On the Incorporation of Upgrades into Airline Network Revenue Management

May 9, 2014

Jochen Gönsch, Claudius Steinhardt

Jochen Gönsch (Corresponding Author)

Department of Analytics & Optimization,
University of Augsburg, Universitätsstraße 16, 86159 Augsburg, Germany

Phone: +49 821 598 4148

Fax: +49 821 598 4226

E-mail: jochen.goensch@wiwi.uni-augsburg.de

Claudius Steinhardt

Department of Quantitative Methods,
Bundeswehr University Munich (UniBw), Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany

E-mail: claudius.steinhardt@unibw.de

Abstract

Recently, the standard dynamic programming model of network revenue management has been extended for integrated upgrade decision-making. However, opposed to the original model, heuristically breaking the extended model down into a series of single-leg problems by dynamic programming decomposition in order to allow for real-world application is not possible. This is because the model's state space does not incorporate resources but commitments reflecting already sold products and capacity consumption is only resolved at the end of the booking horizon, thereby considering upgrade options.

In this paper, we consider arbitrary airline networks with upgrades being performed separately on each flight leg. We show that in this case, there are two reformulations of the extended model. First, we prove that an ad hoc formulation, in which upgrades are technically performed immediately after a sale, is completely equivalent. Second, we present another reformulation whose idea is adapted from linear programing-based production planning with alternative machine types. We prove that the resulting dynamic program is also equivalent. The advantage of both reformulations is that their state space is based on either real or virtual resources instead of the commitments used in the postponement formulation. Thus, dynamic programming decomposition techniques can again be applied. Despite the formal equivalence of both reformulations, applying decomposition techniques leads to different approximations and thus to potentially different results when applied in practice. Therefore, we finally numerically examine the approaches regarding revenue performance and discuss airline revenue management settings in which they differ.

Keywords: revenue management, airlines, upgrades, capacity control, dynamic programming
JEL Classification: L93, M11, C61

On the Incorporation of Upgrades into Airline Network Revenue Management

1 Introduction

In the airline industry, an *upgrade* is defined as an airline's offer to a passenger to fly in a higher class compartment than the one originally booked without any extra charge (see, e.g., McGill and van Ryzin 1999). Upgrades help to reduce temporary capacity and demand imbalance and to improve capacity utilization (see, e.g., Biyalogorsky et al. 2005). This becomes necessary because airlines face stochastic and seasonal demand while their capacity decisions concerning the operating aircraft and the number of seats in each compartment are fixed for the long term, leading to an occasionally inappropriate supply mix. If there is excess demand for one type of supply (e.g., economy seats) and excess supply of another type (e.g., business class seats), the airline can spontaneously upgrade selected economy passengers to business class and then sell the "vacated" economy seats to other customers, thus compensating the effect of the inadequate supply mix (see, e.g., Kimes and Thompson 2004). In general, upgrades are relevant for all capacity providers who offer several services that differ in their quality attributes, so these providers can replace a booked service with a more desirable substitute from a pre-specified set of alternatives (see, e.g., Gallego and Stefanescu 2012). Besides passenger airlines, examples include freight transport providers, hotels, car rental firms, cruise lines, theaters, and opera houses.

We focus on upgrading in the airline industry, where the upgrade decision is made individually for each leg of a passenger's network connection. Specifically, airline passengers do not mind changing, for example, from economy to business class or vice versa when they change planes at a stopover airport. This situation, which we call *legwise upgrading*, differs from many other capacity providers in the service industry, such as hotels or car rental companies. For example, car rental customers would not like to be upgraded on the first day of a two-day rental and then to have to return to change to a smaller car. Therefore, in such settings, only *productwise upgrading* is possible, in other words, the same type of resource must be provided throughout service fulfillment. Note that upgrading is different to upselling. In the latter, passengers are urged to voluntarily buy a higher value (and more expensive) product for potentially a lower price than originally quoted for this product.

An important issue for airlines in the context of upgrading is how to incorporate upgrade options in their revenue management systems. Airline revenue management's task is capacity control, which seeks to maximize profits by controlling the availability of the different booking classes, each with its fare, throughout the booking horizon. Modern revenue management systems performing capacity control are usually based on a number of mathematical optimization techniques and algorithms, which often implement heuristics such as dynamic programming decomposition. These decompositions are based on an exact but computationally intractable dynamic programming formulation that simultaneously considers the whole flight network. This exact formulation is well known in theory (see, e.g., Talluri and van Ryzin 2004, Chapter 3.2) and is widely accepted as the standard model formulation of network capacity control in general. However, its main disadvantage is that it only incorporates one resource type; that is, in the airline setting, one single compartment. The resulting nonconsideration of upgrades is inherited by the heuristics that build on it. To overcome this drawback, practical airline revenue management implementations usually resort to a fairly simple heuristic of successive planning, which means that upgrade contingents in higher compartments are determined first and that the resulting virtual capacity of each compartment is then considered fixed for standard capacity control, which is then performed separately for the different compartments, e.g., by applying decomposition heuristics. However, most airline revenue management system vendors have realized that a more versatile approach is needed – one that considers upgrade and capacity control decisions in an integrated way – and first suggestions have been made (see, e.g., Walczak 2010).

Besides the considerations in practice, the problem has also been tackled from a theoretical point of view (see, e.g., Alstrup et al. 1986; Karaesman and van Ryzin 2004; Shumsky and Zhang 2009; Wu et al. 2011; Gönsch et al. 2013). Gallego and Stefanescu (2009) were the first to extend the standard dynamic programming model of network revenue management for integrated upgrade decision-making. In line with the real-world setup, the upgrading decision is postponed to the end of the booking horizon, which guarantees full flexibility regarding resource utilization (*postponement upgrading*). The state space of the resulting model includes a so-called vector of commitments to track the sales of upgradeable products. Similar to the standard model of network revenue management, the model suffers from the curse of dimensionality and can thus not be computed straight away for real-world problem sizes. In order to make it applicable, the most prominent approach in theory as well as in practice would again be applying dynamic programming decomposition. However, this is not possible as the decomposition relies on a resource-based state space while that of the upgrade model is based on commitments. As a result, even though the dynamic programming model with integrated upgrade decision-making is of high practical interest, it has been unclear until now how to derive appropriate decomposition heuristics that would allow practical applications.

In this paper, we give answers to this open research question. In particular, we present two appropriate reformulations of the model that are valid for arbitrary airline networks and whose state spaces are defined on resources instead of commitments, such that dynamic programming decomposition techniques can again be applied. The first reformulation is based on a simplified model taken from the literature (see Gallego and Stefanescu 2009) which decides upgrades ad hoc at the time of sale (*ad hoc upgrading*). We prove that in the context of airline revenue management, ad hoc upgrading is indeed completely equivalent to postponement upgrading. We also show why the result does not hold for scenarios other than the airline industry in which only productwise upgrading is allowed. The idea of the second reformulation is adapted from linear programing-based production planning with alternative machine types (see Leachman and Carmon 1992). It incorporates upgrades by appropriately modifying the resource network and the consumption parameters to include artificial surrogate resources that are jointly used by multiple products (*surrogate upgrading*). It turns out that the formulation technically corresponds to a standard dynamic programming formulation without upgrades but modified model parameters, such that standard dynamic programming decomposition techniques can be applied. As each reformulation implies a different decomposition scheme, we perform a detailed numerical comparison of the resulting heuristics using typical airline revenue management scenarios. Specifically, we show that the heuristics' performance in terms of revenue achieved depends heavily on the demand structure.

The remainder of the paper is structured as follows: In Section 2, we define the airline revenue management setting, introduce the general notation and restate the basic dynamic programming formulation based on postponement upgrading. In Sections 3 and 4, we present the two reformulations discussed above and provide the analytical results. In Section 5, we apply dynamic programming decomposition to the two reformulations, and then present and discuss the results of our numerical study. In Section 6, we conclude with a summary of our key results.

2 Setting, notation, and basic model formulation

2.1 Setting and notation

We consider an airline network consisting of s flight legs $1,\ldots,s$. On each flight leg, a total of m different compartments $1,\ldots,m$ are offered, which are ordered according to some criteria of quality, with higher numbers indicating higher quality. For a specific flight leg $l \in \{1,\ldots,s\}$, $c_{rl} \geq 0$ denotes the number of seats in compartment $r \in \{1,\ldots,m\}$. $\mathbf{C} = [c_{rl}]$ is the

$m \times s$ matrix containing these capacity values for the whole network. As is common in revenue management, capacity is assumed to be fixed, and capacity that remains unused after the booking horizon is worthless. The airline offers n different products, that is, tickets. A product $k \in \{1,\dots,n\}$ includes transportation on a specific itinerary requiring the flight legs $\mathcal{L}_k \subseteq \{1,...,s\}$ and a seat in a specific compartment $r_k \in \{1,\dots,m\}$. The product price is denoted by p_k and fixed throughout the booking horizon. We assume that legwise upgrading to higher compartments $r \geq r_k$ is possible. Furthermore, let $\mathbf{A}^{(krl)}$ denote the $m \times s$ leg consumption matrix of product k when assigned to compartment r on flight leg l with $\mathbf{A}^{(krl)} = \left[a_{r'l'}^{(krl)}\right]$ and $a_{r'l'}^{(krl)} = 1$ for $r' = r$, $l' = l$ and $l \in \mathcal{L}_k$ and 0 otherwise. The booking horizon can be sufficiently discretized into T time periods, so that no more than one request arrives in each period $t \in \{1,\dots,T\}$. Each request asks for one product, and the independent demand assumption holds regarding demand. The probability of a request for product $k \in \{1,\dots,n\}$ in period $t \in \{1,\dots,T\}$ is given by $\lambda_k(t)$, and consequently, with probability $1-\sum_{k=1}^{n}\lambda_k(t)$, there is no request in a time period t. To ease notation, we omit writing the index sets for the symbols k referring to products, r and i referring to compartments, l referring to legs, and t to time periods. For example, the notation $\forall k$ means $k \in \{1,\dots,n\}$, and $\sum_k$ means $\sum_{k\in\{1,\dots,n\}}$. The symbol $\mathbf{0}$ refers to the zero vector, and $\mathbf{1}_s$ to the all-ones vector with s components. The operator $(\cdot)^+$ returns the maximum of zero and the value in brackets. Table 1 gives an overview of the notation used throughout Section 2. It already includes symbols introduced in the following Subsection 2.2.

Table 1: Notation introduced in Section 2

$l \in \{1,\dots,s\}$: flight legs	$\lambda_k(t)$: arrival probability of product k in period t
$r \in \{1,\dots,m\}$: compartments	$x_{rl} \leq c_{rl}$: remaining number of seats in compartment r on leg l
$c_{rl} \geq 0$: number of seats in compartment r on leg l	$\mathbf{X} = [x_{rl}]$: $m \times s$ matrix of resources' remaining capacity
$\mathbf{C} = [c_{rl}]$: $m \times s$ matrix of capacity values for the whole network	$y_{rl} \geq 0$: number of already accepted requests for products requiring at least compartment r on flight leg l (commitments)
$k \in \{1,\dots,n\}$: products	$\mathbf{Y} = [y_{rl}]$: $m \times s$ matrix of commitments
$\mathcal{L}_k \subseteq \{1,...,s\}$: flight legs required by product k	$V^{PP}(\mathbf{X},\mathbf{Y},t)$: optimal expected revenue-to-go from state $(\mathbf{X},\mathbf{Y},t)$ onwards (Postponement formulation)
$r_k \in \{1,\dots,m\}$: compartment required by product k	$\Delta_k V^{PP}(\mathbf{X},\mathbf{Y},t)$: opportunity cost of selling one unit of product k in state $(\mathbf{X},\mathbf{Y},t)$
p_k: price of product k	$\mathcal{A}$: set of feasible pairs, i.e. $(\mathbf{X},\mathbf{Y}) \in \mathcal{A}$ iif the commitments $\mathbf{Y}$ can be fulfilled with capacity $\mathbf{X}$
$a_{r'l'}^{(krl)}$: capacity consumption of product k in compartment r' on flight leg l' when assigned to compartment r on flight leg l	
$\mathbf{A}^{(krl)} = \left[a_{r'l'}^{(krl)}\right]$: $m \times s$ consumption matrix	
$t \in \{1,\dots,T\}$: time periods	

2.2 Basic dynamic programming formulation

We now restate the basic dynamic programming formulation with integrated postponement upgrading (see Gallego and Stefanescu 2009) in the context of airline revenue management. Each possible state within the capacity control process, that is, the process of accepting and rejecting incoming requests throughout the booking horizon, can be described by $(\mathbf{X},\mathbf{Y},t)$, where t denotes the considered time period, the $m\times s$ matrix $\mathbf{X}=[x_{rl}]$ denotes the resources' remaining capacity, with $\mathbf{X}\le\mathbf{C}$, and the $m\times s$ matrix $\mathbf{Y}=[y_{rl}]$ denotes the commitments, that is, $y_{rl}\ge 0$ is the number of already accepted requests for products requiring at least compartment r on flight leg l. Let $V^{PP}(\mathbf{X},\mathbf{Y},t)$ denote the expected revenue-to-go from state $(\mathbf{X},\mathbf{Y},t)$ onwards. The opportunity cost of selling one unit of product k is then given by $\Delta_k V^{PP}(\mathbf{X},\mathbf{Y},t)=V^{PP}(\mathbf{X},\mathbf{Y},t)-V^{PP}\left(\mathbf{X},\mathbf{Y}+\sum_l \mathbf{A}^{(kr_kl)},t\right)$, because accepting a request would simply lead to an additional commitment given by $\sum_l \mathbf{A}^{(kr_kl)}$. At the end of the booking horizon, all accepted upgradeable products $\mathbf{Y}$ must be provided with the remaining capacity $\mathbf{X}$. Let $\mathcal{A}$ denote the set of feasible pairs of $\mathbf{X}$ and $\mathbf{Y}$ (see, e.g., Gallego and Stefanescu 2009). $(\mathbf{X},\mathbf{Y})\in\mathcal{A}$ holds if and only if $\mathbf{X}\ge\mathbf{0}$ and there exists a feasible allocation of upgrades, that is, if the following feasibility problem has a solution:

$$\sum_{r'\ge r} z_{rr'l}=y_{rl} \qquad \text{for all } r \text{ and } l \tag{1}$$

$$\sum_{r'\le r} z_{r'rl}\le x_{rl} \qquad \text{for all } r \text{ and } l \tag{2}$$

$$z_{rr'l}\ge 0 \qquad \text{for all } r,\ r'\ge r \text{ and } l \tag{3}$$

with the assignment variables $z_{rr'l}$ denoting how many commitments for compartment r on flight leg l will be fulfilled by compartment r'.

