of tickets offered in every time period $[t_{i-1}, t_i - 1]$, $i = 1, \ldots, N$. The following algorithm computes these values as a longest path in an acyclic graph as illustrated by Figure 2.

For any RMCU instance, define the associated RMCU longest path instance as follows: Every vertex $(t, c) \in V$ in the corresponding graph $G = (V, A)$ consists of a tuple with $t \in T_b$ and $c \in \{0, \ldots, c_{\max}\}$, $c_{\max} = \max_{s\in S}\{c^s\}$. Add the source vertex $(t_0, 0)$ and the terminal vertex $(t_{N+1}, c_{\max})$ with $t_0 = \hat{t}$ and $t_{N+1} = -1$. The arc set A connects all pairs (t_i, c) and (t_{i+1}, c'), $i \in \{0, \ldots, N\}$, $c, c' \in \{0, \ldots, c_{\max}\}$, $c \le c'$. Now, associate a revenue $r((t_i, c), (t_{i+1}, c'))$ to an arc by

$$r((t_i, c), (t_{i+1}, c')) = \sum_{\substack{s\in S \\ t^s \le t_{i+1}}} p^s R(t_i, t_{i+1} - 1, c' - c) + \sum_{\substack{s\in S \\ t^s = t_{i+1}}} p^s R(t_{i+1}, 0, c^s - c')$$

and define $R(t_{i+1}, 0, c^s - c') = \sum_{j=1}^{c' - c^s} b_i$ if $c^s - c' < 0$. Finally, set $r((t_i, c), (t_{i+1}, c')) = 0$, if $t_{i+1} = t_{N+1}$. Any $((t_0, 0), (t_{N+1}, c_{\max}))$-path visiting the vertices $(t_1, c_1), \ldots, (t_N, c_N)$ represents a global strategy offering $c_i - c_{i-1}$ tickets during the period $[t_{i-1}, t_i - 1]$.

In the following, we prove that solving the RMCU longest path problem equals solving the RMCU problem.

Theorem 1. *A longest path in the RMCU longest path instance triggers an optimal global strategy and scenario-based strategies for an RMCU instance.*

Proof. Let q be a $((t_0, 0), (t_{N+1}, c_{\max}))$-path visiting the vertices (t_1, c_1), …, (t_N, c_N). The expected revenue for the associated global strategy $\overline{x}$, offering $c_i - c_{i-1}$ tickets during the period $[t_{i-1}, t_i - 1]$

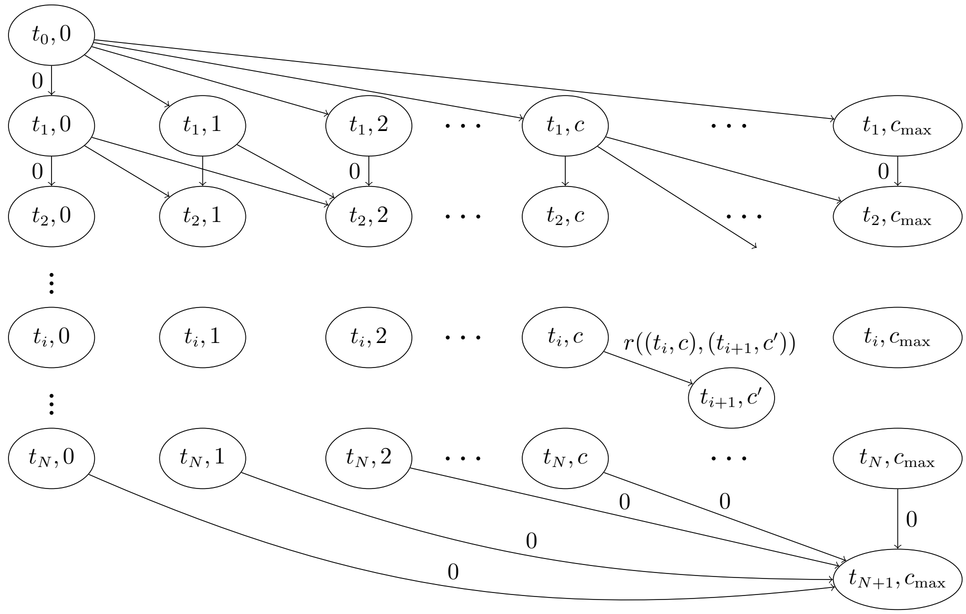

Figure 2 RMCU longest path instance

and the corresponding scenario-based strategies x^s, $s \in S$, or α^s respectively, equals the length $r(q)$ of this path q:

$$\begin{aligned}
R(x,(x^s,\alpha^s)_{s\in S}) &= \sum_{s\in S} p^s \left(\sum_{f\in F} r_f \left(\sum_{t=\hat{t}}^{t^s-1} x_{ft} + \sum_{t=t^s}^{0} x^s_{ft} \right) - \sum_{i=1}^{K} b_i \alpha^s_i \right) \\
&= \sum_{s\in S} p^s \sum_{\substack{t_i\in T_b \\ t_i \geq t^s}} R(t_i, t_{i+1}-1, c_i - c_{i-1}) + \sum_{s\in S} p^s \cdot R(t^s, 0, c^s - \max_{\substack{t_i\in T_b \\ t_i\geq t^s}} \{c_i\}) \\
&= \sum_{i=1}^{N} \sum_{\substack{s\in S \\ t^s \leq t_i}} p^s R(t_{i-1}, t_i - 1, c_i - c_{i-1}) + \sum_{i=1}^{N} \sum_{\substack{s\in S \\ t^s = t_i}} p^s \cdot R(t_{i+1}, 0, c^s - c_i) \\
&= \sum_{i=1}^{N} r((t_i, c_i), (t_{i+1}, c_{i+1})) = r(q)
\end{aligned}$$

Hence, finding a path with maximum length represents an optimal global strategy. Q.E.D.

Since the graph G is acyclic, a modified version of the Dijkstra algorithm can compute such a path in $\mathcal{O}(|A|)$. The number of arcs is bounded by $|T_b| \cdot c^2_{\max}$. Fischer and Helmberg (2014) propose to speed up solving this problem using the graph's special layer-structure.

The appendix includes an algorithmic representation of the combinatorial approach as implemented for the computational study.

3.4. Exploring the Role of Denied Boarding Cost

Clearly, the parameterization of the denied boarding cost plays an important role for the RMCU problem. Therefore, this section theoretically explores the effects of high versus low denied boarding cost. To that end, we consider a given instance of the RMCU problem at time $\hat{t} \in \mathbb{N}$. For each fare class $f \in F$, let r_f indicate the resulting revenue; for any point in time $t \in T$, let the expected demand be indicated by $D_{ft} \in \mathbb{N}$. For a given parameter vector of denied boarding cost $b \in \mathbb{R}^K$, let R_b^* denote the maximum expected revenue.

We start by considering the relation of two denied boarding cost vectors on their corresponding maximum expected revenue. The first lemma states that for higher boarding cost, the expected revenue decreases.

