Dynamic programming decomposition for choice-based revenue management with flexible products

Sebastian Koch, Jochen Gönsch, Claudius Steinhardt

Sebastian Koch (Corresponding Author)

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

Phone: +49 821 598 4147
Fax: +49 821 598 4226
E-mail: sebastian.koch@wiwi.uni-augsburg.de

Jochen Gönsch

Chair of Service Operations, Mercator School of Management, University of Duisburg-Essen, Lotharstraße 65, 47057 Duisburg

E-mail: jochen.goensch@uni-due.de

Claudius Steinhardt

Chair of Business Analytics & Management Science, Bundeswehr University Munich (UniBw), Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany

E-mail: claudius.steinhardt@unibw.de

Abstract

We reconsider the stochastic dynamic program of revenue management with flexible products and customer choice behavior as proposed by Gallego et al. [Gallego G, Iyengar G, Phillips RL, Dubey A (2004) Managing flexible products on a network. Working paper, Columbia University, New York]. In the scientific literature on revenue management, as well as in practice, the prevailing strategy to operationalize dynamic programs is to decompose the network by resources and solve the resulting one-dimensional problems. However, to date, these dynamic programming decomposition approaches have not been applicable to problems with flexible products, because sold flexible products must be included in the dynamic program's state space and do not correspond directly to resources.

In this paper, we contribute to the existing research by presenting a general approach to operationalizing revenue management with flexible products and customer choice in a dynamic programming environment. In particular, we reformulate the original dynamic program by means of Fourier-Motzkin elimination to obtain an equivalent dynamic program with a standard resource-based state space. This reformulation allows the application of dynamic programming decomposition approaches. Numerical experiments show that the new approach has a superior revenue performance and that its average revenues are close to the upper bound on the optimal expected revenue from the choice-based deterministic linear program (CDLP). Moreover, our reformulation improves the revenues by up to 8% compared to an extended variant of a standard choice-based approach that immediately assigns flexible products after their sale.

Keywords: Revenue Management, Flexible Products, Dynamic Programming Decomposition, Customer Choice, Fourier-Motzkin Elimination

1 Introduction

While the specification of common, regular products is fixed in advance, a flexible product consists of two or more alternative specifications, such that the seller will assign the purchaser to one of these alternatives at a later point in time (see, e.g., Gallego et al. (2004)). From a revenue management point of view, this supply-side flexibility leads to improved capacity utilization and mitigates the negative impact of forecast errors that often occur due to demand's stochastic nature. From a marketing perspective, flexible products are an interesting tool for market segmentation. Owing to their inherent uncertainty, and because they are offered at a lower price, they are perceived as inferior by the customer and induce additional low value demand, while avoiding excess cannibalization. Flexible products have to be distinguished from opaque products, whose utilized resources the firm determines immediately after the sale, thus losing the benefit of postponing the products' assignment.

In this paper, we reconsider the problem of choice-based revenue management with flexible products, which Gallego et al. (2004) introduced. In their paper, these authors incorporated flexible products into the dynamic program (DP) for revenue management with arbitrary resource networks, while assuming choice-based demand behavior. To incorporate flexible products, they extended the state space of the DP from a purely resource-based one to a space that contains resources' remaining capacity as well as commitments reflecting sold flexible products that must be assigned to alternatives later on. An inherent feasibility problem ensures that the remaining capacity can satisfy the commitments.

In the traditional setting without flexible products, the standard way to make such a multidimensional DP operational is through dynamic programming decomposition (DPD). Standard DPD can be roughly summarized as follows: At first, a linear approximation of the corresponding DP is solved to obtain dual variables which capture the network effects. The dual variables are then used to decompose the network by resources. Doing so provides single-resource DPs that are easily solved to optimality. Within the single-resource DPs, only products that use the corresponding resource are considered and the capacity consumption of other resources is captured by reducing the products' revenues according to the dual variables from the linear approximation.

However, this approach is not applicable to the DP with flexible products, because its state space contains commitments that do not correspond directly to resources. In other words, if a flexible product is sold, the resources whose capacity is consumed are unknown, as they are determined later. Thus, an assignment to a single-resource DP and the immediate reduction of the remaining capacity are not possible if the flexibility should be preserved.