The optimal expected revenue-to-go can then be computed recursively through the Bellman equation

$$V^{PP}(\mathbf{X},\mathbf{Y},t)=\sum_k \lambda_k(t)\left(p_k-\Delta_k V^{PP}(\mathbf{X},\mathbf{Y},t-1)\right)^+ + V^{PP}(\mathbf{X},\mathbf{Y},t-1) \tag{4}$$

with the boundary conditions $V^{PP}(\mathbf{X},\mathbf{Y},0)=0$ if $(\mathbf{X},\mathbf{Y})\in\mathcal{A}$ and $V^{PP}(\mathbf{X},\mathbf{Y},0)=-\infty$ otherwise.

The above opportunity cost-based formulation of the bellman equation has become standard in the revenue management literature as it allows for a straightforward interpretation. First, note that independently from incoming requests, the term $V^{PP}(\mathbf{X},\mathbf{Y},t-1)$ is always included in the revenue-to-go calculation (last term of (4)). It denotes the revenue-to-go obtainable if there is no request at all or if a current request was not accepted. Now, in case there is a request for a product k, which occurs with probability λ_k, the revenue p_k coming along with acceptance is opposed to its opportunity cost $\Delta_k V^{PP}(\mathbf{X},\mathbf{Y},t-1)$ (see first term of (4)). Intuitively, this directly reflects the fact that a request should only be accepted if its revenue exceeds its (opportunity) cost. Technically, in case the difference was negative, 0 would be returned by the operator $(\cdot)^+$, reflecting rejection and only $V^{PP}(\mathbf{X},\mathbf{Y},t-1)$ remains. In case of a positive difference, that is, acceptance, p_k as well as $\Delta_k V^{PP}(\mathbf{X},\mathbf{Y},t-1)$ would be included, with the latter reducing the (greater) revenue-to-go under rejection $V^{PP}(\mathbf{X},\mathbf{Y},t-1)$ accordingly. By simple algebraic manipulations and rearrangement of terms, the equivalent but more extensive formulation

$$V^{PP}(\mathbf{X},\mathbf{Y},t)=\sum_k \lambda_k(t)\max\left\{V^{PP}(\mathbf{X},\mathbf{Y},t-1),\, p_k+V^{PP}\left(\mathbf{X},\mathbf{Y}+\sum_l \mathbf{A}^{(kr_k l)},t-1\right)\right\}+\left(1-\sum_k \lambda_k(t)\right)V^{PP}(\mathbf{X},\mathbf{Y},t-1)$$

can be obtained.

Finally, note that resource consumption is only resolved at the end of the booking horizon by solving the feasibility problem (1)-(3). This inhibits the application of resource-based dynamic programming decomposition techniques to this model in order to allow for practical applications.

3 Reformulation 1: Ad hoc upgrading

3.1 Model formulation

We first consider a formulation originally proposed for a different setting in which upgrades have to be decided ad hoc at the time of sale (see Gallego and Stefanescu 2009, for a corresponding general formulation, and Steinhardt and Gönsch 2012, for a formulation with productwise upgrading). In this case, opposed to (4), there is a direct reduction of the remaining capacity immediately after acceptance; no commitments need to be stored throughout the booking horizon. Therefore, we can simply specify each possible state within the capacity control process by $(\mathbf{X},t)$. Let $V^{AH}(\mathbf{X},t)$ denote the expected revenue-to-go when the current state is $(\mathbf{X},t)$. Furthermore, $\mathbf{q}=(q_1,...,q_s)^T$ is a vector with s components corresponding to a possible legwise assignment with $0 \le q_l \le m$ for $l=1,...,s$. Then, $\Delta_{k\mathbf{q}}V^{AH}(\mathbf{X},t)=V^{AH}(\mathbf{X},t)-V^{AH}\left(\mathbf{X}-\sum_l \mathbf{A}^{(kq_l l)},t\right)$ is the opportunity cost of accepting a request for product k and fulfilling it with compartment assignment **q**. The expected revenue-to-go can then be calculated by the Bellman equation

$$V^{AH}(\mathbf{X},t)=\sum_k \lambda_k(t)\left(p_k-\min_{\mathbf{q}\ge \mathbf{1}_s\cdot r_k}\Delta_{k\mathbf{q}}V^{AH}(\mathbf{X},t-1)\right)^+ + V^{AH}(\mathbf{X},t-1) \tag{5}$$

with the boundary conditions $V^{AH}(\mathbf{X},0)=0$ for $\mathbf{X}\geq\mathbf{0}$, and $V^{AH}(\mathbf{X},0)=-\infty$ otherwise.

The additional notation introduced in Section 3 is summarized in Table 2.

Table 2: Additional notation introduced in Section 3

$V^{AH}(\mathbf{X},t)$: optimal expected revenue-to-go from state $(\mathbf{X},t)$ onwards (Ad hoc upgrading formulation)	$\mathbf{H}^{(s)}$: $m\times s$ matrix of remaining capacity after allocation $s\in\mathcal{S}_{\mathbf{XY}}$
$\mathbf{q}=(q_1,...,q_s)^T$: $s\times 1$ legwise assignment vector	$\mathcal{H}_{\mathbf{XY}}=\{\mathbf{H}^{(s)}\mid s\in\mathcal{S}_{\mathbf{XY}}\}$: set of all possible remaining capacity matrices.
$\Delta_{k\mathbf{q}}V^{AH}(\mathbf{X},t)$: opportunity cost of selling one unit of product k and assigning it to compartments $\mathbf{q}$ in state $(\mathbf{X},t)$	$\psi(\mathbf{X},\mathbf{Y})$: $m\times s$ matrix of free capacity that remains after the allocation of the given commitments $\mathbf{Y}$ on the given capacity $\mathbf{X}$
$\mathcal{S}_{\mathbf{XY}}$: set of all feasible allocations of commitments $\mathbf{Y}$ on capacity $\mathbf{X}$	

3.2 Monotonicity of opportunity cost

In the following, we show that in the context of airline revenue management, (5) is a reformulation of (4), that is, both are formally equivalent. Therefore, in this subsection, we first state the following monotonicity result that holds in respect of ad hoc upgrading as defined in model (5), generalizing the result shown in Steinhardt and Gönsch (2012) for the single-leg case:

Proposition 1. $\Delta_{k\mathbf{q}}V^{AH}(\mathbf{X},t)\geq\Delta_{k\mathbf{q}'}V^{AH}(\mathbf{X},t)$ *for all* $k,t,\mathbf{X}$ *with* $x_{q_l'l},x_{q_l l}>0\ \forall l\in\mathcal{L}_k$ *and* $\mathbf{q}\geq\mathbf{q}'$.

Proof. In the Online Appendix A.1, we prove this inequality by induction over t. While the base case is obvious, the inductive step is more complex and involves a second induction over the legs. In the latter, we first use the single-leg argumentation by Steinhardt and Gönsch (2012, Online Appendix A.1) to show the base case where the assignments differ in only one leg. We then prove that this also implies the original hypothesis for arbitrary assignments.

From Proposition 1 it follows that a specific product request's opportunity cost is always nondecreasing with regard to the assignment to higher compartments on any of the required legs. This result does not depend on the order of price, for instance, it is not a precondition for two products k and k' with $\mathcal{L}_k=\mathcal{L}_{k'}$ and $r_k>r_{k'}$ that $p_k>p_{k'}$, which means that Proposition 1 also holds in case that, for example, business class tickets are cheaper than corresponding economy class tickets. This can happen in practice owing to different booking class restrictions.

Proposition 1 has a straightforward implication for the ad hoc capacity control process. Owing to the opportunity cost monotonicity, it is no longer necessary to consider all the potential assignments for an incoming request to identify the minimum opportunity cost, but it suffices to determine the lowest available assignment $\mathbf{q}^* = min\left\{\mathbf{q} \geq \mathbf{1}_s \cdot r_k \mid x_{q_l l} > 0 \forall l\right\}$. Since, clearly, $min\left\{\mathbf{q} \geq \mathbf{1}_s \cdot r_k \mid x_{q_l l} > 0 \forall l\right\} = \left(min\left\{q_1 \geq r_k \mid x_{q_1 1} > 0\right\},\ldots,min\left\{q_s \geq r_k \mid x_{q_s s} > 0\right\}\right)$, only the smallest index available in the upgrade hierarchy to which the current request can be upgraded must be identified on each flight leg. If the opportunity cost of the resulting total assignment $\mathbf{q}^*$ is less than the revenue, the request is accepted and assigned according to this assignment, and if not, it can be rejected, since all larger indices leading to assignments $\mathbf{q} > \mathbf{q}^*$ will result in even higher opportunity costs, as implied by Proposition 1. Therefore, due to Proposition 1, the Bellman equation (5) can be reformulated as follows, making it easier to identify a specific request's assignment:

$$V^{AH}\left(\mathbf{X},t\right) = \sum_k \lambda_k\left(t\right)\left(p_k - \Delta_{k,\mathbf{q}^*} V^{AH}\left(\mathbf{X},t-1\right)\right)^+ + V^{AH}\left(\mathbf{X},t-1\right) \qquad (6)$$

with $\mathbf{q}^* = \left(q_1^*,\ldots,q_s^*\right)$ and $q_l^* = min\left\{q_l \geq r_k \mid x_{q_l l} > 0 \vee q_l = m\right\} \forall l \in \left\{1,\ldots,s\right\}$ and the boundary conditions $V^{AH}\left(\mathbf{X},0\right) = 0$ for $\mathbf{X} \geq \mathbf{0}$, and $V^{AH}\left(\mathbf{X},0\right) = -\infty$ otherwise.

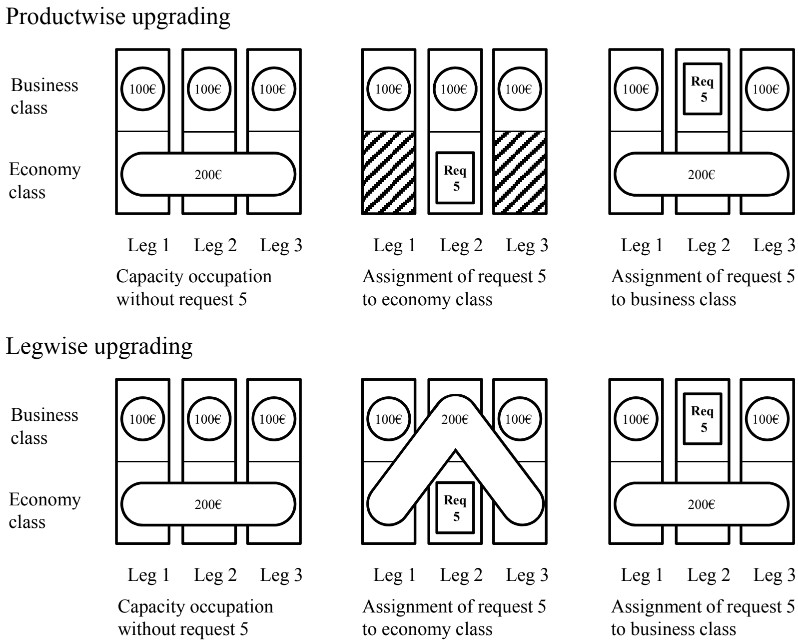
Specifically, the minimization term from (5) right before the opportunity cost is not needed, since the most valuable assignment $\mathbf{q}^*$ can be figured out without having to evaluate several future revenue-to-goes. Note that in the formulation, in case that there is no capacity left on one of the required flight legs $l \in \mathcal{L}_k$, the assignment is simply set to $q_l^* = m$. The opportunity cost calculation will lead to an infinite value in this case anyway, such that the request will not be accepted.

The monotonicity property stated by Proposition 1 is strongly related to the assumption that legwise upgrading is allowed (see Section 1). Specifically, the monotonicity of opportunity cost does not necessarily hold if only productwise upgrading is allowed, which technically implies that the assignment vector $\mathbf{q} = \left(q_1,\ldots,q_s\right)^T$ in (5) and (6) is restricted to $q_1 = q_2 = \ldots = q_s$. This can be shown by the following simple and intuitive counter-example:

Counterexample for product-wise upgrading. We consider 3 flight legs and 2 compartments with $c_{rl} = 1$ for $l \in \left\{1,2,3\right\}$ and $r \in \left\{1,2\right\}$. The booking horizon consists of $T = 5$ periods. With 100% probability, there will be a request for a ticket $k = 4$ – which is an economy class ticket (compartment 1) requiring all 3 legs – in period 4 ($\lambda_4\left(4\right) = 1$). In the following periods, there will be requests for the single-leg tickets $k = 1,2,3$ guaranteeing business class transportation (compartment 2) on flight leg 1, 2, and 3 respectively, with probability 100% each ($\lambda_3\left(3\right) = \lambda_2\left(2\right) = \lambda_1\left(1\right) = 1$). The related prices are $p_1 = p_2 = p_3 = 100$ € and $p_4 = 200$ €. In case there were no other incoming requests, the resulting optimal solution would be to accept all requests for products $k = 1,2,3,4$, leading to an overall revenue of 500€ (see Figure 1, left illustrations). We now consider that in period 5, there is a request for ticket $k=5$, which is a compartment 1 (economy class) single-leg ticket for leg 2. Now we calculate this request's opportunity cost depending on the assignment, given that only productwise upgrading is allowed (Figure 1, upper row):

- If the request is accepted and assigned to compartment 1, this will cause the opportunity cost $\Delta_{5(\cdot,1,\cdot)}V^{AH}(\mathbf{C},5)=200$€, because the future ticket 4 request requiring capacity on all three legs would no longer be accepted (see Figure 1, center illustration of upper row). In particular, upgrading the ticket 4 request would not make sense as this would lead to an opportunity cost of 300€ because all other future requests could no longer be accepted.
- If the current request for $k=5$ is assigned to compartment 2, this will only cause opportunity cost of $\Delta_{5(\cdot,2,\cdot)}V^{AH}(\mathbf{C},5)=100$€, owing to the fact that the future request for product 3 could no longer be accepted (see Figure 1, right illustration of upper row).

Figure 1: Productwise vs. legwise upgrading (Example 1)



Since $\Delta_{5(\cdot,1,\cdot)}V^{AH}(\mathbf{C},5)>\Delta_{5(\cdot,2,\cdot)}V^{AH}(\mathbf{C},5)$, monotonicity clearly does not hold in this example. By contrast, if legwise upgrading were allowed (Figure 1, lower row), assigning the ticket 5 request to compartment 1 would only cause opportunity cost of 100€, such that monotonicity would hold again. In particular, the future ticket 4 request could be upgraded exclusively on leg 2, such that only the request for product $k=2$ with revenue 100€ would not be acceptable in the future.