Lemma 3. *Let $b^1 \in \mathbb{R}^K$ and $b^2 \in \mathbb{R}^K$ be two different denied boarding cost vectors. If $b_i^1 \leq b_i^2$ for all $i \in \{1,\dots,K\}$, then $R_{b^2}^* \leq R_{b^1}^*$.*

Proof. Let x^i be an optimal global strategy for $b^i \in \mathbb{R}^K$, $i \in \{1,2\}$. Let (x^{is},α^{is}) be the corresponding optimal scenario-based strategies. From this, we obtain:

$$\begin{aligned}
R_{b^2}^* &= R(x^2,(x^{2s},\alpha^{2s})_{s\in S}) \\
&= \sum_{s\in S} p^s \left(\sum_{f\in F} r_f \left(\sum_{t=\hat{t}}^{t^s-1} x_{ft}^2 + \sum_{t=t^s}^{0} x_{ft}^{2s} \right) - \sum_{i=1}^{K} b_i^2 \alpha_i^{1s} \right) \\
&\leq \sum_{s\in S} p^s \left(\sum_{f\in F} r_f \left(\sum_{t=\hat{t}}^{t^s-1} x_{ft}^2 + \sum_{t=t^s}^{0} x_{ft}^{2s} \right) - \sum_{i=1}^{K} b_i^1 \alpha_i^{1s} \right) \\
&\leq \sum_{s\in S} p^s \left(\sum_{f\in F} r_f \left(\sum_{t=\hat{t}}^{t^s-1} x_{ft}^1 + \sum_{t=t^s}^{0} x_{ft}^{1s} \right) - \sum_{i=1}^{K} b_i^2 \alpha_i^{2s} \right) \\
&= R(x^1,(x^{2s},\alpha^{2s})_{s\in S}) = R_{b^1}^*.
\end{aligned}$$

This proves the lemma. Q.E.D.

Next, we consider properties of optimal global strategies, given very low or very high denied boarding cost. For low cost, an optimal global strategy offers all tickets at all points of time. Note that for $b_K > \min_{f\in F} r_f$, this is not true.

Lemma 4. *Let the maximum denied boarding cost b_K be lower than the revenue earned by the cheapest fare class, i.e., $b_K \leq \min_{f\in F} r_f$. Then, for a sufficiently large K, one optimal global strategy x^* offers all fare classes throughout the booking horizon, i.e., $x_{ft}^* = 1$ for all $f \in F$, $t \in T$.*

Proof. Assume that $\tilde{x}$ is an optimal global strategy, where $\tilde{x}_{f't'} = 0$ for $f' \in F$ and $t' \in T$. Consider a second strategy $\overline{x}$, where $\overline{x}_{f't'} = 1$ and $\overline{x}_{ft} = \tilde{x}_{ft}$ otherwise. Let $\tilde{x}^s$, $\overline{x}^s$ and $\tilde{\alpha}^s$, $\overline{\alpha}^s$ be the corresponding optimal scenario-based strategies. The change in the global strategy solely influences scenarios $s \in S$ with $t^s + 1 \leq t'$. We denote this set by $S_{t'}$, i.e., $S_{t'} = \{s \in S \mid t^s + 1 \leq t'\}$. Let $s \in S_{t'}$ be such a scenario. Since the number of tickets sold before t^s increases by 1, at most one more denied boarding results. Thus, denied boarding cost increase by at most α_K.

We then obtain

$$R(\overline{x},(\overline{x}^s,\overline{\alpha}^s)_{s\in S}) - R(\tilde{x},(\tilde{x}^s,\tilde{\alpha}^s)_{s\in S}) \geq \sum_{s\in S_{t'}} p^s\,(r_{f'} - b_K) \geq 0$$

due to the assumption that $b_K \leq \min_{f\in F} r_f$. Thus, selling all tickets is an optimal global strategy. Q.E.D.

Finally, we consider high denied boarding cost. Let x^s be an optimal strategy for scenario $s \in S$. We define an upper bound on the expected revenue for the considered instance: $R^{\max} = \sum_{s\in S} p^s \sum_{f\in F}\sum_{t\in T} r_{ft} x^s_{ft}$. For high denied boarding cost, an optimal global strategy never induces denied boardings in any scenario.

Lemma 5. *Let the cost of the first denied boarding exceed* $\max_{s\in S}\{\frac{1}{p^s}R^{\max}\}$. *If* x^* *is an optimal global strategy, then for the corresponding optimal scenario-based strategies* x^s *or denied boardings* $alpha^s$, $alpha^s = 0$ *holds.*

Proof. Let $\tilde{x}$ indicate an optimal global strategy. Let $s' \in S$ denote a scenario, where the number of sold tickets at time $t^{s'}$ surpasses $c^{s'}$. Then,

$$\begin{aligned} R(\tilde{x},(\tilde{x}^s,\tilde{\alpha}^s)) &\leq R^{\max} - p^{s'}\cdot b_1 \\ &< R^{\max} - p^{s'}\cdot \max_{s\in S}\{\frac{1}{p^s}R^{\max}\} \\ &\leq 0. \end{aligned}$$

Q.E.D.

This property allows us to improve the RMCU solution algorithm: Let $T_b := \{t \in T \mid \exists s \in S \text{ s.t. } t^s = t\} = \{t_1,\ldots,t_N\}$. Then, at most $c^{\min}_{i-1}$ tickets are sold during the interval of time $[t_{i-1}, t_i - 1]$ with $c^{\min}_{i-1} = \min_{j=i,\ldots,N}\{\min_{s\in S}\{c^s \mid t^s = t_j\}\}$. Thus, in the corresponding longest path instance, we can delete all nodes (t_i, c) with $c > c^{\min}_i$.

Note that the bound obtained in Lemma 5 is tight: Consider an instance with two points in time, offering revenues 1 and ε, and two scenarios with $c^0 = 1$ and $c^1 = 0$. Table 1 shows the revenue obtained in scenarios s_0 and s_1 as well as the scenarios' expected revenue $R()$. p_0 and p_1 indicate the respective scenario's probability. The first global strategy, indicated by $1-0$, accepts the first customer request but denies the second request. The second global strategy, indicated by $0-1$, denied the first customer request but accepts the second request.

Let $0 < p_1 < p_0$ and set $b_1 = \frac{1}{p_1}\cdot R^{\max} = \frac{1}{p_1}\cdot(1p_0)$. Then

$$\begin{aligned} R(1-0) &= p_0 + p_1(1-b_1) = p_0 + p_1 - p_1\cdot b_1 \\ &= p_0 + p_1 - p_1\cdot\frac{p_0}{p_1} = p_1 \end{aligned}$$

Hence, in this instance and for $p_1 < 0.5$ and $p_0\cdot\varepsilon < p_1$, the strategy inducing a denied boarding in scenario s_1 earns more expected revenue than the strategy inducing no denied boardings.

Table 1 Exemplary RMCU instance

Global Strategy	Revenue in s_0	Revenue in s_1	Expected revenue $R()$
1-0	1	$1-b_1$	$p_0+p_1(1-b_1)$
0-1	ε	0	$p_0\cdot\varepsilon$

3.5. Exploiting the Special Case of a Single Capacity Update

Finally, we consider the special case of a single capacity update, where the final capacity is announced at a single point in time t_1. As before, we model the set of possible resulting capacities by a set of scenarios S. Note that a single update means that $t^s = t_1$ for any scenario $s \in S$. If no change is announced, the flight departs with the original capacity of c^0. For simplicity we do not model this as an additional scenario. The probability of change is given by $p := \sum_{s\in S} p^s$ with $p \leq 1$. Using Lemma 1 and Lemma 2, we can model the expected revenue as a function $R(c)$ depending on the capacity c used until time t_1, $c \geq 0$. This function is given by

$$R(c) := R(\hat{t}, t_1 - 1, c) + \sum_{s\in S} p^s \cdot R(t_1, 0, c^s - c) + (1-p)\cdot R(t_1, 0, c^0 - c).$$

The objective of the RMCU problem with a single update is to maximize $R(c)$. This function $R(c)$ extends the salvage function introduced by Wang and Regan (2006) in two ways: Firstly, it considers more than two scenarios and secondly, it accounts for revenue obtained during time period $[\hat{t}, t_1 - 1]$. Finally, as opposed to the salvage function from Wang and Regan (2006), $R(c)$ is not monotonically increasing and concave in c. However, we can prove that $R(c)$ is piecewise linear and concave, as argued by the following Lemma 6. We exploit these properties to obtain an efficient algorithm to solve the RMCU problem.