Our main contribution is that we show how to obtain an equivalent reformulation of the original DP whose state space is no longer based on commitments. The central idea is to apply Fourier-Motzkin elimination (FME; see, e.g., Schrijver (1998), Chapter 12.2) to the feasibility problem inherent in the DP. In doing so, additional "artificial" resources are added, allowing the flexible products to directly correspond to the artificial resources. This allows the reformulation as a standard revenue management problem without flexible products. We call this the surrogate approach. The key benefit of the new state space is that it enables the application of DPD and other standard methods, which make dynamic programming-based, large-scale implementations operational.

The remainder of this paper is structured as follows: In Section 2, we review the relevant scientific literature and position our work. In Section 3, we briefly summarize the standard DP of revenue management with flexible products, which Gallego et al. (2004) proposed, and restate the relevant notation. On this basis, we derive the surrogate reformulation in Section 4. By using numerical experiments, we evaluate DPD of the surrogate reformulation in Section 5. We use the upper bound from the optimal objective value of the choice-based deterministic linear program (CDLP), as well as two standard methods—the CDLP's primal solution and an adequate extension of the DPD ad hoc approach known from the literature—as benchmarks. In Section 6, we discuss our results and conclude the paper.

2 Related Literature

Initially, revenue management research assumed that demand was independent of the available products and of other customers (the well-known independent demand assumption). A considerable amount of work was done on the solution of single-resource problems (see, e.g., Littlewood (1972) for the earliest work; Belobaba (1987, 1989) for the expected marginal seat revenue heuristic; Lee and Hersh (1993), as well as Lautenbacher and Stidham (1999), for analyses of the exact DP formulation). However, as soon as networks of resources are considered, the corresponding DP formulations are difficult to solve even for small instances. Consequently, many heuristic approaches have been developed to approximate the DP. These approaches are mainly based on the idea of decomposing the network problem into a collection of smaller sub-problems. Decomposition is usually done by resources, while network effects are considered by adequately modifying the revenues of products that use more than one resource. The idea is to subsequently apply single-resource methods to the obtained sub-problems. We refer to Talluri and van Ryzin (2004b), Chapter 3.4, for an overview. Well-known decomposition approaches are origin-destination factor methods, fare proration (see, e.g., Kemmer et al. (2011) for an extension of the standard approach to large-scale applications), and DPD, which is most common in practice and literature, and is this paper's focus. The idea of DPD is to use the dual variables of a corresponding deterministic linear program (DLP) (see, e.g., Talluri and van Ryzin (1998)) to capture network effects and modify the products' revenues in the sub-problems. A further refinement of the standard DPD approach is studied in Zhang (2011). In addition, there are a few specific decomposition ideas (see, e.g., Cooper and Homem-de-Mello (2007), who include ideas from mathematical programming, and Birbil et al. (2014), who decompose the network by product types that require the same combination of resources).

Over the past decade, two major trends have emerged, which we will address in the following: first, the incorporation of demand-side substitution, which stems from customer choice behavior; second, the integration of supply-side substitution via flexible products.

Regarding the first trend, Talluri and van Ryzin (2004a) and Gallego et al. (2004) overcome the assumption of independent demand by considering customer choice behavior in the context of a single resource and a network of resources, respectively. In order to have a counterpart to the traditional DLP, Gallego et al. (2004) formulated the now well-known CDLP as a linear approximation of the underlying DP. Liu and van Ryzin (2008) and Miranda Bront et al. (2009) analyze the CDLP further. They assume that customer segments follow a standard multinomial logit model (see, e.g., Train (2009), Chapter 3) and that these segments consider buying products from disjoint and overlapping consideration sets, respectively. Gallego et al. (2015) reformulate the CDLP in respect of disjoint consideration sets; this reformulation avoids the exponential number of variables. Meissner et al. (2013) and Strauss and Talluri (2015) investigate weaker, but more efficient, deterministic linear approximations than the CDLP. Recent research has also examined many different customer choice models (see, e.g., Davis et al. (2014) for the nested logit model; Hosseinalifam (2014), Chapter 3, for a ranking-based customer choice model). Analogously to the independent demand setting, the CDLP is then used within an appropriate DPD approach. Liu and van Ryzin (2008) were the first to adapt the standard DPD to the choice-based setting. A number of subsequent papers have investigated this approach further (see Miranda Bront et al. (2009), as well as Kunnumkal and Topaloglu (2010), for a refinement of the standard DPD and derivations of upper bounds, respectively; Zhang and Adelman (2009), as well as Vossen and Zhang (2015b), for derivations of upper bounds and connections of DPD and the linear programming approach for approximate dynamic programming).