3.3 Formal equivalence of ad hoc upgrading and postponement upgrading in airline revenue management

Finally, to relate the values of models (4) and (5), similar to the single-leg case investigated by Steinhardt and Gönsch (2012), we use the following definition:

Definition 1. *Given a vector of capacity* $\mathbf{X}$ *and a vector of commitments* $\mathbf{Y}$ *with* $(\mathbf{X},\mathbf{Y}) \in \mathcal{A}$*, let* $\mathcal{S}_{\mathbf{XY}}$ *be the set containing all feasible allocations of commitments* $\mathbf{Y}$ *on capacity* $\mathbf{X}$*. For each* $s \in \mathcal{S}_{\mathbf{XY}}$*, let* $\mathbf{H}^{(s)}$ *be the* $m \times s$ *matrix of remaining capacity after allocation and let* $\mathcal{H}_{\mathbf{XY}} = \left\{ \mathbf{H}^{(s)} \mid s \in \mathcal{S}_{\mathbf{XY}} \right\}$ *be the set of all possible remaining capacity matrices. Then, we define the function* $\psi : \mathcal{A} \rightarrow \mathbb{N}_0^{m \times s}$ *as follows (*$\mathbb{N}_0^{m \times s}$ *is the set of all* $m \times s$ *matrices of nonnegative integer values):*

$$\psi(\mathbf{X},\mathbf{Y}) = \underset{\mathbf{H}^{(s)} \in \mathcal{H}_{\mathbf{XY}}}{arg\,max} \left\{ \sum_{r,l} r \cdot h_{rl}^{(s)} \right\}.$$

The intuition of $\psi(\mathbf{X},\mathbf{Y})$ is as follows: The function returns a matrix of free capacity that remains after the allocation of the given commitments $\mathbf{Y}$ on the given capacity $\mathbf{X}$. The matrix is chosen such that, on each flight leg, compartments with a larger number in the upgrade hierarchy are preferably kept free whenever possible. Within the maximization, this is technically achieved by weighting each compartment's remaining capacity with its index in the upgrade hierarchy on each flight leg. Algorithmically, a feasible allocation leading to $\psi(\mathbf{X},\mathbf{Y})$, given $(\mathbf{X},\mathbf{Y})$, can be obtained by successively moving up the compartment hierarchy from $r = 1,\ldots,m$ on each flight leg l allocating capacity for the commitments y_{rl} as low in the hierarchy as possible. Using Definition 1, the relationship between models (4) and (5) can be stated as follows:

Proposition 2. $V^{PP}(\mathbf{X},\mathbf{Y},t) = V^{AH}(\psi(\mathbf{X},\mathbf{Y}),t)$ *for all* $t,(\mathbf{X},\mathbf{Y}) \in \mathcal{A}$

Proof. The idea behind the proof is that if requests that would have led to commitments $\mathbf{Y}$ in the postponement variant have been accepted, the free capacity is $\psi(\mathbf{X},\mathbf{Y})$ – independent of the order in which the requests have arrived. First, we show that the postponement mechanism and the ad-hoc variant with free capacity $\psi(\mathbf{X},\mathbf{Y})$ have an identical set of feasible decisions (regarding both acceptance and upgrading). By induction over t, we then show the equivalence of the value functions as stated in Proposition 2. The complete proof is given in the Online Appendix A.2.

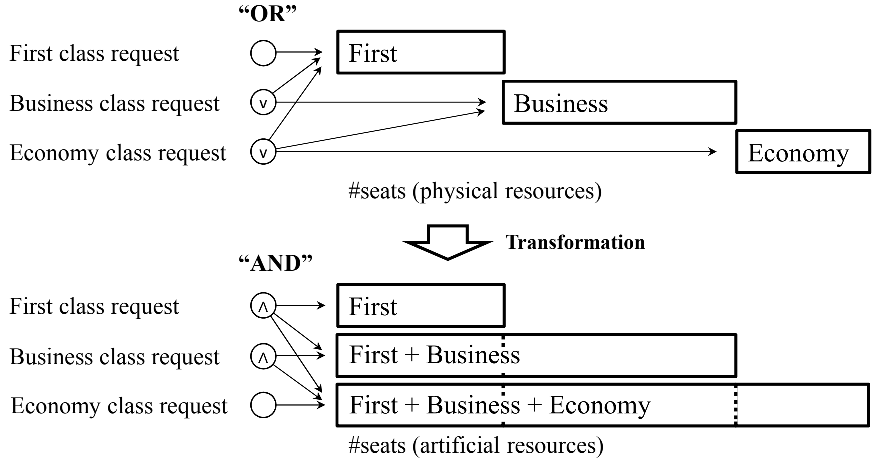
From Proposition 2, it follows that $V^{PP}(\mathbf{C},\mathbf{0},T) = V^{AH}(\mathbf{C},T)$ because $\mathbf{C} = \psi(\mathbf{C},\mathbf{0})$. Thus, Proposition 2 implies the equivalence of models (4) and (5) in the setting under consideration. This has the important implication that in airline revenue management, at least technically, upgrades can be decided at the time of sale without any loss in flexibility and in overall revenue. Note that this result generalizes results obtained for the single-leg case from our earlier work in Steinhardt and Gönsch (2012) to airline networks. The advantage of the ad hoc upgrading formulation is that due to the resource-based state space, adequate dynamic programming decomposition methods can straightforwardly be constructed (see Section 5.1.1).

4 Reformulation 2: Surrogate upgrading

4.1 Model formulation

The idea of the second reformulation is adapted from linear programing-based production planning with alternative machine types (see Leachman and Carmon 1992). On each leg, we define an artificial "surrogate" resource for each physical compartment. This artificial resource sums up the available capacity of the corresponding compartment and all higher compartments. Thus, the artificial resource comprises the maximum total capacity that could be used to fulfill requests concerning the compartment, including upgrade options. A specific product now simultaneously consumes capacity on each artificial resource equal to or lower than the requested compartment. Figure 2 illustrates this transformation by an example with three compartments on a single leg, comparing it with the ad hoc upgrading reformulation presented in Section 3. Within ad hoc upgrading, which is based on the original product and resource definitions, we consider the physical compartments' capacity separately and, for each incoming request, an OR decision must be made; that is, for example, a business class request can either be fulfilled with the business class compartment OR with a first class compartment (Figure 2, top). Within surrogate upgrading, which is based on transformed product and resource definitions, this disjunction is technically replaced by a conjunction; that is, for example, a business class request needs one unit from overall capacity (the surrogate economy class resource) AND one unit of capacity in the surrogate business class resource (lower part of Figure 2).

Figure 2: Modeling upgrades with artificial surrogate resources



To formalize the idea, let $\mathbf{X}'$ denote the $m \times s$ matrix of the artificial compartments' capacity. For a given matrix of remaining capacity $\mathbf{X}$, $\mathbf{X}'$ is obtained by applying the following transformation function $\varphi(\mathbf{X})$:

Definition 2. *Given a vector of remaining capacity* $\mathbf{X}$*, we define the function* $\varphi : \mathbb{N}_0^{m\times s} \to \mathbb{N}_0^{m\times s}$ *as follows: (*$\mathbb{N}_0^{m\times s}$ *is the set of all* $m \times s$ *matrices of nonnegative integer values):*

$$\varphi(\mathbf{X})_{rl} = \sum_{r' \geq r} x_{r'l} \ .$$

In the transformed problem, $\mathbf{C}' = \varphi(\mathbf{C})$ is the initial artificial compartments' capacity matrix. Furthermore, let $\mathbf{A}^{(k)}$ denote the $m \times s$ artificial consumption matrix of a product k with $\left[a_{rl}^{(k)}\right] = 1 \ \forall l \in \mathcal{L}_k \wedge r \leq r_k$ and $\left[a_{rl}^{(k)}\right] = 0$ otherwise. The expected revenue-to-go is denoted by $V^{Surr}(\mathbf{X}',t)$ with *t* remaining periods and remaining artificial capacity $\mathbf{X}'$. The opportunity cost of accepting a request for product k can then be defined as $\Delta_k V^{Surr}(\mathbf{X}',t) = V^{Surr}(\mathbf{X}',t) - V^{Surr}\left(\mathbf{X}' - \mathbf{A}^{(k)},t\right)$. The expected revenue-to-go can be calculated by the Bellman equation

$$V^{Surr}(\mathbf{X}',t) = \sum_k \lambda_k(t)\left(p_k - \Delta_k V^{Surr}(\mathbf{X}',t-1)\right)^+ + V^{Surr}(\mathbf{X}',t-1) \tag{7}$$

with the boundary conditions $V^{Surr}(\mathbf{X}',0) = 0$ for $\mathbf{X}' \geq \mathbf{0}$, and $V(\mathbf{X}',0) = -\infty$ otherwise.

Note that in surrogate upgrading, regarding the original meaning of the problem, compared to ad hoc upgrading, upgrade decisions are not made immediately after booking but, similar to postponement upgrading, at the end of the booking horizon. However, compared to postponement upgrading, there is no explicit storage of commitments and no separate feasibility problem must be solved, since feasibility is captured by the definition of the artificial resources and product resource consumption.

Analogously to Section 3, Table 3 contains an overview of the notation newly introduced in Section 4.

Table 3: Additional notation introduced in Section 4

$\mathbf{X}' = \varphi(\mathbf{X})$: $m \times s$ matrix of artificial compartments' capacity corresponding to remaining capacity $\mathbf{X}$	$V^{Surr}(\mathbf{X}', t)$: optimal expected revenue-to-go from state $(\mathbf{X}', t)$ onwards (Surrogate upgrading formulation)
$\mathbf{A}^{(k)}$: $m \times s$ artificial consumption matrix of a product k	$\Delta_k V^{Surr}(\mathbf{X}', t)$: opportunity cost of selling one unit of product k in state $(\mathbf{X}', t)$

4.2 Equivalence of surrogate upgrading and postponement upgrading in airline revenue management

Based on Definition 2, the following relationship between models (4) and (7) can be established:

Proposition 3. $V^{PP}(\mathbf{X}, \mathbf{Y}, t) = V^{Surr}(\varphi(\mathbf{X} - \mathbf{Y}), t)$ *for all* $t, (\mathbf{X}, \mathbf{Y}) \in \mathcal{A}$

Proof. The complete proof given in Online Appendix A.3 can be outlined as follows. Analogous to the proof of Proposition 2, it is first necessary to show that the same decisions are possible in the postponement and in the surrogate formulation. Here, this means that the same product requests can be accepted in both formulations. Using the definition of the feasibility problem $\mathcal{A}$ and some algebra, it is quite easy to show that all requests that can be accepted in the postponement formulation can also be accepted in the surrogate formulation. Showing that the inverse also holds is more cumbersome. To this end, we explicitly state a solution for the feasibility problem for every request that can be accepted in the surrogate formulation. To show the feasibility of this solution, we first consider constraints (2) and (3). Then, beginning with the highest compartment $r = m$, we show that (1) holds by induction over r. Finally, we use this to show the equivalence of the value functions as stated in Proposition 3 by induction over t.

From Proposition 3, it follows that $V^{PP}(\mathbf{C}, \mathbf{0}, T) = V^{Surr}(\varphi(\mathbf{C}), T)$ because $\varphi(\mathbf{C}) = \varphi(\mathbf{C} - \mathbf{0})$. Thus, Proposition 3 proves the equivalence of models (4) and (7) in the setting under consideration. Interestingly, given transformed remaining capacity $\mathbf{X}'$ and consumption matrices $\mathbf{A}^{(k)}$, the resulting Bellman equation for the capacity control problem given by (7) has the same form as the Bellman equation for traditional standard network revenue management without upgrades (see, e.g., Talluri and van Ryzin 2004, Chapter 3.2). Thus, by means of the surrogate approach, the revenue management problem incorporating upgrades can be transformed into an equivalent standard network revenue management problem without upgrades. Consequently, standard dynamic programming decomposition can straightforwardly be applied (see Section 5.1.2).

5 Application and numerical results

In this section, we apply dynamic programming decomposition techniques to the two model reformulations stated in the previous Sections 3 and 4. The idea of dynamic programming decomposition is to substitute the original multi-leg dynamic program with a number of single resource dynamic programs. Network effects are heuristically captured via adjusted revenues derived from a linear programming formulation (see, e.g., Talluri and van Ryzin 2004, Chapter 3.4.4). As we will demonstrate, even though both formulations were proven to be equivalent to the original postponement upgrading dynamic program presented in Section 2, depending on the formulation chosen as a starting point, different approximations are obtained when applying dynamic programming decomposition, with a corresponding potentially different revenue performance when used for capacity control.

In Section 5.1, we show how the dynamic programming decomposition technique is applied to the models presented in the previous sections. In Sections 5.2 to 5.4, we computationally compare the revenue performance of the resulting approaches, using simulations based on a network structure taken from the literature as illustration. Specifically, we present the setting under consideration in Section 5.2, we perform a detailed analysis providing an intuition of when and why differences in revenue performances may generally arise in Section 5.3, and we place them in context in a discussion in Section 5.4.

5.1 Application of dynamic programming decomposition

5.1.1. Ad hoc upgrading

The ad hoc upgrading dynamic program (5) can be decomposed according to a straightforward modification of the approach proposed by Steinhardt and Gönsch (2012) for product-wise upgrading in the context of car rental revenue management. The starting point is the deterministic linear programming (DLP) formulation corresponding to the ad hoc upgrading dynamic program (5) (**DLP-Upgrade**):

$$V^{DLP-AH}\left(\mathbf{X},t\right)=max\sum_{k}\sum_{\mathbf{q}\geq\mathbf{1}_s\cdot r_k}p_k z_{k\mathbf{q}} \tag{8}$$

subject to

$$\sum_{k}\sum_{\mathbf{q}\geq \mathbf{1}_s\cdot r_k} a_{rl}^{(kq_l l)} z_{k\mathbf{q}} \leq x_{rl} \qquad \text{for all } r \text{ and } l \tag{9}$$

$$\sum_{\mathbf{q}\geq \mathbf{1}_s\cdot r_k} z_{k\mathbf{q}} \leq D_{kt} \qquad \text{for all } k \tag{10}$$

$$z_{k\mathbf{q}} \geq 0 \qquad \text{for all } k \text{ and } \mathbf{q}\geq \mathbf{1}_s\cdot r_k \tag{11}$$

The decision variables in this model are $z_{k\mathbf{q}}$ for all products k and upgrade assignments $\mathbf{q}\geq \mathbf{1}_S \cdot r_k$, denoting with respect to each product k, the number of requests planned for acceptance and upgrading to $\mathbf{q}$. Similar to the traditional DLP without upgrades, constraints (9) and (10) reflect the limitations in capacity and demand respectively, with D_{kt} denoting the expected aggregated demand-to-come over the remaining periods $t,\ldots,1$. From solving DLP-Upgrade, we obtain the optimal values π_{rl} of the dual variables associated with the capacity constraints (9). If $x_{rl}=0$, we define $\pi_{rl}=\infty$.