Lemma 6. *The function $R(c)$ is piecewise linear and concave.*

Proof. $R(c)$ is the weighted sum over the functions $R(\hat{t}, t_1 - 1, c)$, $R(t_1, 0, c^0 - c)$ and $R(t_1, 0, c^s - c)$, $s \in S$. These functions are piecewise linear and concave since their derivative is monotonically decreasing. Since the sum of concave functions leads to a concave function and the same holds for the piecewise linearity, we have proven these properties. Q.E.D.

Figure 3 illustrates the proof given above: Function $R(\hat{t}, t_1 - 1, c)$ is monotonically increasing by increasing values of the capacity c. However, the other functions $R(t_i, 0, c^i - c)$ are monotonically decreasing. The values of the functions are plotted on the y-axis according to the different values of c plotted on the x-axis. The capacity in scenario 1 is lower than the original capacity c^0 and the capacity in scenario 2 is higher. If $c^s - c$ is negative, the function $R(\cdot)$ represents the denied boarding cost.

A straightforward approach to finding the value c^* that maximizes $R(c)$ is to evaluate the function for all $c \in \{0, \ldots, c_{\max}\}$ with $c_{\max} = \max_{s\in S}\{c^s\}$. However, we can improve the computational

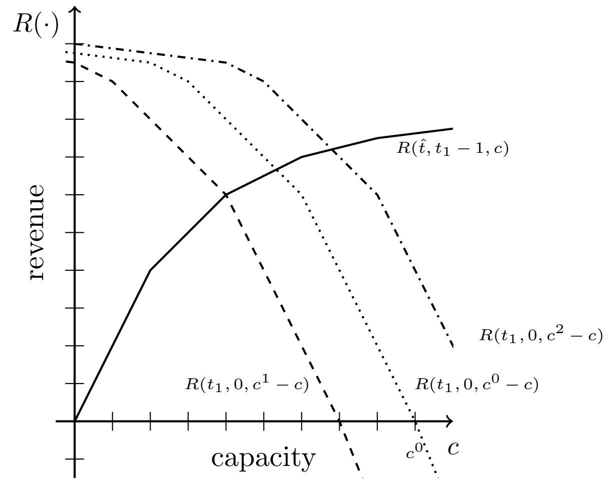

Figure 3 **Revenue functions $R(\cdot)$ depending on capacity c**

efficiency as follows. Firstly, $R(c) = -\infty$ for $c > \min_{s\in S}\{c^s + K\}$, since at most K denied boardings are allowed. Secondly, we need to consider only capacities where one of the functions $R(\hat{t}, t_1 - 1, c)$, $R(t_1, 0, c^s - c)$ or $R(t_1, 0, c^0 - c)$, $s \in S$ changes slope. In the simplest case, the capacity c exceeds the capacity of any scenario c^s and the K following values due to the non-linear denied boarding costs, i.e., $c = c^s$, $s \in S$ and $c = c^s + \ell$, $\ell \in \{1, \ldots, K\}$. Furthermore, the function's slope changes if all tickets are offered in one fare class and the strategy is switched to offering tickets to the next cheaper fare class.

Formally, we obtain these values by defining D_f^1 as the cumulative demand of the fare class $f \in F$ from $\hat{t}$ to $t_1 - 1$, i.e., $D_f^1 := \sum_{t=\hat{t}}^{t_1-1} D_{ft}$, and D_f^0 as the cumulative demand of the fare class $f \in F$ from t_1 to 0, i.e., $D_f^0 := \sum_{t=t_1}^{0} D_{ft}$. Let the fare classes be ordered according to their value, i.e., $r_1 \geq \ldots \geq r_{|F|}$. Then the slope changes if $c = \sum_{f=1}^{\tau} D_f^1$, or $c^s - c = \sum_{f=1}^{\tau} D_f^0$, $\tau = 1, \ldots, |F|$, $s \in S$. Just considering these values for the capacity used until time t_1 guarantees an optimal solution.

Lemma 7. *There always exists a capacity c^* maximizing the function $R(c)$ out of the following set C,*

$$
\begin{aligned}
C := \{0\} &\cup \{c \in \mathbb{N} \mid c^s = c + \ell, s \in S, \ell \in \{0, \ldots, K\}\} \\
&\cup \{c \in \mathbb{N} \mid c = \sum_{f=1}^{\tau} D_f^1, \tau = 1, \ldots, |F|\} \\
&\cup \{c \in \mathbb{N} \mid c^s - c = \sum_{f=1}^{\tau} D_f^0, \tau = 1, \ldots, |F|, s \in S\}.
\end{aligned}
$$

Proof. Since function $R(c)$ is piecewise linear and concave, an optimal solution will be found at one of the points where the function's slope changes. This happens if one of the functions $R(\hat{t}, t_1 - 1, c)$, $R(t_1, 0, c^0 - c)$, or $R(t_1, 0, c^s - c)$, $s \in S$ changes their slope. These points define the set C. Q.E.D.

The set C contains at most $1+|S|+K+|F|+|S|\cdot|F|$ different values. Using a binary search, we just need to compute for $\log(|C|)$ many different points $c \in C$ the function values $R(c)$ and $R(c+1)$. The evaluation of $R(c)$ and $R(c+1)$ determines whether the function is increasing or decreasing. If $R(c)-R(c+1)>0$, we change the lower interval bound of the considered interval to its midpoint. If $R(c)-R(c+1)<0$, we change the upper interval bound of the considered interval. Finally, if $R(c)-R(c+1)=0$ or the interval bounds match, the value of c is optimal. $R(c)$ can be computed in linear time depending on the number of fare classes through the greedy algorithm and some preliminary sorting. Thus, runtime is $\mathcal{O}\left(\log(1+|S|+K+|F|+|S|\cdot|F|)\cdot\max\{|S|,|F|\}\right)$. This is much more efficient than solving the longest path problem in roughly $\mathcal{O}(c_{\max}^2\cdot\hat{t}^2)$.

The ideas used to obtain faster algorithms for the case of a single capacity update cannot be trivially extended to the case of multiple change events. First of all, it is not clear whether the expected revenue function, which in the latter case depends on several variables, remains concave. If the expected revenue function is not concave, no binary search can be performed. Furthermore, the set of relevant capacity values, in the single update denoted by C, is much more complicated to describe.

3.6. RMCU and Static Overbooking

This subsection highlights the close connection of the RMCU problem to a simple static overbooking problem aiming to compensate for no-shows. Consider an overbooking model that assumes perfect knowledge on the expected demand $D_{ft}\in\mathbb{N}$ in each fare class $f\in F$ to any point in time $t\in T$, on the corresponding revenue r_f, and on the physical capacity c^0. At $t=0$, the flight departs, but a certain number of passengers do not show up and therefore do not utilize capacity.

For every fare class i, we assume a given probability β_i of an accepted customer showing up and requiring a seat. Then, static overbooking aims to define a total authorized capacity b. Given this authorized capacity, revenue management can calculate the number of customers to accept per class.

To demonstrate the relationship of uncertain capacity and overbooking, we transform the static overbooking problem into an RMCU problem. Let c^0 be the physical capacity. For a number of scenarios N, we define alternative expected numbers of no-shows $n_1,\ldots,n_N\in\mathbb{N}$. For instance, given overall show-up probability β, we may compute the probability p_j that n_j no-shows happen, $j\in\{1,\ldots,N\}$. According to these values, we can transform the uncertain number of no-shows to a scenario setting. For each scenario $s_j\in S$, the update time is 0 and the final capacity is $c^{s_j}=c^0+n_j$. The probability of scenario $s_j\in S$ is given by $p^{s_j}=p_j$.