The second trend, that is, the consideration of flexible products, is also rooted in Gallego et al. (2004). These authors present a generalized DP formulation for flexible products in arbitrary resource networks that extends the state space by commitments. Their formulation, which also incorporates customer choice behavior, is standard for revenue management with flexible products today (see Section 3). However, subsequent research largely continued to follow the independent demand assumption. Among others, flexible products were investigated in the context of passenger aviation (see, e.g., Gallego and Phillips (2004)), air cargo revenue management (see, e.g., Bartodziej et al. (2006)), and the broadcasting industry (see, e.g., Kimms and Müller-Bungart (2007)).

Upgrades can be seen as a special case of flexible products with hierarchically ordered alternatives (see, e.g., Gallego and Stefanescu (2009)).

Overbooking problems with no shows are somehow also related, because there are also commitments in the state space of the corresponding DP formulations. At the end of the booking horizon, an optimization problem is solved to determine which reservations should be denied, which is similar to the feasibility problem inherent in the DP formulation with flexible products. However, in overbooking, commitments correspond directly to resources, which is similar to the traditional revenue management setting without flexible products. Erdelyi and Topaloglu (2010) are thus able to separate the optimization problem at the end of the booking horizon by resources, thus making standard DPD applicable. Similarly, Erdelyi and Topaloglu (2009) and Kunnumkal and Topaloglu (2008) approximate the optimization problem at the end of the booking horizon with a function that is separable by products reflecting the individual overbooking costs. In doing so, the authors are able to decompose the DP by products.

In contrast, in revenue management with flexible products, the standard decomposition by resources as in DPD is not possible, because the products do not correspond directly to resources. Basically, two literature streams tackling this issue have emerged:

- In the first stream, the supply-side flexibility is relinquished, allowing a flexible product to actually become an opaque product. Technically, the flexible product is irrevocably assigned immediately after the sale to one of the alternatives. We refer to Talluri (2001) and Chen et al. (2010) for revenue management with opaque products. In doing so, the need to store a commitment for later assignment is eliminated, and the solely resource-based state space is retained, which renders DPD possible again (see Gönsch and Steinhardt (2013) for DPD with opaque products). Other authors have sought to at least partially retain the flexibility. For example, Petrick et al. (2010, 2012) use bid prices from a deterministic linear programming (DLP) formulation and reassigned the sold flexible products when the DLP is resolved during the booking horizon.
- In the second stream, the supply-side flexibility is preserved at the cost of a restriction to special network structures. Often, only parallel resources and just one

flexible product are considered (see, e.g., Gallego and Phillips (2004) and Oosten (2004)). Gönsch and Steinhardt (2015) also considered DPD approaches, but restrict themselves to independent demand and airline upgrading, where hierarchical upgrades can be granted independently on each leg of a multi-leg flight. They use results from production planning that Leachman and Carmon (1992) obtained earlier.

This paper overcomes the drawbacks inherent in both literature streams. It enables DPD under customer choice behavior with arbitrary network structures, while fully retaining the supply-side flexibility. Additionally, in order to obtain a valid benchmark procedure for comparison, we adequately adapt an existing approach from choice-based revenue management to the flexible products setting. Our benchmark approach follows the idea of the first literature stream described above. In particular, it incorporates flexible products into the DPD approach of Liu and van Ryzin (2008) by immediately assigning them after sale.

3 Standard model formulation with flexible products

In the following, we first summarize the choice-based revenue management problem with flexible products (see Gallego et al. (2004)) and repeat the relevant notation (Section 3.1). Thereafter, we restate the corresponding DP (Section 3.2).