Following Steinhardt and Gönsch (2012), a number of dynamic programs that each considers capacity only in one compartment on one flight leg are constructed. Specifically, for every compartment r' and leg l', we obtain a dynamic program that considers only the capacity $x_{r'l'}$ as follows:

$$V_{r'l'}^{AH}\left(x_{r'l'},t\right)=\sum_{k}\lambda_k\left(t\right)\left(p_k-\min_{\mathbf{q}\geq \mathbf{1}_s\cdot r_k}\left(\sum_{l\in\mathcal{L}_k}\left(\pi_{q_l l}-a_{r'l'}^{(kq_l l)}\pi_{r'l'}\right)+\Delta_{kq_{l'}}V_{r'l'}^{AH}\left(x_{r'l'},t-1\right)\right)\right)^{+} + V_{r'l'}^{AH}\left(x_{r'l'},t-1\right) \tag{12}$$

with the boundary conditions $V_{r'l'}^{AH}\left(x_{r'l'},0\right)=0$ for $x_{r'l'}\geq 0$, and $V_{r'l'}^{AH}\left(x_{r'l'},0\right)=-\infty$ otherwise. The opportunity cost is defined as

$$\Delta_{kq_{l'}}V_{r'l'}^{AH}\left(x_{r'l'},t\right)=V_{r'l'}^{AH}\left(x_{r'l'},t\right)-V_{r'l'}^{AH}\left(x_{r'l'}-a_{r'l'}^{(kq_{l'}l')},t\right). \tag{13}$$

Formula (12) can be seen as a variant of (5) that considers capacity only in compartment r' on leg l' ($x_{r'l'}$). Accordingly, opportunity cost $\Delta_{k\mathbf{q}}V^{AH}\left(\mathbf{X},t-1\right)$ must be calculated approximately without knowing capacity on other legs and compartments. If a product k with assignment $\mathbf{q}$ needs capacity on other legs and/or compartments, this is now captured by increasing the opportunity cost on the considered compartment r' on leg l' ($\Delta_{kq_{l'}}V_{r'l'}^{AH}\left(x_{r'l'},t-1\right)$) with an approximation of the opportunity cost on these other legs/compartments used. More specifically, the values π_{rl} of all legs and compartments required (except compartment r' on leg l') are summed up to calculate this approximation ($\sum_{l\in\mathcal{L}_k}\left(\pi_{q_l l}-a_{r'l'}^{(kq_l l)}\pi_{r'l'}\right)$). Thus, compared to (5), we now have $\Delta_{k\mathbf{q}}V^{AH}\left(\mathbf{X},t-1\right)$

$$\approx \sum_{l\in\mathcal{L}_k}\left(\pi_{q_l l} - a_{r'l'}^{(kq_l l)}\pi_{r'l'}\right) + \Delta_{kq_{l'}} V_{r'l'}^{AH}\left(x_{r'l'}, t-1\right).$$

Using (13), the total opportunity cost necessary to decide on the acceptance of requests is approximated as

$$\Delta_{k\mathbf{q}} V^{AH}\left(\mathbf{X},t\right) \approx \sum_{l\in\mathcal{L}_k} \Delta_{kq_l} V_{q_l\,l}^{AH}\left(x_{q_l\,l}, t\right) \tag{14}$$

and a request for product k is accepted if and only if its revenue p_k exceeds the approximation of opportunity cost, that is,

$$p_k \geq \sum_{l\in\mathcal{L}_k} \Delta_{kq_l} V_{q_l\,l}^{AH}\left(x_{q_l\,l}, t-1\right). \tag{15}$$

Analogously to the previous sections, Table 4 gives an overview of the new notation.

Table 4: Additional notation introduced in Section 5

$V^{DLP-AH}\left(\mathbf{X},t\right)$: objective value of the DLP ad hoc model in state $\left(\mathbf{X},t\right)$	$V^{DLP-Surr}\left(\mathbf{X}',t\right)$: objective value of the DLP ad hoc model in state $\left(\mathbf{X}',t\right)$
$z_{k\mathbf{q}}$: contingent for product k with upgrade assignments . $\mathbf{q}$	z_k: contingent for product k
D_{kt}: expected aggregated demand-to-come over the remaining periods $t,\ldots,1$	π_{rl}: optimal value of dual variable associated with leg l and compartment r
π_{rl}: optimal value of the dual variable associated with leg l and compartment r	$V_{r'l'}^{Surr}\left(x'_{r'l'},t\right)$: expected revenue-to-go from DP surrogate approximation considering capacity only on leg l' in compartment r'
$V_{r'l'}^{AH}\left(x_{r'l'},t\right)$: expected revenue-to-go from DP ad hoc approximation considering capacity only on leg l' in compartment r'	$\Delta_{kq_{l'}} V_{r'l'}^{Surr}\left(x'_{r'l'},t\right)$: opportunity cost from DP surrogate approximation considering capacity only on leg l' in compartment r'
$\Delta_{kq_{l'}} V_{r'l'}^{AH}\left(x_{r'l'},t\right)$: opportunity cost from DP ad hoc approximation considering capacity only on leg l' in compartment r'	$\Delta_{k\mathbf{q}} V^{Surr}\left(\mathbf{X}',t\right)$: opportunity cost of selling one unit of product k in state $\left(\mathbf{X},t\right)$ from DP surrogate approximation
$\Delta_{k\mathbf{q}} V^{AH}\left(\mathbf{X},t\right)$: opportunity cost of selling one unit of product k in state $\left(\mathbf{X},t\right)$ from DP ad hoc approximation	

5.1.2. Surrogate upgrading

As discussed in Section 4, the surrogate upgrading dynamic program (7) corresponds to a standard network revenue management dynamic program and the standard dynamic programming decomposition as described, e.g., in Talluri and van Ryzin 2004, Chapter 3.4.4, can be used. Again, the starting point for the decomposition is the deterministic linear programming (DLP) formulation. We call this formulation **DLP-Surrogate** to indicate the derivation from (7), although it is a standard network revenue management DLP formulation with remaining capacity $\mathbf{X}' = \varphi\left(\mathbf{X}\right)$ and consumption matrices $\mathbf{A}^{(k)}$ for all products:

$$V^{DLP-Surr}\left(\mathbf{X}',t\right)=max\sum_{k}p_k z_k \tag{16}$$

subject to

$$\sum_{k}a_{rl}^{(k)}z_k \le x'_{rl} \qquad \text{for all } r \text{ and } l \tag{17}$$

$$z_k \le D_{kt} \qquad \text{for all } k \tag{18}$$

$$z_k \ge 0 \qquad \text{for all } k \tag{19}$$

The decision variables in this model are z_k for all products k, denoting for each product k the number of requests planned for acceptance. Constraints (17) and (18) reflect the limitations in capacity and demand, respectively, with D_{kt} denoting the expected aggregated demand-to-come over the remaining periods $t,\dots,1$. From solving DLP-Surrogate, we obtain the optimal values π'_{rl} of the dual variables associated with the capacity constraints (17). If $x'_{rl}=0$, we define $\pi'_{rl}=\infty$.

Next, the standard approach involves constructing independent dynamic programs for each resource considering only this resource's capacity (see, e.g., Talluri and van Ryzin 2004 for a detailed explanation). Straightforwardly, this means constructing dynamic programs that each consider only one surrogate resource's capacity in our context. For the surrogate resource corresponding to compartment r' on leg l' (that is, summing up capacity in all compartments $r \ge r'$ on leg l'), we formulate a dynamic program that considers only the capacity $x'_{r'l'}$ as follows:

$$\begin{aligned}V_{r'l'}^{Surr}\left(x'_{r'l'},t\right)=\sum_{k}\lambda_k\left(t\right)&\left(p_k-\sum_{l,r}a_{rl}^{(k)}\pi'_{rl}+a_{r'l'}^{(k)}\pi_{r'l'}-\Delta_k V_{r'l'}^{Surr}\left(x'_{r'l'},t-1\right)\right)^{+}\\&+V_{r'l'}^{Surr}\left(x'_{r'l'},t-1\right)\end{aligned} \tag{20}$$

with the boundary conditions $V_{r'l'}^{Surr}\left(x'_{r'l'},0\right)=0$ for $x'_{r'l'}\ge 0$, and $V_{r'l'}^{Surr}\left(x'_{r'l'},0\right)=-\infty$ otherwise. In (20), the term $p_k-\sum_{l,r}a_{rl}^{(k)}\pi'_{rl}+a_{r'l'}^{(k)}\pi_{r'l'}$ is the adjusted revenue of product k for the surrogate resource corresponding to compartment r' on leg l', capturing the value of capacity consumed on other surrogate resources by the shadow prices π'_{rl}. The opportunity cost is defined as

$$\Delta_k V_{r'l'}^{Surr}\left(x'_{r'l'},t\right)=V_{r'l'}^{Surr}\left(x'_{r'l'},t\right)-V_{r'l'}^{Surr}\left(x_{r'l'}-a_{r'l'}^{(k)},t\right). \tag{21}$$

Using (21), the total opportunity cost necessary to decide on the acceptance of requests is approximated as

$$\Delta_k V^{Surr}\left(\mathbf{X}',t\right)\approx\sum_{r'\le r_k}\sum_{l\in\mathcal{L}_k}\Delta_k V_{r'l'}^{Surr}\left(x'_{q_l l},t\right) \tag{22}$$

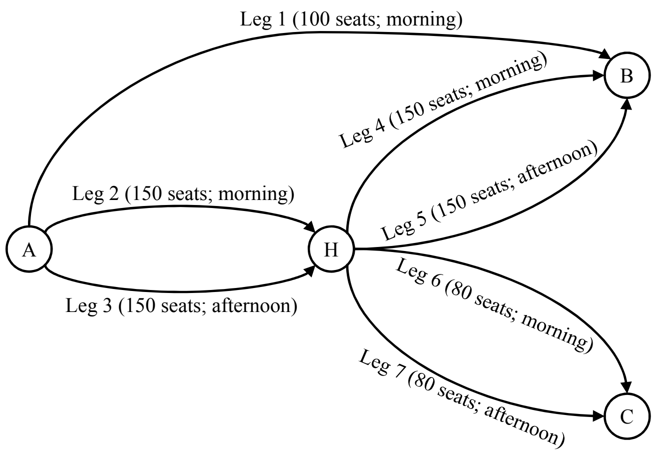
and, again, a request for product k is accepted if and only if its revenue p_k exceeds the approximation of opportunity cost, that is,

$$p_k \geq \sum_{r' \leq r_k} \sum_{l \in \mathcal{L}_k} \Delta_k V_{r'l'}^{Surr} \left(x'_{q_l l}, t-1 \right). \tag{23}$$

5.2 Simulation experiment design

The numerical experiments we conduct are based on a network structure taken from the literature that was originally proposed by Liu and van Ryzin (2008, Section 7.2) and that has been used as basis for a number of follow-up studies (see, e.g., Miranda Bront et al. 2009; Steinhardt and Gönsch, 2012).

Figure 3: Airline network



There are four cities A, B, C, and H that are connected by seven flight legs with capacities of between 80 and 150 seats (see Figure 3). On each flight leg, three compartments are available. 60% of the seats are economy class, 30% are business class, and 10% are first class. In addition to the seven single-leg flights, customers can also travel on four connecting itineraries: B can be reached from A via the hub H in the morning (legs 2 & 4) and in the afternoon (legs 3 & 5). Likewise, C can be reached from A via the hub H in the morning (legs 2 & 6) and in the afternoon (legs 3 & 7). Leg 1 is a direct flight from A to B which we refer to as "direct" in what follows. Each of the compartments can be booked in an expensive (full fare) and a discounted booking class. The prices depend on the itinerary type and are provided in Table 5. Thus, in our example, a total of 66 products are available.

Table 5: Booking classes with prices

	A (first)	F (first)	C (business)	P (business)	Y (eco)	Q (eco)
single leg	800	600	500	300	400	200
connect	1,400	1,050	875	525	700	350
direct (leg 1)	2,000	1,500	1,250	750	1,000	500

In case that total expected demand equals the capacity on all flights, expected demand is calculated as follows: demand for each connecting itinerary equals 30% of its total capacity and demand for the single leg flights is set equal to the remaining capacity.

To model different situations regarding demand for the compartments, we consider two realistic demand patterns. In the pattern *Balanced*, demand is balanced in the sense that expected demand is proportional to the compartment sizes, that is, on each itinerary, 60% of demand is for economy class, 30% is for business class, and 10% is for first class. By contrast, in *StrongEco* we have a very strong demand of 80% for economy and only 15% for business class and 5% for first class. In both demand patterns, 70% of each compartment's customers request the cheaper booking class (that is, F, P, or Q) and the remaining 30% buy the expensive one (A, C, or Y).

In line with our model assumptions from Section 2.1, the booking process is discretized into T periods using a standard procedure (see, e.g., Subramanian et al. 1999, Section 3.2.1). We assume the booking process to be time-homogeneous and the time-independent arrival rate is calculated accordingly from the expected demand value: $\lambda_k(t) = D_{kT}/T \ \forall t$. Furthermore, the total number of time periods is calculated so that there is at most one incoming customer request in each period: $\sum_k \lambda_k(t) < 1$.

We fix the number of simulation runs, each representing a stream of product requests, for all considered scenarios to 40, and, depending on the matter of interest, report resulting aggregated performance indicators along with the corresponding confidence levels. If different control methods are compared, we use the same set of 40 streams of product requests for all of them. Furthermore, we generate additional scenarios by varying the demand intensity in advance in order to simulate different load factors. Therefore, before splitting demand into single leg and connecting itinerary demand as described above, we scale the expected demand using a parameter $\alpha \in \{0.9, 1.0, ..., 1.5\}$, where $\alpha = 1$ corresponds to the case that expected demand equals capacity on all flights. Thus, by combining the six load factors with the three demand patterns, we obtain 18 scenarios.