As the resulting strategy accounts for the cost of denied boardings, it will obtain more expected revenue than the strategy of not overbooking at all or of statically overbooking for the expected average number of no-shows (termed EMSR/NO and EMSR/MD respectively in Aydin et al. (2012)). Furthermore, as opposed to the static approaches EMSR/Risk and EMSR/SL proposed in Aydin et al. (2012), it does not necessarily assume a binomial distribution of no-shows.

The RMCU problem for overbooking is a special case of the RMCU with a single capacity update. Thus, the algorithm proposed in Subsection 3.5 can solve the problem. However, in the following, we propose an alternative greedy algorithm. While this alternative comes at potentially higher computation cost, its simple nature and close relationship to the greedy-algorithm for the deterministic revenue management problem render it interesting.

The idea of the algorithm is to iteratively decide on whether to offer one more ticket. This decision solely depends on a so-called *relative revenue* and not on the capacity limit c^0. A ticked is sold as long as this relative revenue is positive. More formally, assume $\tilde{c}$ tickets to be already sold. Let $r^{\max}$ represent the revenue of the most expensive fare class, for which not all tickets have been sold yet. Then, compute the relative revenue $\tilde{r}$ by including the potential denied boarding cost and obtain

$$\tilde{r} = r^{\max} - \sum_{\substack{s \in \mathcal{S} \\ \tilde{c} \geq c^s}} p^s b_{\tilde{c}+1-c^s}.$$

If $\tilde{r} > 0$, offer the ticket, increase $\tilde{c}$ by one and update the relative revenue. If $\tilde{r} \leq 0$, offer no further tickets. This leads to an optimal strategy, which can be easily shown by a proof of contradiction.

We consider a set of relevant instances as a numerical example in Section 4. Of course, we are aware of much more sophisticated published approaches to overbooking such as described in Aydin et al. (2012) and Topaloglu et al. (2012). Nevertheless, we conclude that this detail and a more thorough comparison to common overbooking methods may be interesting for future research.

Finally, while the discussion above showed that solution strategies for the RMCU problem can be applied to static overbooking problems, this cannot be trivially reversed: The RMCU problem cannot be solved by applying existing overbooking strategies. Overbooking models commonly view cancellations and no-shows as occurring in increments of one, excluding group cancellations. However, the RMCU problem considers capacity updates to occur in arbitrary increments, as assigning a new aircraft may increase or decrease capacity by any multiple of one seat.

4. Computational Study

This section describes the results of a computational study used to evaluate RMCU solution performance given varying problem characteristics. First, it documents the experimental setup. Subsequently, it introduces the implemented benchmark strategies. Finally, it presents the study's numerical results and highlights implications for the model's behavior.

Table 2 Fare class value and demand

Fare Class	1	2	3	4	5	6	7	8	9
Value	1.00	0.78	0.65	0.53	0.41	0.31	0.22	0.16	0.12
Demand	7%	8%	5%	6%	10%	8%	16%	25%	15%

4.1. Experimental Set Up

In accordance with the leg-based model, all evaluated problem instances consider a single flight and a single compartment. Therefore, every possible final capacity can be described by a single parameter, namely the number of available seats. The initial capacity is identical for all instances and is set to 100. Four alternative final capacities (50, 90, 110, 150) result from a capacity change increasing or decreasing the initial capacity by 10% or 50%.

Over a booking horizon of 360 days, nine fare classes are offered within the single compartment. When calibrating the experimental setup, this number of fare classes was found sufficiently large to observe relevant effects without unnecessarily increasing complexity. Revenue per fare class increases exponentially from the cheapest class, r_9, to the most expensive class, r_1. The value of each fare class is listed in Table 2. This table expresses revenue not in absolute currency but in terms of relative price indeces. Fares are calibrated based on empirical airline data describing the average revenue earned per class in the economy compartment.

Both the distribution of request arrivals over the booking horizon and that of requests over fare classes are parameterized to fit empirical distributions. The resulting demand arrival distribution implements the idea of low-value demand arriving before high-value demand, but allows for overlaps. Given an independent demand model, every booking request exclusively targets one fare class. 43% of customers request earlier than 200 days before departure, but they only book tickets for the four cheapest classes. From 201 to 50 days before departure, another 21% of customers request. From 50 days before departure until departure, the final 36% of customers request, focusing mostly on the four most expensive classes. The demand share per fare class is listed in Table 2. There are no cancellations or no-shows, i.e., all bookings turn into passengers expecting to board.

The denied boarding cost calculation is based on input from industry experts: In practice, the airline does not decide whom to deny boarding, but allows passengers to volunteer. To approximate the resulting cost, the model is parameterized to assume the first denied boarding will incur costs that correspond to the average yield expected based on the flight's demand forecast. Therefore, the cost of the first denied boarding depends on the demand scenario; in terms of fare indices, it varies between 0.352 and 0.356. With every further denied boarding, the cost increases by a factor

Table 3 Parameterization of problem instances

Parameter	Levels	Values
Demand level	3	60%, 120%, 180%
Update probability $\sum_{s\in S} p^s$	21	20%, 21%, ..., 40%
Magnitudes of change	4	+10%/−10%, +50%/−10%, +10%/−50%, +50%/−50%
No. of update times $\lvert T_b \rvert$	2	5, 10
Update times T_b	3	200-150, 200-0, 50-0 days before departure
Probability ratios of change	3	1:3, 1:1, 3:1 (positive:negative)

of 1.1. Consequentially, from the 11th denied boarding on, the cost surpasses the value of the most expensive class.

Note that in contrast to demand and fare class value, capacity changes were not calibrated to fit empirical data. Instead, we decided for a systematic variation to explore the model's behavior. As a result, the general study spans 4,536 test instances. Table 3 describes the parameterization of problem instances, indicating the relevant notation and the number of variation levels. Instances were created for three levels of demand in percent of capacity, 21 levels of overall change probability, four combinations of change magnitudes, two potential numbers of change events, three degrees of change timing, and three combinations of scenario probability. For the purpose of this study, we assume perfect knowledge of both expected demand and scenario probabilities.

In addition to the RMCU problem, we implemented two benchmark approaches: BLIND represents an industry standard approach of defining the global strategy according to the initially planned capacity; EX POST calculates an upper revenue bound by using perfect information on whether a capacity change will actually occur to implement the best deterministic strategy.

4.2. Bounds on the expected revenue

To evaluate and benchmark the newly introduced RMCU problem and its solution, we develop an upper and lower bound for the expected revenue. The upper bound, EX POST, is based on a perfect knowledge of the final capacity (as available only *ex post*, in hindsight) and the lower bound, BLIND, on a planner ignoring (being *blind* with regard to) potential future capacity updates.

EX POST uses perfect hindsight information on the final capacity. Based on this, it chooses a revenue optimal strategy x^{*s} to match the realized scenario $s \in S$, which maximized the revenue according to the capacity c^s. The resulting expected revenue is

$$R(x^{*s}, s \in S) = \sum_{s\in S} p^s \Big(\sum_{t=\hat{t}}^{0} \sum_{f\in F} x^{*s}_{ft}\Big).$$

This is an upper bound on the optimal solution of RMCU: In every scenario $s \in S$, the revenue obtained by combining the global and the scenario-based strategy is smaller or equal to the revenue obtained by x^{*s}.

BLIND neglects potential capacity updates and defines the global strategy $\overline{x}$ according to the initially announced capacity. Note that this strategy adapts whenever a capacity update is announced, by re-optimizing the allocation of the newly available capacity according to the demand expected from that time on. Since this global strategy is based on a single scenario, it represents a feasible solution to the RMCU problem. Accordingly, the expected revenue for BLIND is smaller or equal to the optimal RMCU solution.