3.1 Problem formulation and notation

We consider a firm that sells regular products $j \in \mathcal{J} = \{1, \dots, n^{reg}\}$ and flexible products $k \in \mathcal{K} = \{1, \dots, n^{flex}\}$. These products use resources $h \in \mathcal{H} = \{1, \dots, m\}$ jointly and may be linked to sale restrictions or rules in order to segment the market. The customers arrive successively and stochastically over time before service provision. The regular products are associated with revenues $\boldsymbol{r}^{reg} = \left(r_1^{reg}, \dots, r_{n^{reg}}^{reg}\right)^T$. Furthermore, each regular product j has a capacity consumption $\boldsymbol{a}_j = \left(a_{1j}, \dots, a_{mj}\right)^T$, which is $a_{hj} = 1$ if product j uses resource h, and $a_{hj} = 0$ otherwise. Regarding the flexible products with revenues $\boldsymbol{r}^{flex} = \left(r_1^{flex}, \dots, r_{n^{flex}}^{flex}\right)^T$, the resources to be utilized can be decided just before service provision. More precisely, a customer who buys flexible product k is guaranteed the resources $\boldsymbol{a}_j$ of one of the alternative regular products

$j \in \mathcal{M}_k \subseteq \mathcal{J}$. The obtained revenue r_k^{flex} is fixed in advance and independent of this assignment.

For notational convenience, we use $(\boldsymbol{A}, \boldsymbol{\mathcal{M}})$ to denote the network structure, where $\boldsymbol{A} = [a_{hj}]_{m \times n^{reg}}$ is the regular products' capacity consumption matrix and, by slightly abusing notation, $\boldsymbol{\mathcal{M}} = (\mathcal{M}_1, \dots, \mathcal{M}_{n^{flex}})^T$ is the flexible products' vector of alternatives.

The state of the selling process is described by the remaining capacity $\boldsymbol{c} = (c_1, \dots, c_m)^T$ and the vector of commitments $\boldsymbol{y} = (y_1, \dots, y_{n^{flex}})^T$, which denotes the number of sold flexible products. Selling a regular product j reduces the remaining capacity to $\boldsymbol{c} - \boldsymbol{a}_j$, and selling a flexible product k increases the commitment vector to $\boldsymbol{y} + \boldsymbol{e}_k$, with $\boldsymbol{e}_k$ referring to the k-th standard basis vector in $\mathbb{R}^{n^{flex}}$.

We discretize the booking horizon into T time periods, such that in each period t there is, at most, one customer arrival. The periods are numbered backward in time, and w.l.o.g., the probability λ of a customer's arrival is time-homogeneous. Any capacity remaining at the end of the booking horizon is worthless and overbooking of the given resources' capacity is not allowed. In each period t, the firm's decision problem is to determine a subset of products to offer, called the offer set. Given an offer set $S \subseteq \mathcal{J} \cup \mathcal{K}$, an arriving customer purchases product j with probability $P_j^{reg}(S)$, product k with probability $P_k^{flex}(S)$, and makes no purchase with probability $P_0(S)$. The firm aims to maximize its total overall revenue.

In what follows, we omit the index sets of the products and resources where possible. For example, the notation $\forall k$ means $\forall k \in \mathcal{K}$, $\sum_k$ means $\sum_{k \in \mathcal{K}}$, and $\max_S$ means $\max_{S \subseteq \mathcal{J} \cup \mathcal{K}}$.

3.2 Dynamic programming formulation

Given the remaining capacity $\boldsymbol{c}$ and the commitments $\boldsymbol{y}$, the optimal expected revenue-to-go with t time periods left is denoted by $V_t(\boldsymbol{c}, \boldsymbol{y})$ and satisfies the Bellman equation (**DP-flex**)

$$V_t(\boldsymbol{c}, \boldsymbol{y}) = \max_S \Big\{ \textstyle\sum_j \lambda \cdot P_j^{reg}(S) \cdot \big(r_j^{reg} + V_{t-1}(\boldsymbol{c} - \boldsymbol{a}_j, \boldsymbol{y}) \big)$$

$$+\sum_k \lambda \cdot P_k^{flex}(S) \cdot \left(r_k^{flex} + V_{t-1}(\boldsymbol{c}, \boldsymbol{y} + \boldsymbol{e}_k)\right)$$
$$+(\lambda \cdot P_0(S) + 1 - \lambda) \cdot V_{t-1}(\boldsymbol{c}, \boldsymbol{y})\} \quad (1)$$

with the boundary conditions $V_t(\boldsymbol{c}, \boldsymbol{y}) = -\infty$ if $(\boldsymbol{c}, \boldsymbol{y}) \notin \mathcal{Z}_{(\boldsymbol{A}, \boldsymbol{\mathcal{M}})}$ and $V_0(\boldsymbol{c}, \boldsymbol{y}) = 0$ if $(\boldsymbol{c}, \boldsymbol{y}) \in \mathcal{Z}_{(\boldsymbol{A}, \boldsymbol{\mathcal{M}})}$.