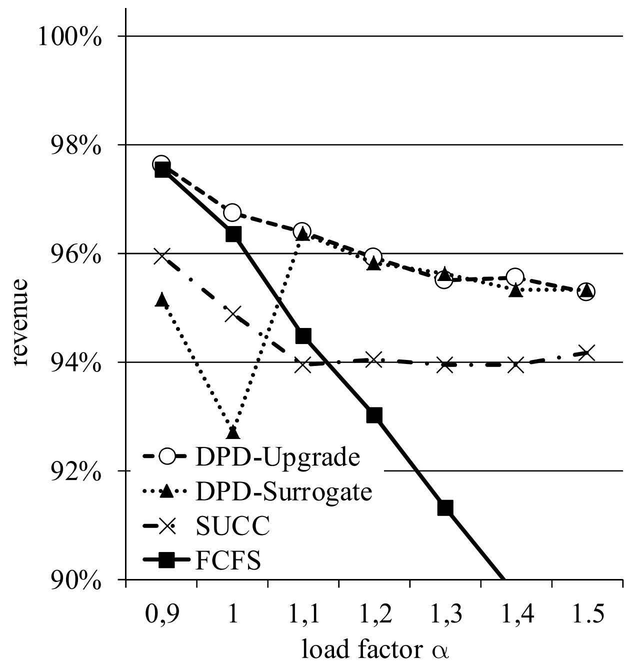
The following alternative control mechanisms were implemented to decide on the acceptance of product requests in each scenario:

- *DPD-Upgrade* is the dynamic programming decomposition described in Section 5.1.1 that directly works with the problem formulation with upgrades and uses (15) to decide on the acceptance of requests based on approximated opportunity cost.
- *DPD-Surrogate* is the dynamic programming decomposition (Section 5.1.2) where the revenue management problem with upgrades is first transformed to a network problem without upgrades before a standard dynamic programming decomposition is applied. Here, (23) is used for the acceptance decisions.
- *SUCC* is based on successive planning and mimics an upgrade control that is currently widely used in commercial revenue management software systems. In the first step, virtual capacities are determined for each of the three compartments by estimating in advance the number of upgrades that must be performed. For example, if 10 upgrades from economy class to business class are required on a certain leg, the capacity of business class is reduced by 10, and that of economy class is increased by the same number. To realize this step, we use the primal solution of model (8)-(11) and adjust the different compartments' capacities according to the optimal values of the decision variables. In the second step, the resulting virtual capacity vector is fed into a traditional control method without upgrade capabilities, which then performs capacity control over time. In our study, we use a traditional dynamic programming decomposition (the same approach as (16)-(23), but applied to the original problem and using virtual capacities instead of modeling upgrades directly or via surrogate resources) to determine opportunity cost and decide on the acceptance of requests. Both steps are iterated three times during the control process, that is, the virtual capacities are twice recalculated according to the current demand information and capacity load.
- *FCFS* implements a very simple first-come-first-served control mechanism, which is widely used in the literature to judge the performance of control approaches. Requests are accepted as long as they can be fulfilled by the remaining capacity. Upgrades are undertaken if necessary, moving up the upgrade hierarchy successively.
- *ExPost* calculates the perfect hindsight optimal revenue that can be obtained if full information on the incoming demand is used. For each simulation run, a model of type (8)-(11) is solved, optimally allocating capacity to the current demand stream's requests, which are used as the RHS of constraints (10) instead of the forecasted demand. The obtained revenue is an upper bound for all the other methods' output.

5.3 Performance results

Figure 4 shows the revenues obtained by the control mechanisms for the *Balanced* demand pattern relative to *ExPost*'s perfect hindsight revenue. Regarding load factors $\alpha \geq 1.1$, the results show what is intuitively expected: *DPD-Surrogate*'s and *DPD-Upgrade*'s revenues differ only slightly, with no clear advantage for any mechanism. Both significantly outperform *SUCC*, which in turn significantly outperforms *FCFS* in all scenarios with $\alpha \geq 1.2$. Please note that here and in the remainder of this paper, we always refer to the 99% confidence level when using the term significant (see Table A.1 in Appendix A.4 for a detailed comparison with confidence intervals). Furthermore, by and large, all mechanisms' revenues decline in α as capacity control is more difficult – and important – for scarcer capacities. However, for $\alpha \leq 1.0$, *DPD-Surrogate* clearly yields the lowest revenue of all mechanisms. A closer investigation of the demand streams shows that this is because, compared to the other approaches, *DPD-Surrogate* initially rejects more requests for the cheap first (F) and business (P) booking classes. For $\alpha = 0.9$, for example, this leads to a load of only 80.5% compared to 83% for *DPD-Upgrade*, even though capacity is not scarce in expectation.

Figure 4: Performance of the control methods relative to *ExPost* (*Balanced* demand)



DPD-Surrogate's behavior can be explained as follows: The weak expected demand is reflected in low shadow prices obtained from the *DLP-Surrogate*, which are even 0 in this case because capacity is not scarce. Thus, according to equation (20), the adjusted revenues used in the dynamic programs for the surrogate resources are equal to the products' prices. This can have the following two effects if expected demand is smaller than capacity, but not too weak:

- First, on a surrogate resource that aggregates multiple compartments' capacity, the opportunity cost can exceed the prices of the lowest compartment's products even if no upgrade would be necessary to accept the request, because, since the dynamic program does not consider compartment restrictions, it also considers the possibility to accept more requests than capacity in higher compartments. When the control mechanism actually accepts requests, downgrades are not performed and capacity might remain unused.
- Second, regarding products belonging to higher compartments, the effect of high opportunity cost is even more severe. Since every dynamic program uses the full price, opportunity cost will be close to the cheapest product's price if expected demand almost equals capacity, because the probability of selling the last unit of capacity is fairly high. However, in this case, from condition (23) it follows that a product from a higher compartment, requiring capacity on several surrogate resources on each leg used would require a revenue exceeding the sum of these resources' opportunity cost to be sold. However, this is often not the case for classes F and P, because the true opportunity costs are clearly overestimated, leading to the rejection of such requests.

Note that the described effect is relatively more severe for $\alpha = 1.0$, because the shadow prices obtained from the *DLP-Surrogate* are still 0, while demand is even stronger than in the case of $\alpha = 0.9$, which explains the kink in the graph.

Overall, *DPD-Surrogate*'s behavior is clearly suboptimal and leads to counter-intuitive control decisions, but technically follows from the general construction of the standard dynamic programming decomposition approach. However, in the *Balanced* demand pattern this is only the case for very low load factors which are unlikely to exist in revenue management settings in reality. *DPD-Upgrade* does not suffer from comparable drawbacks in this case. While low shadow prices reflecting abundant capacity are also obtained from *DLP-Upgrade*, this – however – correctly signals lower compartments' dynamic programs that there is sufficient capacity for upgrading, decreasing opportunity cost and leading to the acceptance of requests, as can be seen from the definition of the lower compartments' dynamic program given by (12).

Note that independently of specific values of the given load factors, similar drawbacks of *DPD-Surrogate* arising from the DLP's shadow prices also occur in unbalanced patterns with comparatively weak demand for economy class. However, such patterns are also quite unrealistic in practice, as they could conceivably only arise through gross mispricing of the different types of products, and thus are unlikely to persist.

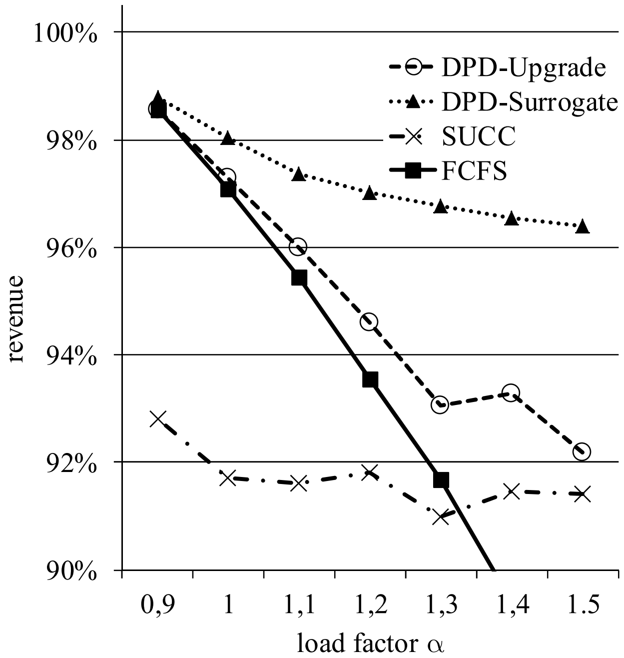
Next, we consider the *StrongEco* demand pattern; the results are displayed in Figure 5. Again, the dynamic programming decomposition approaches significantly outperform the *SUCC* mechanism. However, regarding *DPD-Surrogate* and *DPD-Upgrade*, the picture is different to the *Balanced* demand pattern. Both mechanisms still yield similar revenues for $\alpha = 0.9$ where capacity control is not really necessary, and even *FCFS* obtains a similar revenue, but *DPD-Upgrade*'s revenue quickly declines in the load factor to approximately 92%, whereas *DPD-Surrogate* manages to stay more constant and yields approximately 96.5% of *ExPost* for $\alpha = 1.5$.

A closer look at this load factor shows that *DPD-Upgrade* accepts more requests (71%) and performs more upgrades (18%) than *DPD-Surrogate*, at 68% and 13% respectively. This is because the strong economy class demand comes along with a weak demand for business class and first class, leading to low shadow prices for these compartments, which are mostly even 0 here. Thus, in *DPD-Upgrade*, upgrading is considered cheap by the economy class dynamic programs, leading to the impression of an unlimited capacity, low opportunity cost (which can never exceed the price of upgrading), and the acceptance of all requests for economy class products. If, in a demand stream, economy class capacity becomes scarce, it might be better to upgrade a Y request than to reject it; but this only partially counteracts the acceptance of too many low value requests, since this upgrade might displace more expensive requests for higher compartments that arrive later.

By contrast, *DPD-Surrogate* contains a dynamic program modeling the capacity of a surrogate resource reflecting total capacity for every leg. These dynamic programs perfectly capture the tradeoff between requests for lower compartments and requests for higher ones, as long as compartment restrictions are not binding, which would be the case in patterns with weak economy demand. Thus, *DPD-Surrogate* does not suffer from the acceptance of too many economy products in the situation described above.

Note that all mentioned effects are relevant in the sense that they are not restricted to the corresponding pathologic example used to illustrate them, but generally occur if the mentioned conditions regarding the demand situation prevail.

Figure 5: Performance of the control methods relative to *ExPost* (*StrongEco* demand)



5.4 Discussion

DPD-Upgrade and *DPD-Surrogate* both decompose the computationally intractable multi-class network revenue management problem, using a linear program to obtain easy-to-calculate dynamic programs considering only a single resource's capacity. However, comparing them leads to ambiguous results. What can be stated with certainty is that there are situations where these mechanisms' revenues differ significantly. But for each mechanism, there are situations of superiority and of near-equivalence. These fluctuations in revenue are unsurprising, since both decompositions are based on the DLP's dual solution, which often strongly depends on the exact demand situation and the resulting capacity scarcity. Nonetheless, their behavior is rooted in the very different approaches to capture the possibility of upgrading. The single-resource dynamic programs (12) used in *DPD-Upgrade* integrate upgrading via the possibility of assigning a request to a resource with a capacity that is not modeled and is thus technically considered unrestricted, but is somehow captured via the corresponding dual price that decreases the product's revenue in the case of upgrading. This approach suffers if the dual values do not adequately reflect the value of higher compartments' capacity. By contrast, *DPD-Surrogate* accounts for all capacity restrictions. However, the integration of upgrading via the construction of artificial surrogate resources increases the complexity of the network structure, because even single-leg products for higher compartments now need capacity on multiple surrogate resources. This more complex network is more vulnerable to fluctuations in the dual prices used to decompose it into the single-resource dynamic programs (20). However, it turns out that these vulnerabilities of DPD-Surrogate are only prevalent in situations that are unlikely to persist in revenue management practice, that is, when capacity is not scarce at all or when economy demand is comparatively weak.

A standard way to mitigate the sensitivity of DLP's dual values regarding demand and capacity situations is to resolve the model during the booking horizon. This usually improves revenues at the cost of an increased runtime resulting from the recalculation of all the dynamic programs after obtaining the new dual values. We also tried this and found that revenues mostly improve slightly, but sometimes also decline. The extreme outliers where a mechanism especially suffers from 'bad' dual values become less severe. For example, the biggest difference between *DPD-Upgrade* and *DPD-Surrogate* in the *Balanced* demand pattern at $\alpha = 1.0$ declines from almost exactly 4% to 3%. Since this effect of resolving is standard and does not change the overall message, we do not report the results here.

SUCC mimics the traditional way of incorporating upgrade decisions in airline revenue management systems. It also 'decomposes' the original problem in the sense that the DLP's primal solution is used to explicitly reserve seats in advance for upgrades based on the expected demand. Compared to the other approaches, *SUCC* provides very stable but mostly inferior revenues. This mirrors a well-known observation from standard revenue management, where capacity control policies based on a primal solution are usually less sensitive to changes in the demand forecast, but provide considerably lower revenues without sophisticated nesting structures allowing the acceptance of requests exceeding the expected values.

6 Conclusions

In this paper, we present two reformulations of the recently proposed dynamic programming model of airline network revenue management with integrated upgrade decision-making, namely the ad hoc upgrading formulation and the surrogate upgrading formulation. We derive a number of structural properties and insights, such as opportunity cost monotonicity of the ad hoc formulation which allows us to further simplify the procedure in the airline context, and a counter example why monotonicity does not hold in general for arbitrary industry settings with upgrades. Furthermore, the surrogate upgrading formulation is formally equivalent to a standard dynamic programming formulation without upgrades which many revenue management systems are based on, with only the model parameters having to be adapted. The most important aspect, however, is that we are able to show that both formulations are formally equivalent to the original model and thus really are lossless reformulations of it in the airline context. Even more, opposed to the original model, both reformulations are suited for applying dynamic programming decomposition, which is the most common technique to handle real-world problem sizes in theory as well as in practical applications. This is an important result, as until now, it has not yet been clear how to make the dynamic programming model with integrated upgrade decision-making applicable in practice. However, depending on the chosen reformulation, the result of applying dynamic programming decomposition differs, such that one could also expect different results in terms of attainable profit when the approaches are applied in practice. To investigate potential differences, we perform a number of computational experiments based on an extract from an airline network that has been adapted from the literature. Depending on the specific demand situation, we observe large revenue differences in some cases. This is because the approaches are strongly dependent on the quality of the used shadow prices from the corresponding DLP. For example, low shadow prices in the decomposition of the ad hoc upgrading model formulation lead to cheap upgrade opportunities, which might in turn result in too much low value demand being accepted. Additionally, many other effects are identified and discussed.

Based our results, we derive the following managerial implications:

- Similar to standard capacity control without upgrades, both proposed decomposition approaches are applicable to realistic problem sizes owing to their one-dimensionality regarding capacity.
- Our results indicate that both decomposition approaches significantly outperform successive planning as the de facto industry standard, given real world revenue management assumptions like scarce capacity.
- Under conditions that are usually given in real world revenue management practice, the surrogate upgrading decomposition performs at least as good as or better than the ad hoc upgrading decomposition.
- The surrogate approach allows successfully addressing upgrade settings with standard methods for revenue management without upgrades, simply by adequately defining artificial resources and changing the different tickets' consumption values, without having to modify the logic of the existing revenue management systems. However, the modification of the system's input comes at the cost of the results no longer being readily interpretable. This needs to be taken into consideration when implementing the approach in practice.
- In case of very weak overall demand compared to capacity as well as in case of a comparatively weak demand for economy class, the ad hoc upgrading decomposition

outperforms the surrogate upgrading decomposition. Even though those conditions are unlikely to persist in practice, this shows that the choice of the approach should strongly depend on which typical situations regarding for instance demand intensity are regularly experienced by the specific airline and what the prevalent upgrade ratio is.

- The ad hoc upgrading approach is quite intuitive, because the dynamic programs directly incorporate the upgrade decision in terms of a disjunction and the results of applying the decomposition approach are readily interpretable. However, it must be considered that the logic of the existing revenue management systems needs to be changed slightly, since the disjunction must be integrated into the existing dynamic programming routines.

Overall, both approaches proposed in this paper can be a meaningful alternative to the existing traditional successive planning approaches. Therefore, we strongly encourage airline practitioners to implement and test the proposed approaches and explicitly investigate the revenue advantage over successive planning, as well as which of the decomposition approaches turns out to be more suited to their specific setting.