Note that BLIND may be arbitrarily bad compared to RMCU. To show this, consider a booking horizon of $\hat{t} = n+2$, $n \in \mathbb{N}$. In the first n points in time, revenue equals ε, $\varepsilon > 0$. Two days before departure, no customer requests, but one day before departure a request with a revenue of 1 arrives.

Now, consider two scenarios s_0, s_1 where s_0 is the basic scenario. The scenario s_1 has a capacity of $c^{s_1} = 1$ and an update probability of $p^{s_1} > \varepsilon$ two days before departure, i.e., $t^{s_1} = 2$. The basic scenario has a capacity of $c^{s_0} = n+1$ with a probability $p^{s_0} = 1 - p^{s_1}$. Finally, let denied boarding cost b_i be 1 for $i = 1, \ldots, \hat{t}$.

Table 4 compares the expected revenue resulting from BLIND and a *SIMPLE* strategy, which offers a seat to the last customer:

Table 4 Exemplary expected revenue from BLIND and SIMPLE

Global Strategy	Revenue in s_0	Revenue in s_1	Expected Revenue
BLIND	$\varepsilon \cdot n + 1$	$\varepsilon \cdot n - (n-1) \cdot 1$	$\varepsilon \cdot n + p_0 - p_1(n-1)$
SIMPLE	1	1	1

Next, consider the absolute gap between the expected revenue of BLIND and RMCU, as denoted by $R(\text{BLIND})$ and $R(\text{RMCU})$. Since the expected revenue obtained by SIMPLE is a lower bound on the expected revenue of RMCU, the absolute gap is lower bounded by

$$R(\text{RMCU}) - R(\text{BLIND}) \geq 1 - (\varepsilon \cdot n + p_0 - p_1(n-1)) = n \cdot (p_1 - \varepsilon).$$

For large n, the right hand side tends linearely towards infinity. Hence, the gap may be arbitrarily large.

4.3. Numerical Results

This section first considers the revenue increase from the BLIND to the RMCU solution across all instances. Subsequently, we consider revenue as depending on update probability, timing, direction, and on the relationship between demand volume and the updates' direction.

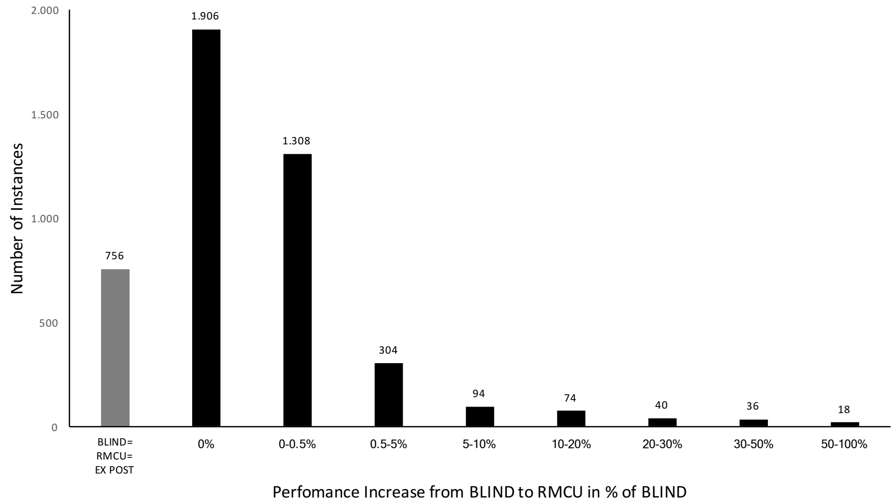

Figure 4 **Performance gap between RMCU and BLIND**

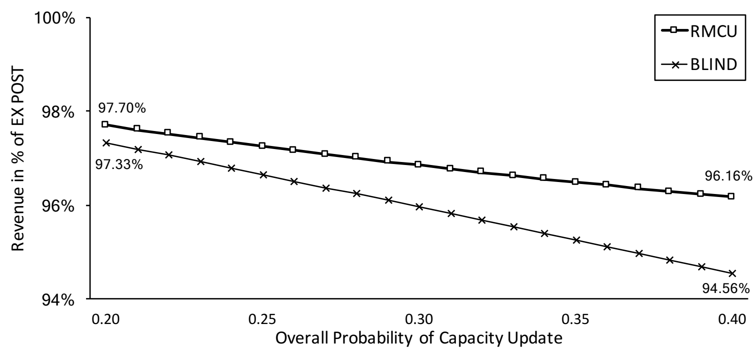

Figure 5 **Performance of RMCU and BLIND approaches given different degrees of overall update probability**

Figure 4 shows the performance gap between the RMCU solution and BLIND in percent of the latter. In 756 instances (17%), all solutions perform equally well. Analysis shows that this group includes no instances including a possible capacity change of -50%. In 1,906 instances (42%), RMCU and BLIND perform equally well, but inferior to EX POST. There appears to be no clear pattern for this behavior. For 1,874 instances (41%), RMCU is clearly superior to BLIND. RMCU's advantage is below 0.5% in 1,308 instances (29%), between 0.5% and 5% in 304 instances (7%) and even larger for the remaining 262 instances (5%).

To evaluate the influence of the overall update probability, we tested 21 probability levels ranging from 20% to 40% in steps of 1%. Figure 5 shows the results by plotting the revenue gained from RMCU and BLIND in terms of the upper revenue bound computed by EX POST. The performance gap between RMCU and BLIND increases with the update probability. This is intuitive, as the information advantage of the stochastic approach increases in these cases. Note that the relationship between update probability and solution performance is almost, but not quite, linear. In all further analyses, we consider an overall update probability of 40%.

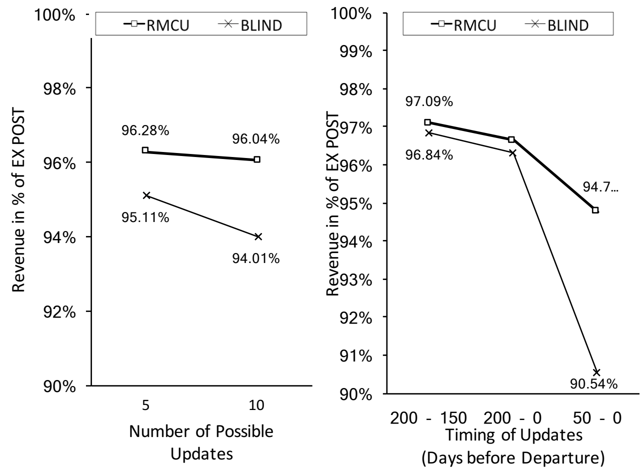

Figure 6 Performance of RMCU and BLIND approaches given differences in potential update timing

Figure 6 illustrates the performance of RMCU and BLIND in percent of EX POST when the timing of capacity updates varies. We vary both the number of possible update times and their timing within the booking horizon. In the left panel, the x-axis depicts two potential numbers of update times, 5 and 10. Both solutions perform better with a lower number of potential update times. However, BLIND's decline in performance given an increasing number of update times is significantly steeper than that of RMCU. We conclude that, given accurate scenario probabilities, the number of scenarios barely affects RMCU performance. However, we expect that for an increasing number of possible update times, estimating accurate scenario probabilities becomes more difficult.

In the right panel of Figure 6, the x-axis depicts alternative update timing through three time intervals: Early updates are uniformly distributed between 200 and 150 days before departure; intermediate updates are uniformly distributed between 200 and 0 days before departure; late updates are uniformly distributed between 50 and 0 days before departure. Our results show that more revenue is lost when updates occur late in the booking horizon. However, RMCU does not seem to be as vulnerable to this effect as BLIND: Whereas BLIND's revenue gap to EX POST increases by 6.3 percent points, RMCU's revenue gap increases by only 2.3 percent points.

Figure 7 illustrates the performance of RMCU and BLIND in terms of EX POST when the quality of capacity updates is varied. In the left panel, the x-axis depicts four potential sets of relative differences between the initial capacity and the alternative capacities, assuming that three final capacities are possible. The analysis considered changes of $+10\%$ vs. -10%, $+50\%$ vs. -10%, $+10\%$ vs. -50%, and $+50\%$ vs. -50%. Intuitively, a larger magnitude of change will degrade revenue performance for both compared approaches. When capacity increases, employing RMCU does not make a difference – this is due to the independent demand model and lack of capacity cost considered in this study. However, when capacity strongly decreases, the RMCU solution's

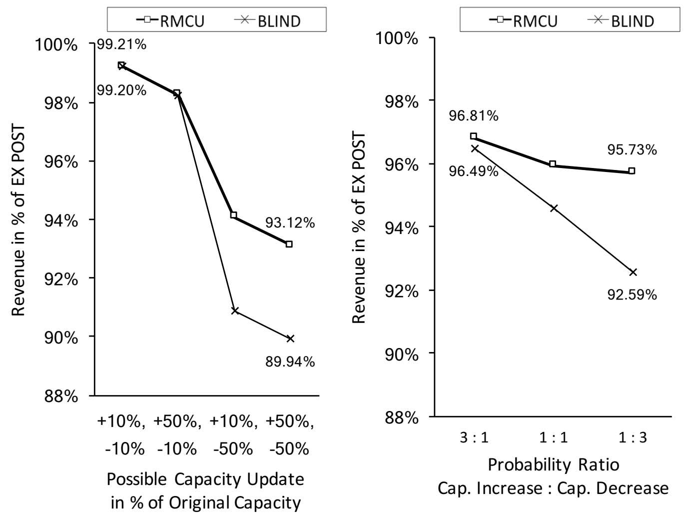

Figure 7 Performance of RMCU and BLIND approaches given differences in the quality of capacity updates

advantage over BLIND increases. This is due to potential denied boardings being anticipated by RMCU, so that the resulting denied boarding costs can be minimized.

The right panel of Figure 7 confirms this finding: Here, the magnitude of updates is described by a probability ratio of increase vs. decrease. The x-axis represents three probability ratios of a change increasing the capacity to a change decreasing the capacity: 3:1, 1:1, and 1:3. In line with the previous results, a high probability for a capacity decrease increases the advantage of RMCU.

Figure 8 considers solution performance in terms of the relationship between the ratio of demand volume and capacity. The left panel splits the set of instances by the ratio of demand volume to initial capacity by 60%, 120%, and 180% along the x-axis. Note that the distribution of requests over fare classes is identical across instances. When the demand ratio is low, RMCU only slightly outperforms the BLIND solution. This is due to both solutions accepting all potential requests in most instances, capacity increases rarely exceed 40%. As this study models demand as independent, all inventory controls are motivated by scarce capacity. We expect divergent results for models that include dependent demand.

RMCU clearly outperforms BLIND, by 3.64 percent points, when demand slightly exceeds the initial capacity. According to our information, this is the most frequently observed situation in practice. For very high levels of demand, the performance gap between RMCU and BLIND decreases again, to 1.09 percent points. In these cases, an unexpected capacity increase can be compensated even by BLIND, as the number of future requests remains high.

Finally, the right panel of Figure 8 depicts instances experiencing desirable versus those with undesirable capacity updates. From the point of view of revenue management, updates are desirable if they improve the ratio between the expected demand and the final capacity. If demand exceeds

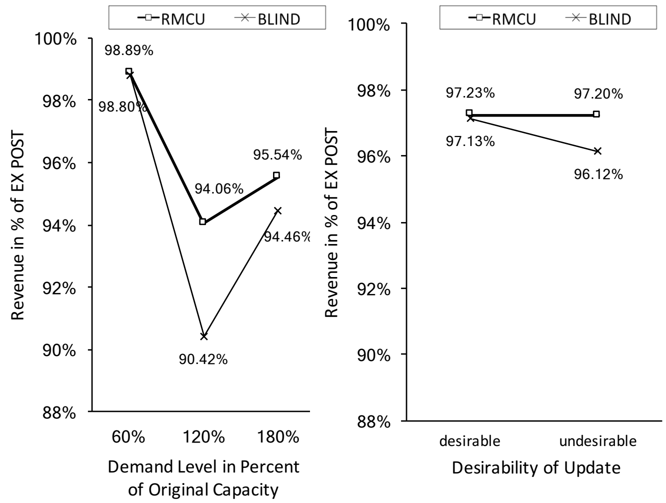

Figure 8 **Performance of RMCU and BLIND approaches given the relationship of demand volume and capacity**

capacity, revenue management calls for a capacity increase. Therefore, for high demand volumes, capacity increases are more desirable than decreases; capacity decreases are undesirable. For low demand volumes, capacity decreases are desirable while capacity increases are undesirable.

In accordance with our expectations, the gap between RMCU and BLIND is particularly large (1.08 percent points) when undesirable capacity updates are probable. The gap grows further when excluding those instances where RMCU and BLIND achieve the same revenue – however, for the sake of consistency, these instances are included in all results given here.

When the update is undesirable, the stochastic solution prepares a strategy that alleviates the negative effects, i.e. spoilage or spill. When updates are most likely of a desirable quality, RMCU and BLIND perform equally well in the vast majority of instances. Note that such desirable updates are the objective of research aiming to further the integration of fleet assignment and revenue management. Our final result therefore strengthens the motivation for research on compensating undesirable exogenous capacity changes.

4.4. Computational Performance

In Section 3.3, we propose a combinatorial algorithm as a computationally more efficient approach to solving the RMCU problem. Here, we compare this algorithm's run time to the run time required by the conventional solver CPLEX when optimizing the MIP-formulation given in Section 3.2.

Run times were recorded when solving a set of benchmark instances on a 64-bit Windows 10 system with an Intel Core i7-7500U CPU with 2.7 GHz and 16 GB RAM. In line with the previously discussed setup, the instances feature demand volumes that randomly vary from 60% to 180%. Customer arrivals are uniformly distributed across the booking horizons and over fare classes. New capacities vary from 50% to 150% of the initial capacity. Change times are drawn from a uniform distribution over the booking horizon of 360 days. Instances differ in the number of potential changes (2, 4, 8, or 16) and in the number of potential new capacities (also 2, 4, 8, or 16). Thus, in the most extreme case, an instance can include 257 scenarios (16×16 combinations of new capacity and change time in addition to the original capacity). The initial capacity can be 100, 250, 500, 750, or 1,000. We also vary the number of fares by considering ten alternative fare structures, which include ten to 100 fares in steps of ten. Fares always range from 100 to 1,000, so that fare differences shrink with an increasing number of fares. Thus, we differentiate 800 distinct variants of scenario set, initial capacity, and fare structure. For each variant, we draw 100 demand instances, arriving at 80,000 problem instances.

To exclude implementation issues, we benchmark the combinatorial algorithm's run time on that of the CPLEX function solve(). Thereby, we exclude the time required to set up the CPLEX model. In our implementation, this accounts, on average, for 79% of the overall observed CPLEX run time. While there may be more efficient approaches to setting up a CPLEX model, this is not a negligible effort. Excluding it renders the terms of the benchmark more favorable for CPLEX.

All instances were solved to optimality by both CPLEX and the combinatorial algorithm, each instance within 324 seconds. However, optimizing the MIP requires, on average, 330 times longer than using the combinatorial algorithm to find the optimal solution. For 56,950 out of 80,000 instances, the combinatorial algorithm requires less than a millisecond of run time, while CPLEX still requires an average of 800 milliseconds. Table 5 shows the maximum run time of the two solution approaches per percentile of analysed instances. This highlights that in 90% of instances, the combinatorial algorithm required at most 16 milliseconds, whereas the optimization via CPLEX achieved a runtime of less than 79 milliseconds for only 10% of instances. Furthermore, the percentile distribution also highlights that the increase in runtime is largest for the top 10% of instances in both cases, with the maximum CPLEX run time being more than 100 times that of the maximum measured for 90% of instances.