References

Alstrup J, Boas S, Madsen OBG, Vidal RVV (1986) Booking policy for flights with two types of passengers. Eur J Oper Res 27:274–288

Biyalogorsky E, Weiss D, Xie J (2005) The economics of service upgrades. J Serv Res 7:234–244

Gallego G, Stefanescu C (2009) Upgrades, upsells and pricing in revenue management. Working paper. IEOR Department, Columbia University

Gallego G, Stefanescu C (2012) Services engineering: Design and pricing of service features. In Özer Ö, Phillips R (eds.), The Oxford handbook of pricing management. Oxford University Press, Oxford, pp 713–737

Gönsch J, Koch S, Steinhardt C (2013) An EMSR-based approach for revenue management with integrated upgrade decisions. Comp & Oper Res 40:2532–2542

Karaesmen I, van Ryzin GJ (2004) Overbooking with substitutable inventory classes. Oper Res 52:83–104

Kimes SE, Thompson GM (2004) Restaurant revenue management at Chevys: Determining the best table mix. Decis Sci 35:371–392

Leachman RC, Carmon TF (1992) On capacity modeling for production planning with alternative machine types. IIE Trans 24:62–72

Liu Q, van Ryzin GJ (2008) On the choice-based linear programming model for network revenue management. Manuf Serv Oper Manag 10:288–310

Miranda Bront JJ, Méndez-Díaz I, Vulcano G (2009) A column generation algorithm for choice-based network revenue management. Oper Res 57:769–784

McGill JI, van Ryzin GJ (1999) Revenue management: Research overview and prospects. Transp Sci 33:233–256

Talluri KT, van Ryzin GJ (2004) The theory and practice of revenue management. Kluwer, Boston

Shumsky RA, Zhang F (2009) Dynamic capacity management with substitution. Oper Res 57:671–684

Steinhardt C, Gönsch J (2012) Integrated revenue management approaches for capacity control with planned upgrades. Eur J Oper Res 223:380–391

Subramanian J, Stidham S, Lautenbacher CJ (1999) Airline yield management with overbooking, cancellations, and no-shows. Transp Sci 33:147–167

Walczak D (2010) Capacity sharing. Presentation at the 2010 AGIFORS airline revenue management conference, New York

Wu HH, Wu CH, Lin JT (2011) Dynamic selling of quality-graded products under demand uncertainties. Comp Ind Eng 61:141–149

Online Appendix

On the Incorporation of Upgrades into Airline Network Revenue Management

May 09, 2014

In the following Appendices A.1 to A.3, we present the proofs of the Propositions 1, 2, and 3 stated in Sections 3 and 4 of the paper. Unless otherwise mentioned, we use the mathematical notation as defined in Sections 2 to 4. In Appendix A.4, we present an additional comparison of the revenues obtained with the different mechanisms in Section 5.

A.1. Proof of proposition 1

Proposition 1. $\Delta_{k\mathbf{q}}V^{AH}(\mathbf{X},t) \geq \Delta_{k\mathbf{q}'}V^{AH}(\mathbf{X},t)$ *for all* $k, t, \mathbf{X}$ *with* $x_{q_l'l}, x_{q_l l} > 0\ \forall l \in \mathcal{L}_k$ *and* $\mathbf{q} \geq \mathbf{q}'$.

Proof. The proof uses and generalizes the single-leg proof stated in Steinhardt and Gönsch (2012). To simplify the derivation without loss of generality, we assume concerning the value function (5) that if a request is accepted, the controller chooses the upgrade assignment with the smallest index on each flight leg if $\underset{\mathbf{q}\geq \mathbf{1}_s \cdot r_k}{arg\,min}\, \Delta_{k\mathbf{q}}V^{AH}(\mathbf{X},t-1)$ is not unique; that is, if there are several possible assignments with the same minimal opportunity cost. Furthermore, we reformulate the value function (5) by means of an $n \times s$ policy matrix $\mathbf{U}$ with $u_{kl} \in \{0,1,\ldots,m\}\ \forall k,l$ determined for each stage with the following meaning: if a request for product k arrives in the current time period, it will be denied if $\exists l$ with $u_{kl} = 0$, and otherwise accepted and upgraded to resource type u_{kl} on flight leg l. The set $\mathcal{U}(\mathbf{X})$ contains all policies $\mathbf{U}$ that are feasible with the remaining capacity $\mathbf{X}$ and is formally defined by

$$\mathcal{U}(\mathbf{X}) = \left\{ \mathbf{U} \mid u_{kl} \in \{0,1,\ldots,m\}\ \forall k,l \wedge \left((u_{kl} = 0 \forall l) \vee \left(\sum_l \mathbf{A}^{(ku_{kl}l)} \leq \mathbf{X} \wedge u_{kl} \geq r_k \forall l \right) \forall k \right) \right\}. \tag{A.1}$$

Then (5) can be rewritten as

$$V^{AH}(\mathbf{X},t) = \max_{\mathbf{U}\in\mathcal{U}(\mathbf{X})} \left\{ \sum_{k|u_{kl}\neq 0 \forall l} \lambda_k(t) \cdot \left\{ p_k + V^{AH}\left(\mathbf{X} - \sum_l \mathbf{A}^{(ku_{kl}l)}, t-1 \right) \right\} + \left(1 - \sum_{k|u_{kl}\neq 0 \forall l} \lambda_k(t) \right) \cdot V^{AH}(\mathbf{X},t-1) \right\} \tag{A.2}$$

with the boundary condition $V^{AH}(\mathbf{X},0) = 0$.

From the definition of the opportunity cost, showing $\Delta_{k\mathbf{q}}V^{AH}(\mathbf{X},t) \geq \Delta_{k\mathbf{q}'}V^{AH}(\mathbf{X},t)$ is equivalent to showing $V^{AH}\left(\mathbf{X} - \sum_l \mathbf{A}^{(kq_l l)}, t\right) \leq V^{AH}\left(\mathbf{X} - \sum_l \mathbf{A}^{(kq_l' l)}, t\right)$. We prove this inequality by induction over t. It holds for $t = 0$, because $V^{AH}\left(\mathbf{X} - \sum_l \mathbf{A}^{(kq_l l)}, 0\right) = V^{AH}\left(\mathbf{X} - \sum_l \mathbf{A}^{(kq_l' l)}, 0\right) = 0$. Next, assume that the result holds for $t-1$. We now show that it will then also hold for *t*.

Therefore, let us consider the optimal policies $\mathbf{U}^{(\mathbf{q})*}$ and $\mathbf{U}^{(\mathbf{q}')*}$ at stage t for the problems $V^{AH}\left(\mathbf{X} - \sum_l \mathbf{A}^{(kq_l l)}, t\right)$ and $V^{AH}\left(\mathbf{X} - \sum_l \mathbf{A}^{(kq_l' l)}, t\right)$. We first restrict ourselves to the special case that $\mathbf{q}$ and $\mathbf{q}'$ may only have different values on one single flight leg l', that is, $q_l = q_l'\ \forall l \neq l'$ and $q_{l'} \geq q_{l'}'$. We refer to this as the single-leg monotonicity. The problems can then be rewritten as $V^{AH}\left(\left(\mathbf{X} - \sum_{l\neq l'} \mathbf{A}^{(kq_l l)}\right) - \mathbf{A}^{(kq_{l'}l')}, t\right)$ and $V^{AH}\left(\left(\mathbf{X} - \sum_{l\neq l'} \mathbf{A}^{(kq_l l)}\right) - \mathbf{A}^{(kq_{l'}'l')}, t\right)$, respectively. From the base case, it immediately follows that $u_{kl}^{(\mathbf{q})^*} = u_{kl}^{(\mathbf{q}')^*} \forall k, l \neq l'$. Now monotonicity can be shown componentwise for each of the terms that add up to the term within the outer brackets of (A.2), by relating the terms product by product. As the assignments may only differ on leg l', the argumentation completely follows the single-leg argumentation by Steinhardt and Gönsch (2012, Online Appendix A.1) and is thus omitted here. It follows that the inductive step holds with respect to single-leg monotonicity, which can more generally be stated as follows for an arbitrary resource l':

$$V^{AH}\left(\tilde{\mathbf{X}} - \mathbf{A}^{(k\tilde{q}'l')}, t\right) \leq V^{AH}\left(\tilde{\mathbf{X}} - \mathbf{A}^{(kql')}, t\right) \quad \forall t,\ \tilde{\mathbf{X}} \text{ with } \tilde{x}_{\tilde{q}'l'}, \tilde{x}_{\tilde{q}l'} > 0 \text{ and } \tilde{q} \geq \tilde{q}'. \qquad \text{(A.3)}$$

Based on this, we now consider arbitrary assignments $\mathbf{q} \geq \mathbf{q}'$ and the hypothesis

$$V^{AH}\left(\mathbf{X} - \sum_{l} \mathbf{A}^{(kq_l l)}, t\right) \leq V^{AH}\left(\mathbf{X} - \sum_{l<j} \mathbf{A}^{(kq'_l l)} - \mathbf{A}^{(kq'_j j)} - \sum_{l>j} \mathbf{A}^{(kq_l l)}, t\right) \forall j = 1, \ldots, s. \qquad \text{(A.4)}$$

For $j = 1$, (A.4) simplifies to

$$V^{AH}\left(\mathbf{X} - \sum_{l} \mathbf{A}^{(kq_l l)}, t\right) \leq V^{AH}\left(\left(\mathbf{X} - \sum_{l>1} \mathbf{A}^{(kq_l l)}\right) - \mathbf{A}^{(kq'_1 1)}, t\right) \qquad \text{(A.5)}$$

which holds because it directly follows from single-leg monotonicity (A.3), which has already been shown to hold for t. Specifically, the state of the corresponding two value functions in (A.5) only differs on the first flight leg. The inequality therefore directly follows from (A.3) by setting $\tilde{\mathbf{X}} = \mathbf{X} - \sum_{l>1} \mathbf{A}^{(kq_l l)}$, $\tilde{q} = q_1$, $\tilde{q}' = q'_1$, and $l' = 1$.

Next, we show that if (A.4) holds for $1 \leq j-1 < s$, it also holds for j:

$$V^{AH}\left(\mathbf{X} - \sum_{l} \mathbf{A}^{(kq_l l)}, t\right) \leq V^{AH}\left(\mathbf{X} - \sum_{l<j-1} \mathbf{A}^{(kq'_l l)} - \mathbf{A}^{(kq'_j j-1)} - \sum_{l>j-1} \mathbf{A}^{(kq_l l)}, t\right) =$$

$$V^{AH}\left(\mathbf{X} - \sum_{l<j} \mathbf{A}^{(kq'_l l)} - \mathbf{A}^{(kq_j j)} - \sum_{l>j} \mathbf{A}^{(kq_l l)}, t\right) \leq V^{AH}\left(\mathbf{X} - \sum_{l<j} \mathbf{A}^{(kq'_l l)} - \mathbf{A}^{(kq'_j j)} - \sum_{l>j} \mathbf{A}^{(kq_l l)}, t\right)$$

where the last inequality again follows from single-leg monotonicity, which becomes clear by setting $\tilde{\mathbf{X}} = \mathbf{X} - \sum_{l<j+1} \mathbf{A}^{(kq'_l l)} - \sum_{l>j+1} \mathbf{A}^{(kq_l l)}$, $\tilde{q} = q_j$, $\tilde{q}' = q'_j$, and $l' = j$ in (A.3). Thus, it also holds for $j = s$. It follows from this that the original inductive step also holds for the non-single-leg case that completes the proof:

$$V^{AH}\left(\mathbf{X} - \sum_{l} \mathbf{A}^{(kq_l l)}, t\right) \leq V^{AH}\left(\mathbf{X} - \sum_{l<s} \mathbf{A}^{(kq'_l l)} - \mathbf{A}^{(kq'_j s)} - \sum_{l>s} \mathbf{A}^{(kq_l l)}, t\right) = V^{AH}\left(\mathbf{X} - \sum_{l} \mathbf{A}^{(kq'_l l)}, t\right)$$

□

A.2. Proof of proposition 2

Proposition 2. $V^{PP}\left(\mathbf{X},\mathbf{Y},t\right)=V^{AH}\left(\psi\left(\mathbf{X},\mathbf{Y}\right),t\right)$ *for all* $t,\left(\mathbf{X},\mathbf{Y}\right)\in\mathcal{A}$

Proof. We first reformulate the value functions $V^{PP}\left(\mathbf{X},\mathbf{Y},t\right)$ and $V^{AH}\left(\mathbf{X},t\right)$ by means of an $n\times1$ policy vector $\mathbf{u}=\left(u_1,...,u_n\right)^T$ with $u_k\in\left\{0,1\right\}\forall k$ determined for each stage with the following meaning: if a request for product k arrives in the current time period, it will be denied if $u_k=0$, and accepted if $u_k=1$. Thus, (4) can be rewritten as

$$V^{PP}\left(\mathbf{X},\mathbf{Y},t\right)=\max_{\mathbf{u}\in\mathcal{U}^{PP}(\mathbf{X},\mathbf{Y})}\sum_{k|u_k=1}\lambda_k\left(t\right)\left(p_k+V^{PP}\left(\mathbf{X},\mathbf{Y}+\sum_l\mathbf{A}^{(kr_kl)},t-1\right)\right)+\left(1-\sum_{k|u_k=1}\lambda_k\left(t\right)\right)\cdot V^{PP}\left(\mathbf{X},\mathbf{Y},t-1\right)\tag{A.6}$$

with

$$\mathcal{U}^{PP}\left(\mathbf{X},\mathbf{Y}\right)=\left\{\mathbf{u}\,|\,u_k\in\left\{0,1\right\}\wedge\left(\mathbf{X},\mathbf{Y}+u_k\sum_l\mathbf{A}^{(kr_kl)}\right)\in\mathcal{A}\ \forall k\right\}\tag{A.7}$$

and boundary condition $V^{PP}\left(\mathbf{X},\mathbf{Y},0\right)=0$.