The run time of both solution approaches mostly increases with the number of scenarios, as shown in Table 6. However, the increase is much steeper for CPLEX. To highlight this, we provide a runtime ratio, dividing the runtime of the combinatorial algorithm (COMB) by that of solving CPLEX. This runtime ratio is nearly constant for five and nine scenarios, but it decreases steadily when 17 or more scenarios have to be considered. It is smallest for the maximum of 257 scenarios, where for every millisecond of runtime that the combinatorial algorithm requires, solving CPLEX requires 814 milliseconds. From this analysis, we conclude that run time improvements achieved via the combinatorial algorithm is more appealing for revenue management practice, where many flights have to be optimized and re-optimized in a limited span of time.

Table 5 Maximum runtime of solving via CPLEX and combinatorial algorithm (COMB), measured in milliseconds, per percentile of instances

Percentile of instances	Runtime CPLEX	Runtime COMB
0.1	78	0
0.2	141	0
0.3	234	0
0.4	344	0
0.5	485	0
0.6	691	0
0.7	1,031	0
0.8	1,625	15
0.9	3,078	16
1.0	323,495	141

Table 6 Average runtime of solving via CPLEX and combinatorial algorithm (COMB), measured in milliseconds, per number of scenarios in the instances

Number of scenarios	Runtime CPLEX	Runtime COMB	Runtime Ratio
5	74	1	1 : 74
9	146	2	1 : 73
17	293	3	1 : 98
33	623	4	1 : 156
65	1,377	6	1 : 230
129	3,515	9	1 : 391
257	12,207	15	1 : 814

4.5. Computational Consideration of Static Overbooking

As an addendum to the general study, we include computational results illustrating the relationship to static overbooking. To this end, we adapt parts of the experimental setup described in Aydin et al. (2012) to create four combinations of demand and show-up rates: For a capacity of 100 seats, we consider two levels of demand, expressed as a percentage of capacity, namely, 140% and 180%. We consider two degrees of show-up probability as dependent on the booked class. Extending the example set by Aydin et al. (2012) to nine fare classes, low show-up rates are {0.95, 0.93, 0.91, 0.89, 0.83, 0.81, 0.79, 0.77, 0.75}, while high show-up rates are {0.98, 0.96, 0.94, 0.92, 0.86, 0.84, 0.82, 0.80, 0.78}. For each combination of demand level and show-up rates, we draw 1,000 random instances of demand from uniform distributions over fare classes and days before departure.

From the expected demand and show-up probabilities, we compute the average show-up rate β. We iteratively consider scenarios with growing authorized capacity by deriving the expected number of customers that can be expected without denied boardings from the binomial distribution. $B(\beta, b)$ yields the number of passengers expected to show up given an authorized capacity b. To each increase in authorized capacity i, starting with $i = 0$ for the initial capacity $c^0 = 100$, we assign probability $p_i = \beta\mathbb{P}(B(\beta, c^0 + i) \geq c^0)) - \sum_{j=0}^{i-1} p_j$. Subsequently, we only consider scenarios with $p_i \geq 0.001$, and normalize the scenario probabilities to ensure $\sum_i p_i = 1$.

We benchmark the overbooking based on RMCU on the four EMSR-based alternatives described in Aydin et al. (2012). Dropping the prefix "EMSR", we compare RMCU to:

NO: A strategy that does not allow for any overbooking.

MP: A strategy that deterministically converts the expected average show-up rate into an authorized capacity as described in Belobaba (2006).

SL: A strategy that computes the authorized capacity to limit the probability denied boardings to a given δ, as denoted as Service Level 1 in (Talluri and van Ryzin 2004, p. 141). We set $\delta = 10^{-3}$.

Risk: A strategy that explicitly computes the authorized capacity to limit the probability denied boardings to a dynamic bound depending on the denied boarding cost.

From the binomial distribution, *Risk* and *SL* calculate the probability γ_b that the number of customers showing up exceeds capacity c^0 given authorized capacity b, i.e., $\gamma_b = \mathbb{P}[B(\beta, b) \geq c^0]$.

SL assumes a bound δ on γ_b as given. In contrast, *Risk* dynamically defines this bound by $\mu_0 = \theta \cdot \beta$, where θ denotes the denied boarding cost. For increasing denied boarding costs, we adapt this formulation by considering the cost caused by denied boarding $i = b - c^0$, $\mu_0 = \theta \cdot (1.1)^i \cdot \beta$.

For the authorized capacity b resulting per overbooking approach, we calculate the optimal revenue $r(b)$. Subsequently, we derive the expected show-up probability $\hat{\beta}$ of the resulting accepted bookings. To calculate expected denied boardings, we compute the expected probability of bookings that exceed the physical capacity becoming the n-th denied boarding from the binomial distribution. Finally, we compute the expected revenue by subtracting the corresponding denied boarding cost multiplied by that probability from the optimal revenue from bookings: $\hat{r}(b) = r(b) - \sum_{i=0}^{c_0-b}(\sum_{j=0}^{i}(\mathbb{P}[B(\hat{\beta}, b) = c_0 + j] \cdot \theta \cdot (1.1)^j))$.

Table 7 lists the expected revenue per approach as a percentage of revenue expected from the no-overbooking approach *NO*. The highest revenue per row is marked in bold. Note that *NO* constitutes a lower bound for revenue from *RMCU*. Other approaches can perform inferior to *NO* when the denied boarding costs exceed the revenue gained from additional accepted bookings. This is clearly true for *MP*, which consistently performs worse than not overbooking at all, as it incurs excessive denied boarding costs. In comparison, *LS*, *Risk*, and *RMCU* increase revenue by two to three percent. These gains grow with the demand volume. While *LS* and *Risk* perform slightly better for high show-up rates, *RMCU* earns slightly more revenue for low shop-up rates. Overall, these results underline the close relation of RMCU to overbooking when considering incremental capacity changes.

5. Conclusion

This paper was motivated by the idea that in the airline industry, flight capacities are frequently updated over the booking horizon. This violates the common revenue management assumption of

Table 7 Revenue from Static Overbooking Approaches in % of NO

Demand	Show-Up Rate	MP	SL	Risk	RMCU
140%	low	95.40%	102.28%	102.29%	**102.47%**
	high	95.41%	102.34%	**102.35%**	102.34%
180%	low	95.34%	103.16%	103.17%	**103.39%**
	high	95.58%	103.17%	103.19%	**103.38%**

a fixed capacity. Triggers are events such as fleet assignment re-optimizations and technical issues. The possibility of capacity updates causes uncertainty that is not considered by standard revenue optimization models.

Two types of revenue management models already consider uncertainty as related to capacity: Those considering overbooking and those considering integrating fleet assignment and revenue management. However, the latter predominantly focus on capacity adjustments that are fully controlled by revenue management. Here, we termed the resulting capacity changes as *endogenous*, to delineate them from the *exogenous* changes that are not triggered by revenue management.

In this paper, we extended a revenue optimization model anticipating aircraft swaps as described by Wang and Regan (2002, 2006). We proposed the quantity-based revenue management under capacity uncertainty (RMCU) problem, which considers multiple possible update times and multiple possible final capacities through stochastic scenarios. Instead of focusing on swaps triggered to support revenue management, we explicitly account for capacity updates that may not be aligned to revenue maximization.