Regarding $V^{AH}\left(\mathbf{X},t\right)$, we use the reformulation (6) that is based on Proposition 1. To guarantee that the policy $\mathbf{u}$ is feasible with the remaining capacity $\mathbf{X}$, the vector must satisfy

$$\mathbf{u}\in\mathcal{U}^{AH}\left(\mathbf{X}\right)=\left\{\mathbf{u}\,|\,\mathbf{u}\in\left\{0,1\right\}^n\wedge\left(\left(u_k=0\right)\vee\left(\exists\mathbf{q}\geq\mathbf{1}_s\cdot r_k\,|\sum_l\mathbf{A}^{(kq_ll)}\leq\mathbf{X}\right)\forall k\right)\right\}.\tag{A.8}$$

Then, (6) can be rewritten as

$$V^{AH}\left(\mathbf{X},t\right)=\max_{\mathbf{u}\in\mathcal{U}^{AH}(\mathbf{X})}\left\{\sum_{k|u_k=1}\lambda_k\left(t\right)\cdot\left\{p_k+V^{AH}\left(\mathbf{X}-\sum_l\mathbf{A}^{(k,\min\{q\geq r_k|x_{ql}>0\},l)},t-1\right)\right\}+\left(1-\sum_{k|u_k=1}\lambda_k\left(t\right)\right)\cdot V^{AH}\left(\mathbf{X},t-1\right)\right\}\tag{A.9}$$

with the boundary condition $V^{AH}\left(\mathbf{X},0\right)=0$. Note that the simpler term $\min\left\{q\geq r_k\,|\,x_{ql}>0\right\}$ can be used instead of $q_l^*=\min\left\{q\geq r_k\,|\,x_{ql}>0\vee q=m\right\}$ as defined in (6). The or-condition ($\vee q=m$) capturing the case that there is no capacity left on one of the required flights cannot occur here because only feasible policies $\mathbf{u}\in\mathcal{U}^{AH}\left(\mathbf{X}\right)$ are considered.

First note that

$$\mathcal{U}^{AH}\left(\psi\left(\mathbf{X},\mathbf{Y}\right)\right)=\mathcal{U}^{PP}\left(\mathbf{X},\mathbf{Y}\right)\tag{A.10}$$

which follows directly from the single-leg proof in Steinhardt and Gönsch (2012, Online Appendix A.2). Specifically, let us consider an arbitrary vector $\mathbf{u}\in\mathcal{U}^{AH}\left(\psi\left(\mathbf{X},\mathbf{Y}\right)\right)$ with a vector component k with $u_k=1$. This implies that the request needs to be acceptable on each leg l, that is $\left(\exists q\geq r_k\,|\,\mathbf{A}^{(kql)}\leq\mathbf{X}\right)$, which corresponds exactly to the restriction imposed on the definition of $\mathcal{U}^{AH}\left(\mathbf{X}\right)$ in the single-leg case. Conversely, let us consider an arbitrary vector $\mathbf{u}\in\mathcal{U}^{PP}\left(\mathbf{X},\mathbf{Y}\right)$ with a vector component k with $u_k=1$, that is, request k can be accepted in the postponement dynamic program. Then note that the definition of $\mathcal{A}$ is separable in the legs, that is, for each leg l, the conditions $\sum_{r'\geq r} z_{rr'l}=y_{rl}\ \forall r$, $\sum_{r'\leq r} z_{r'rl}\leq x_{rl}\ \forall r$ and $z_{rr'l}\geq 0$ $\forall r,r'\geq r$ must hold. This exactly corresponds to the definition of $\mathcal{A}$ in the single-leg case and thus to the restrictions imposed on the definition of $\mathcal{U}^{PP}\left(\mathbf{X},\mathbf{Y}\right)$ therein. Therefore, the proof of (A.10) can be performed leg by leg. For each leg, it strictly follows the argumentation in the single-leg proof.

Moreover, from the argumentation there we also have

$$\psi\left(\mathbf{X},\mathbf{Y}+\sum_l \mathbf{A}^{(kr_kl)}\right)=\psi\left(\mathbf{X},\mathbf{Y}\right)-\sum_l \mathbf{A}^{(kr_l'l)} \text{ with } r_l'=min\left\{q\geq r_k \mid \psi\left(\mathbf{X},\mathbf{Y}\right)_{ql}>0\right\}\forall l. \quad \text{(A.11)}$$

We now perform induction over t to show Proposition 2. From (A.6), (A.9), and (A.10) it follows that the assumption holds for $t=0$ for all $\left(\mathbf{X},\mathbf{Y}\right)\in\mathcal{A}$:

$$V^{PP}\left(\mathbf{X},\mathbf{Y},0\right)=0=V^{AH}\left(\psi\left(\mathbf{X},\mathbf{Y}\right),0\right).$$

Next, we assume the result holds for $t-1$ and show that it holds for t. From (A.10), (A.11), and the inductive step for $t-1$ we have

$$\begin{aligned}
V^{PP}\left(\mathbf{X},\mathbf{Y},t\right)&=\max_{\mathbf{u}\in\mathcal{U}^{PP}(\mathbf{X},\mathbf{Y})}\sum_{k|u_k=1}\lambda_k(t)\left(p_k+V^{PP}\left(\mathbf{X},\mathbf{Y}+\sum_l \mathbf{A}^{(kr_kl)},t-1\right)\right)+\left(1-\sum_{k|u_k=1}\lambda_k(t)\right)\cdot V^{PP}\left(\mathbf{X},\mathbf{Y},t-1\right)\\
&=\max_{\mathbf{u}\in\mathcal{U}^{AH}(\psi(\mathbf{X},\mathbf{Y}))}\sum_{k|u_k=1}\lambda_k(t)\left(p_k+V^{PP}\left(\mathbf{X},\mathbf{Y}+\sum_l \mathbf{A}^{(kr_kl)},t-1\right)\right)+\left(1-\sum_{k|u_k=1}\lambda_k(t)\right)\cdot V^{PP}\left(\mathbf{X},\mathbf{Y},t-1\right)\\
&=\max_{\mathbf{u}\in\mathcal{U}^{AH}(\psi(\mathbf{X},\mathbf{Y}))}\left\{\sum_{k|u_k=1}\lambda_k\cdot\left\{p_k+V^{AH}\left(\psi\left(\mathbf{X},\mathbf{Y}+\sum_l \mathbf{A}^{(kr_kl)}\right),t-1\right)\right\}+\left(1-\sum_{k|u_k=1}\lambda_k\right)\cdot V^{AH}\left(\psi\left(\mathbf{X},\mathbf{Y}\right),t-1\right)\right\}\\
&=\max_{\mathbf{u}\in\mathcal{U}^{AH}(\psi(\mathbf{X},\mathbf{Y}))}\left\{\sum_{k|u_k=1}\lambda_k\cdot\left\{p_k+V^{AH}\left(\psi\left(\mathbf{X},\mathbf{Y}\right)-\sum_l \mathbf{A}^{(k,min\{q\geq r_k|\psi(\mathbf{X},\mathbf{Y})_{ql}>0\},l)},t-1\right)\right\}+\left(1-\sum_{k|u_k=1}\lambda_k\right)\cdot V^{AH}\left(\psi\left(\mathbf{X},\mathbf{Y}\right),t-1\right)\right\}\\
&=V^{AH}\left(\psi\left(\mathbf{X},\mathbf{Y}\right),t\right) \qquad\qquad for\ all\left(\mathbf{X},\mathbf{Y}\right)\in\mathcal{A}.
\end{aligned}$$

□

A.3. Proof of proposition 3

Proposition 3. $V^{PP}\left(\mathbf{X},\mathbf{Y},t\right)=V^{Surr}\left(\varphi\left(\mathbf{X}-\mathbf{Y}\right),t\right)$ *for all* $t,\left(\mathbf{X},\mathbf{Y}\right)\in\mathcal{A}$

Proof. To show the equivalence of (4) and (7), we first reformulate both value functions, similar to the proof of Proposition 2, by means of an $n\times 1$ policy vector $\mathbf{u}=\left(u_1,...,u_n\right)^T$ with $u_k\in\left\{0,1\right\}\forall k$ determined for each stage with the following meaning: if a request for product k arrives in the current time period, it will be denied if $u_k=0$, and accepted if $u_k=1$. Thus, we need to show that

$$V^{PP}\left(\mathbf{X},\mathbf{Y},t\right)=\max_{\mathbf{u}\in\mathcal{U}^{PP}\left(\mathbf{X},\mathbf{Y}\right)}\sum_k\lambda_k\left(t\right)\left(u_k p_k+V^{PP}\left(\mathbf{X},\mathbf{Y}+u_k\sum_l\mathbf{A}^{(kr_kl)},t-1\right)\right) \tag{A.12}$$

with $\mathcal{U}^{PP}\left(\mathbf{X},\mathbf{Y}\right)$ as defined in (A.7) and boundary condition $V\left(\mathbf{X},\mathbf{Y},0\right)=0$ is equivalent to

$$V^{Surr}\left(\mathbf{X}',t\right)=\max_{\mathbf{u}\in\mathcal{U}^{Surr}\left(\mathbf{X}'\right)}\sum_k\lambda_k\left(t\right)\left(u_k p_k+V^{Surr}\left(\mathbf{X}'-u_k\mathbf{A}^{(k)},t-1\right)\right) \tag{A.13}$$

with

$$\mathcal{U}^{Surr}\left(\mathbf{X}'\right)=\left\{\mathbf{u}\,|\,u_k\in\left\{0,1\right\}\wedge\mathbf{X}'-u_k\mathbf{A}^{(k)}\geq\mathbf{0}\,\forall k\right\} \tag{A.14}$$

and the boundary condition $V^{Surr}\left(\mathbf{X}',0\right)=0$.

To prove that $V^{PP}\left(\mathbf{X},\mathbf{Y},t\right)=V^{Surr}\left(\varphi\left(\mathbf{X}-\mathbf{Y}\right),t\right)\ \forall t,\left(\mathbf{X},\mathbf{Y}\right)\in\mathcal{A}$, we first show that the same product requests can be accepted in the postponement and in the surrogate formulation, that is,

$$\mathcal{U}^{PP}\left(\mathbf{X},\mathbf{Y}\right)=\mathcal{U}^{Surr}\left(\varphi\left(\mathbf{X}-\mathbf{Y}\right)\right)\ \forall\left(\mathbf{X},\mathbf{Y}\right)\in\mathcal{A}. \tag{A.15}$$

Note that given Proposition 2, this equation can be shown to follow from a proof stated in the context of static production planning models with machine flexibility by Leachman and Carmon (1992), which is based on reformulating the linear program as a transportation-type network flow problem and then applying Gale's flow feasibility theorem to show its equivalence to a corresponding surrogate formulation. In what follows, we present an alternative proof for the special case of upgrades in the airline context.

To prove (A.15), we need to show that $\mathbf{u}\in\mathcal{U}^{PP}\left(\mathbf{X},\mathbf{Y}\right)\Leftrightarrow\mathbf{u}\in\mathcal{U}^{Surr}\left(\varphi\left(\mathbf{X}-\mathbf{Y}\right)\right)$, which is equivalent to $\left(\mathbf{X},\mathbf{Y}+u_k\sum_l\mathbf{A}^{(kr_kl)}\right)\in\mathcal{A}\Leftrightarrow\varphi\left(\mathbf{X}-\mathbf{Y}\right)-u_k\mathbf{A}^{(k)}\geq\mathbf{0}\ \forall k$. Regarding the components of the right hand side of the latter equivalence, by using the definition of $\mathbf{A}^{(k)}$ and that of φ (Definition 2), we obtain

$$\left(\varphi\left(\mathbf{X}-\mathbf{Y}\right)-u_k\mathbf{A}^{(k)}\right)_{rl}=\varphi\left(\mathbf{X}-\mathbf{Y}\right)_{rl}-\sum_{i\geq r}u_k\mathbf{A}_{il}^{(kr_kl)}=\sum_{i\geq r}x_{il}-y_{il}-u_k\mathbf{A}_{il}^{(kr_kl)}=\varphi\left(\mathbf{X}-\left(\mathbf{Y}+u_k\mathbf{A}_{il}^{(kr_kl)}\right)\right)_{rl}\ \forall rl. \tag{A.16}$$

Thus, it suffices to show that $\left(\mathbf{X},\tilde{\mathbf{Y}}\right)\in\mathcal{A}\Leftrightarrow\varphi\left(\mathbf{X}-\tilde{\mathbf{Y}}\right)\geq\mathbf{0}$ with $\tilde{\mathbf{Y}}=\mathbf{Y}+u_k\mathbf{A}^{(kr_kl)}$ to prove that the same requests can be accepted.

We first show that $\left(\mathbf{X},\tilde{\mathbf{Y}}\right)\in\mathcal{A}\Rightarrow\varphi\left(\mathbf{X}-\tilde{\mathbf{Y}}\right)\geq\mathbf{0}$. From the definition of $\mathcal{A}$, we have $\left(\mathbf{X},\tilde{\mathbf{Y}}\right)\in\mathcal{A}\Rightarrow\exists z_{ril}\geq 0\,\forall r,i\geq r,l\,|\sum_{i\geq r}z_{ril}=\tilde{y}_{rl}\wedge\sum_{i\leq r}z_{irl}\leq x_{rl}\;\forall r,l$. Now, we have:

$$
\begin{aligned}
&\exists z_{ril}\geq 0\,\forall r,i\geq r,l\,|\sum_{i\geq r}z_{ril}=\tilde{y}_{rl}\wedge\sum_{i\leq r}z_{irl}\leq x_{rl}\;\forall r,l\\
&\Rightarrow\exists z_{ril}\geq 0\,\forall r,i\geq r,l\,|\sum_{i\geq r}z_{ril}=\tilde{y}_{rl}\wedge\sum_{r'\geq r}\sum_{i\leq r'}z_{ir'l}\leq\sum_{r'\geq r}x_{r'l}\;\forall r,l\\
&\Rightarrow\exists z_{ril}\geq 0\,\forall r,i\geq r,l\,|\sum_{i\geq r}z_{ril}=\tilde{y}_{rl}\wedge 0\leq\sum_{r'\geq r}x_{r'l}-\sum_{r'\geq r}\tilde{y}_{r'l}-\sum_{i<r}\sum_{r'\geq r}z_{ir'l}\;\forall r,l\\
&\Rightarrow\exists z_{ril}\geq 0\,\forall r,i\geq r,l\,|\sum_{i\geq r}z_{ril}=\tilde{y}_{rl}\wedge 0\leq\sum_{r'\geq r}x_{r'l}-\sum_{r'\geq r}\tilde{y}_{r'l}\;\forall r,l\\
&\Rightarrow\exists z_{ril}\geq 0\,\forall r,i\geq r,l\,|\sum_{i\geq r}z_{ril}=\tilde{y}_{rl}\wedge 0\leq\sum_{r'\geq r}\left(x_{r'l}-\tilde{y}_{r'l}\right)\;\forall r,l\\
&\Rightarrow 0\leq\varphi\left(\mathbf{X}-\tilde{\mathbf{Y}}\right)_{rl}\;\forall r,l\\
&\Rightarrow\mathbf{0}\leq\varphi\left(\mathbf{X}-\tilde{\mathbf{Y}}\right),
\end{aligned}
$$

where the first implication is obtained by adding up the conditions $\sum_{i\leq r}z_{irl}\leq x_{rl}$ for all compartments higher than r, the second by using $\sum_{r'\geq r}\sum_{i\leq r'}z_{ir'l}=\sum_{r'\geq r}\sum_{i\geq r'}z_{r'il}+\sum_{i<r}\sum_{r'\geq r}z_{ir'l}$ and $\sum_{i\geq r'}z_{r'il}=\tilde{y}_{r'l}$ and the third by omitting $\sum_{i<r}\sum_{r'\geq r}z_{ir'l}$ using that the $z_{ir'l}$ are nonnegative. The fourth implication is a simple rearrangement, the fifth directly follows from the definition of φ (Definition 2) and the last states the result for the whole matrix.