As a theoretical contribution, we proposed computationally efficient ways of solving the RMCU problem. To that end, we described both a regular mixed-integer program, a combinatorial solution approach, and a solution approach that exploits particular characteristics of a special case including only a single update. Furthermore, we formally explored the effects of parameterizing the denied boarding cost and stated the RMCU problem's relationship to the static overbooking problem.

In a detailed numerical study, we evaluated RMCU's performance compared to that of the industry standard solution BLIND and hindsight upper bound EX POST. Subsequently, we analyzed the model's sensitivity to several problem characteristics: scenario probabilities, demand level, the magnitude and quality of capacity updates, and the number of possible update times as well as update timing. In addition, we analyzed the effects of (desirable) changes that are in alignment with expected demand as opposed to (undesirable) changes that increase the discrepancy between expected demand and available capacity. Furthermore, we quantify run time reductions that result when solving the RMCU problem via a combinatorial algorithm as opposed to solving the corresponding mixed-integer formulation via CPLEX. Finally, we considered a numerical example that illustrates the relationship to existing approaches to static overbooking.

We conclude that RMCU's revenue potential increases with the difference between possible final capacities. The later the capacity is updated, the more revenue can be gained from using a stochastic model. As it takes into account potential denied boardings but no demand dependencies, the RMCU appears particularly useful when expecting capacity decreases. Note that when capacity increases, it can become optimal to offer additional, cheaper classes. This may go against the idea of monotonously increasing fares, such as commonly recommended to discourage strategic customer behavior. However, this effect can also be observed when updating demand forecasts given stochastic demand.

The analysis presented here focused on a leg-based model. However, we are aware that capacity uncertainty certainly entails network effects when customers book travel-itineraries rather than individual flights. Furthermore, most recent choice-based revenue management models consider demand to depend on the offered alternatives. For such dependent demand models, not anticipating capacity increases may further hurt revenue, as unexpected capacity increases cause cheap classes to become available shortly before departure. This enables both short-term cannibalization and long-term strategic customer behavior. Further research could exploit scenario-based approaches to effectively deal with overbooking challenges resulting from group bookings, when the number of expected no-shows may not grow incrementally.

Acknowledgments

We thank Lufthansa German Airlines for access to empirical data and industry expertise.

Appendix. Combinatorial Algorithm

Algorithm 1 Combinatorial Algorithm

1. Initialize parameters:

for all change times before departure, $t_i \in \{t_0, \dots, t_{N-1}\}$ **do**
    $demand[t_i][t_{i+1} - 1] \leftarrow$ list of revenues per expected customer in descending order in the time interval $[t_i, t_{i+1} - 1]$
    $p_{t_i} \leftarrow$ sum of probabilities of scenarios that may realize in $[t_i + 1, 0]$
$c_{\max} \leftarrow$ maximum capacity expected by any scenario

2. Calculate global revenue $GloRev\,[t, c]$ achievable from any change time $t \in \{t_0, \dots, t_n\}$ to the next for any feasible capacity c:

for all change times $t_i \in \{t_0, \dots, t_N\}$ **do**
    $GloRev\,[t_i, 0] \leftarrow 0$
    for capacity $c = 1$ to $c_{\max}$ **do**
        $GloRev\,[t_i, c] \leftarrow GloRev[t_i][c-1] + demand[t_i][t_{i+1} - 1][c-1]$;

3. Calculate scenario-based revenue $ScenRev\,[t, c]$ from any change time $t \in \{t_0, \dots, t_n\}$ until departure and for any feasible capacity c:

for all scenarios s **do**
    $c^s \leftarrow$ capacity expected by scenario$s$
    $p^s \leftarrow$ probability of scenario$s$
    Compute maximum revenue or denied boarding cost maxRev[s] for capacity $c^s - c_{\max}$
    $ScenRev\,[t^s, c_{\max}] \leftarrow p^s \cdot \text{maxRev}[s]$
    for capacity $c = c_{\max} - 1$ to 0 **do**
        $ScenRev\,[t^s, c] \leftarrow ScenRev\,[t, c+1] + p^s \cdot$ revenue from next most valuable customer if $c^s - c > 0$ and less denied boarding cost if $c^s - c \leq 0$

4. Calculate longest path as shown in Figure 2

for change times $t_i = t_1$ to t_N **do**
    for capacities $c = 0$ to $c_{\max}$ **do**
        $Revenue\,[t_i, c] = Revenue\,[t_{i-1}, c]$
        for $d = 0$ to c **do**
            $edgeRev = p_{t_i} \cdot GloRev[t_i][d] + ScenRev[t_i][c]$
            if then $Revenue[t_{i-1}, c-d] + edgeRev > Revenue[t_i, c]$
                $Revenue[t_i, c] = Revenue[t_{i-1}, c-d] + edgeRev$
Return $\max_{c \in \{1, \dots, c_{\max}\}} \text{Revenue}[t_N, c]$

References

Alstrup, J., S. Boas, O. B. G. Madsen, R. V. V. Vidal. 1986. Booking policy for flights with two types of passengers. *European Journal of Operational Research* **27**(3) 274–288.

Aydin, N., S. I. Birbil, J.B.G. Frenk, Noyan N. 2012. Single-Leg Airline Revenue Management with Overbooking. *Transportation Science* **47**(4) 560–583.

Barnhart, C., A. Farahat, M. Lohatepanont. 2009. Airline fleet assignment with enhanced revenue modeling. *Operations Research* **57**(1) 231–244.

Beckmann, M. J., F. Bobkoski. 1958. Airline demand: An analysis of some frequency distributions. *Naval Research Logistics Quarterly* **5**(1) 43–51.

Belobaba, P. 2006. Flight Overbooking: Models and Practice. Lecture notes. Massachusetts Institute of Technology, Boston, MA.

Belobaba, P., A. Farkas. 1999. Yield management impacts on airline spill estimation. *Transportation Science* **33**(2) 217–232.

Berge, M. E., C. A. Hopperstad. 1993. Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms. *Operations Research* **41**(1) 153–168.

Bish, E. K., R. Suwandechochai, D. R. Bish. 2004. Strategies for managing the flexible capacity in the airline industry. *Naval Research Logistics* **51**(5) 654–685.

De Boer, S. 2004. The impact of dynamic capacity management on airline seat inventory control. *Journal of Revenue and Pricing Management* **2**(4) 315–330.

Fischer, F., C. Helmberg. 2014. Dynamic graph generation for the shortest path problem in time expanded networks. *Mathematical Programming* **143**(1-2) 257–297.

Frank, M., M. Friedemann, M. Mederer, A. Schroeder. 2006. Airline revenue management: A simulation of dynamic capacity management. *Journal of Revenue and Pricing Management* **5**(1) 62–71.

Karaesmen, I., G. van Ryzin. 2004. Overbooking with substitutable inventory classes. *Operations Research* **52**(1) 84–104.

Liang, Y. 1999. Solution to the continuous time dynamic yield management model. *Transportation Science* **33**(1) 117–123.

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

Subramanian, J., S. Stidham, C. Lautenbacher. 1999. Airline yield management with overbooking, cancellations, and no-shows. *Transportation Science* **33**(2) 147–167.

Talluri, K.T., G. J. van Ryzin. 2004. *The theory and practice of revenue management*. Springer.

Topaloglu, H., S. I. Birbil, J. B. G. Frenk, N. Noyan. 2012. Tractable Open Loop Policies for Joint Overbooking and Capacity Control Over a Single Flight Leg with Multiple Fare Classes. *Transportation Science* **46**(4) 460–481.

Wang, X., A. Regan. 2002. Stochastic and dynamic airline yield management when aircraft assignments are subject to change. Tech. rep., Institute of Transportation Studies, University of California, Irvine, USA.

Wang, X., A. Regan. 2006. Dynamic yield management when aircraft assignments are subject to swap. *Transportation Research Part B: Methodological* **40**(7) 563–576.