Next, we show that $\varphi\left(\mathbf{X}-\tilde{\mathbf{Y}}\right)\geq\mathbf{0}\Rightarrow\left(\mathbf{X},\tilde{\mathbf{Y}}\right)\in\mathcal{A}$, which is equivalent to showing that the feasibility problem (1)-(3) has a solution: $\varphi\left(\mathbf{X}-\tilde{\mathbf{Y}}\right)\geq\mathbf{0}\Rightarrow\exists z_{ril}\geq 0\,\forall r,i\geq r,l\,|\sum_{i\geq r}z_{ril}=\tilde{y}_{rl}\wedge\sum_{i\leq r}z_{irl}\leq x_{rl}\;\forall r,l$. To this end, we show that conditions (1)-(3) hold if we define the z_{ril} values recursively as follows:

$$
z_{ril}=min\left\{\tilde{y}_{rl}-\sum_{i'=i+1}^{m}z_{ri'l};x_{il}-\sum_{r'=r+1}^{i}z_{r'il}\right\}. \tag{A.17}
$$

Thinking algorithmically, the intuition behind (A.17) is for each leg l to assign commitments to compartments as high as possible. We start with the highest compartment m and first assign commitments y_{ml} that require this compartment. If capacity in compartment m remains unallocated, we move on to commitments $\tilde{y}_{m-1,l}$ that need at least the next lower compartment and assign them to compartment m. When all the capacity of compartment m is allocated, we move on to the next lower compartment $m-1$. Again, we first assign the remaining commitments $\tilde{y}_{m-1,l}-z_{m-1,m,l}$ that need this compartment. If capacity in compartment $m-1$ remains unallocated, we move on to commitments that need at least the next lower compartment and have not yet been assigned. This continues until all commitments are assigned to compartments. In doing so, the number of commitments assigned in each step is restricted by the number of commitments yet to be assigned (left term in the minimum) and the unallocated capacity (right term).

First, we need to show that constraints (3) are fulfilled given this definition, that is, $z_{ril} \geq 0 \, \forall r, i \geq r, l$. To see this, note that both terms in the minimum in (A.17) are nonnegative. Clearly, the left term is nonnegative for $i = m$. For $i < m$, the definition (A.17) ensures that $z_{ril} \leq \tilde{y}_{rl} - \sum_{i'>i} z_{ri'l}$ from which the nonnegativity of the left term recursively follows. The right term is clearly nonnegative for $i = r$. For $i > r$, the definition (A.17) ensures that $z_{ril} \leq x_{il} - \sum_{r<r'\leq i} z_{r'il}$ from which the nonnegativity of the right term recursively follows.

We now turn to constraints (2). For all legs l and compartments r, the corresponding capacity constraint $\sum_{i \leq r} z_{irl} \leq x_{rl}$ (2) directly follows (A.17) because from there we have $z_{1rl} = min\left\{ \tilde{y}_{1l} - \sum_{i'=r+1}^{m} z_{1i'l} ; x_{rl} - \sum_{r'=2}^{r} z_{r'rl} \right\} \leq x_{rl} - \sum_{r'=2}^{r} z_{r'rl}$ which represents the capacity constraint.

Finally, we must show that constraints (1) are fulfilled. Here, we show for all legs l and compartments r that $z_{rrl} = \tilde{y}_{rl} - \sum_{i'=r+1}^{m} z_{ri'l}$ which is equivalent to the minimum for z_{rrl} in (A.17) being on the left hand side. This means that capacity on compartment r (x_{rl}) is sufficient for all commitments for compartment r that have not already been assigned to higher compartments ($\tilde{y}_{rl} - \sum_{i'=r+1}^{m} z_{ri'l}$). For z_{rrl}, (A.17) simplifies to $z_{rrl} = min\left\{ \tilde{y}_{rl} - \sum_{i'=r+1}^{m} z_{ri'l} ; x_{rl} \right\}$ because the last sum can be omitted as its lower limit of summation ($r+1$) is greater than its upper limit of summation (r). Thus, the minimum in (A.17) for z_{rrl} being on the left hand side is equivalent to

$$\tilde{y}_{rl} - \sum_{i'=r+1}^{m} z_{ri'l} \leq x_{rl} , \tag{A.18}$$

To show that constraints (A.18) are fulfilled, we distinguish whether the left side or the right side of the minimum in definition (A.17) of $z_{r,r+1,l}$ applies:

- The left side applies if all commitments for compartment r have already been assigned to higher compartments. Then, we have $z_{r,r+1,l} = \tilde{y}_{rl} - \sum_{i'=r+2}^{m} z_{ri'l}$ and, clearly, the left hand side of (A.18) is 0 for z_{rrl} and (A.18) obviously holds.
- Let us now consider the case that the right side of (A.17) applies, that is, $z_{r,r+1,l} = x_{r+1l} - \sum_{r'=r+1}^{r+1} z_{r'r+1l}$, and, thus,

$$z_{ril} = x_{il} - \sum_{r'=r+1}^{i} z_{r'il} \forall i > r , \tag{A.19}$$

which follows from the recursive definition of (A.17). Condition (A.19) is quite interpretable, because it simply states that, in case that not all commitments for compartment r have already been assigned to higher compartments, there is no free capacity on these higher compartments.

Now, for $r = m$, we have

$$\begin{aligned}
&\varphi\left(\mathbf{X}-\tilde{\mathbf{Y}}\right)\geq \mathbf{0}\\
&\Rightarrow 0 \leq \sum_{r'\geq r}\left(x_{r'l}-\tilde{y}_{r'l}\right)\ \forall l\\
&\Rightarrow \sum_{r'\geq r}\tilde{y}_{r'l} \leq \sum_{r'\geq r} x_{r'l}\ \forall l\\
&\Rightarrow \sum_{r'\geq r}\tilde{y}_{r'l} - \sum_{i>r}\left(z_{ril}+\sum_{r'=r+1}^{i} z_{r'il}\right) \leq \sum_{r'\geq r} x_{r'l} - \sum_{i>r} x_{il}\ \forall l\\
&\Rightarrow \sum_{r'\geq r}\tilde{y}_{r'l} - \sum_{i>r}\left(z_{ril}+\sum_{r'=r+1}^{i} z_{r'il}\right) \leq x_{rl}\ \forall l\\
&\Rightarrow \tilde{y}_{rl} + \sum_{r'>r}\sum_{r''\geq r'} z_{r'r''l} - \sum_{i>r}\left(z_{ril}+\sum_{r'=r+1}^{i} z_{r'il}\right) \leq x_{rl}\ \forall l\\
&\Rightarrow \tilde{y}_{rl} + \sum_{r'>r}\sum_{r''\geq r'} z_{r'r''l} - \sum_{i>r} z_{ril} - \sum_{i>r}\sum_{r'=r+1} z_{r'il} \leq x_{rl}\ \forall l\\
&\Rightarrow \tilde{y}_{rl} - \sum_{i>r} z_{ril} \leq x_{rl}\ \forall r,l
\end{aligned}$$

where the third implication follows by subtracting $\sum_{i>r}\left(z_{ril}+\sum_{r'=r+1}^{i} z_{r'il}\right)=\sum_{i>r} x_{il}$ which is obtained by rearranging terms and adding up (A.19) for all $i > r$. For $r = m-1,...,1$, the same chain of implications applies, by inductively using $\tilde{y}_{r'l}=\sum_{i'=r'}^{m} z_{r'i'l}$ for $r' > r$ in the fifth implication.

Thus, we have $\varphi\left(\mathbf{X}-\tilde{\mathbf{Y}}\right)\geq \mathbf{0} \Rightarrow \left(\mathbf{X},\tilde{\mathbf{Y}}\right)\in\mathcal{A}\ \forall \mathbf{X},\tilde{\mathbf{Y}}$. Overall, this shows that $\mathcal{U}^{PP}\left(\mathbf{X},\mathbf{Y}\right)=\mathcal{U}^{Surr}\left(\varphi\left(\mathbf{X}-\mathbf{Y}\right)\right)$ holds for all $\mathbf{X},\mathbf{Y}$.

We now perform induction over t to show $V^{PP}\left(\mathbf{X},\mathbf{Y},t\right)=V^{Surr}\left(\varphi\left(\mathbf{X}-\mathbf{Y}\right),t\right)\ \forall t,\left(\mathbf{X},\mathbf{Y}\right)\in\mathcal{A}$ (Proposition 3). From (A.12), (A.13), and (A.15), it follows that the assumption holds for $t = 0$ for all $\left(\mathbf{X},\mathbf{Y}\right)\in\mathcal{A}$:

$$V^{PP}\left(\mathbf{X},\mathbf{Y},0\right)=0=V^{Surr}\left(\varphi\left(\mathbf{X}-\mathbf{Y}\right),0\right).$$

Next, we assume that the result holds for $t-1$ and show that it holds for t. In the following, the first equality is (A.12), the second follows from (A.15), the third is the inductive step for $t-1$, the fourth follows from (A.16), and the fifth is (A.13):

$$\begin{aligned}
&V^{PP}\left(\mathbf{X},\mathbf{Y},t\right)\\
&=\max_{\mathbf{u}\in\mathcal{U}^{PP}(\mathbf{X},\mathbf{Y})}\sum_{k}\lambda_k\left(t\right)\left(u_k p_k + V^{PP}\left(\mathbf{X},\mathbf{Y}+u_k\sum_{l}\mathbf{A}^{(kr_kl)},t-1\right)\right)
\end{aligned}$$

$$= \max_{\mathbf{u} \in \mathcal{U}^{Surr}(\varphi(\mathbf{X}-\mathbf{Y}))} \sum_k \lambda_k(t) \left(u_k p_k + V^{PP} \left(\mathbf{X}, \mathbf{Y} + u_k \sum_l \mathbf{A}^{(kr_k l)}, t-1 \right) \right)$$

$$= \max_{\mathbf{u} \in \mathcal{U}^{Surr}(\varphi(\mathbf{X}-\mathbf{Y}))} \sum_k \lambda_k(t) \left(u_k p_k + V^{Surr} \left(\varphi\left(\mathbf{X} - \left(\mathbf{Y} + u_k \mathbf{A}_{il}^{(kr_k l)} \right) \right), t-1 \right) \right)$$

$$= \max_{\mathbf{u} \in \mathcal{U}^{Surr}(\varphi(\mathbf{X}-\mathbf{Y}))} \sum_k \lambda_k(t) \left(u_k p_k + V^{Surr} \left(\varphi(\mathbf{X} - \mathbf{Y}) - u_k \mathbf{A}^{(k)}, t-1 \right) \right)$$

$$= V^{Surr} \left(\varphi(\mathbf{X} - \mathbf{Y}), t \right)$$

□

A.4. Supplementary material

In Table A.1, we present a detailed comparison of the revenues obtained with the different mechanisms. For example, the *SUCC* column contains the percentual revenue gain over *SUCC*. To avoid redundancy, only values for mechanisms that are usually superior to *SUCC* are reported here. For example, if *DPD-Surrogate* and *SUCC* are compared, the second line of the *SUCC* column contains the revenue gain of *DPD-Surrogate* over *SUCC*.

Table A.1: Relative performance at the 99% confidence level

Balanced **demand, α=0.9**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	0.09 ±0.19	1.77 ±0.78	2.26 ±0.72
DPD-Surrogate	-2.45 ±0.92	-0.82 ±1.14	
SUCC	-1.65 ±0.77		

StrongEco **demand, α=0.9**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	0.01 ±0.01	6.20 ±1.25	-0.22 ±0.53
DPD-Surrogate	0.23 ±0.53	6.44 ±1.03	
SUCC	-5.84 ±1.18		

Balanced **demand, α=1.0**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	0.38 ±0.44	1.95 ±0.84	4.36 ±1.10
DPD-Surrogate	-3.81 ±1.11	-2.31 ±1.16	
SUCC	-1.53 ±0.87		

StrongEco **demand, α=1.0**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	0.20 ±0.27	6.08 ±1.15	-0.75 ±0.57
DPD-Surrogate	0.96 ±0.66	6.89 ±1.20	
SUCC	-5.54 ±1.10		

Balanced **demand, α=1.1**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	2.00 ±0.58	2.60 ±0.78	0.01 ±0.47
DPD-Surrogate	1.99 ±0.69	2.58 ±0.87	
SUCC	-0.58 ±0.91		

StrongEco **demand, α=1.1**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	0.59 ±0.26	4.79 ±0.96	-1.41 ±0.55
DPD-Surrogate	2.03 ±0.62	6.30 ±0.78	
SUCC	-4.01 ±0.95		

Balanced **demand, α=1.2**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	3.10 ±0.66	1.99 ±0.73	0.09 ±0.49
DPD-Surrogate	3.01 ±0.72	1.90 ±0.75	
SUCC	1.09 ±0.89		

StrongEco **demand, α=1.2**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	1.11 ±0.39	3.05 ±1.09	-2.49 ±0.77
DPD-Surrogate	3.69 ±0.88	5.68 ±0.78	
SUCC	-1.88 ±1.15		

Balanced **demand, α=1.3**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	4.55 ±0.76	1.66 ±0.73	-0.13 ±0.57
DPD-Surrogate	4.70 ±0.84	1.79 ±0.75	
SUCC	2.85 ±0.97		

StrongEco **demand, α=1.3**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	1.49 ±0.44	2.27 ±1.17	-3.83 ±0.77
DPD-Surrogate	5.53 ±2.41	6.34 ±0.90	
SUCC	-0.76 ±1.15		

Balanced **demand, α=1.4**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	6.47 ±0.82	1.69 ±0.55	0.22 ±0.54
DPD-Surrogate	6.24 ±0.99	1.47 ±0.67	
SUCC	4.70 ±0.94		

StrongEco **demand, α=1.4**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	4.33 ±0.97	2.00 ±1.02	-3.37 ±0.86
DPD-Surrogate	7.98 ±1.07	5.56 ±0.73	
SUCC	2.29 ±1.14		

Balanced **demand, α=1.5**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	8.16 ±1.22	1.17 ±0.63	-0.06 ±0.60
DPD-Surrogate	8.23 ±1.26	1.23 ±0.74	
SUCC	6.91 ±1.24		

StrongEco **demand, α=1.5**

% over	FCFS	SUCC	DPD-Surrogate
DPD-Upgrade	5.54 ±0.92	0.84 ±1.18	-4.36 ±0.88
DPD-Surrogate	10.36 ±1.13	5.45 ±0.72	
SUCC	4.66 ±1.29		

References

Leachman RC, Carmon TF (1992) On capacity modeling for production planning with alternative machine types. IIE Trans 24:62–72

Steinhardt C, Gönsch J (2012) Integrated revenue management approaches for capacity control with planned upgrades. Eur J Oper Res 223:380–391