The condition $(\boldsymbol{c}, \boldsymbol{y}) \in \mathcal{Z}_{(\boldsymbol{A}, \boldsymbol{\mathcal{M}})}$ describes a feasible state and holds if the capacity is non-negative and can satisfy all commitments in the given network structure $(\boldsymbol{A}, \boldsymbol{\mathcal{M}})$. More formally, $(\boldsymbol{c}, \boldsymbol{y}) \in \mathcal{Z}_{(\boldsymbol{A}, \boldsymbol{\mathcal{M}})}$ if and only if there exist (nonnegative and integer) distribution variables y_{kj} denoting how many commitments regarding flexible product k will be fulfilled with alternative j satisfying the **feasibility problem** (see Gallego et al. (2004)):

$$\sum_k \sum_{j \in \mathcal{M}_k} a_{hj} \cdot y_{kj} \leq c_h \qquad \forall h \qquad (2)$$
$$\sum_{j \in \mathcal{M}_k} y_{kj} = y_k \qquad \forall k \qquad (3)$$
$$y_{kj} \in \mathbb{Z}^+ \qquad \forall k, j \in \mathcal{M}_k \qquad (4)$$

To illustrate the problem, we introduce the following running example (expressed in airline terminology) that will also be reconsidered in Section 4 to illustrate the reformulation as well as the transformation we propose.

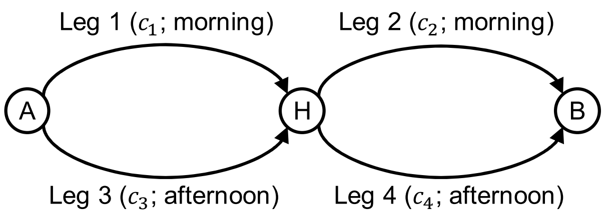

Figure 1: Airline network in running example

Example: Consider an airline that offers transportation from A to B over a hub H at two different times of day as depicted in Figure 1, resulting in a small network with $m = 4$ legs. There is one flexible product $k = 1$ that uses either legs $h = 1$ and $h = 2$ (alternative y_{11}) or legs $h = 3$ and $h = 4$ (alternative y_{22}). Table 1 represents the corresponding instance of the feasibility problem (constraints (2) and (3)) without integrality constraints, where the first column refers to the row number (rows (i)–(iv) are instances of constraint (2), row (v) is an instance of constraint (3)):

Row	y_{11}	y_{12}		Right-hand side
(i)	1		$\leq$	c_1
(ii)	1		$\leq$	c_2
(iii)		1	$\leq$	c_3
(iv)		1	$\leq$	c_4
(v)	1	1	$=$	y_1

Table 1: Feasibility problem in running example

4 Surrogate approach

The commitments in the state space of DP-flex (1) are obviously necessary to solve the feasibility problem (2)–(4) throughout the booking horizon. However, because the commitments do not correspond directly to resources, they inhibit the use of decomposition by resources.

To overcome this problem, we suggest applying FME in order to project the distribution variables y_{kj} out of the feasibility problem. In doing so, additional "artificial" resources are added, with the flexible products now corresponding directly to the artificial resources (Section 4.1). In Section 4.2, we use the DP formulation to show that this allows the reformulation as a standard revenue management problem without flexible products, such that standard solution approaches and heuristics can be used. In Section 4.3, we analyze the problem size of several network types in which flexible products occur in practice.

4.1 Transformation of the feasibility problem

In this subsection, we focus on the feasibility problem (2)–(4) by thinking of it as a static problem that must be solved for a given network $(\boldsymbol{A}, \boldsymbol{\mathcal{M}})$ at some point in time during the booking horizon, in order to decide whether a state $(\boldsymbol{c}, \boldsymbol{y})$ is feasible. We show how the distribution variables can be eliminated and explain the output of this elimination process.

In the feasibility problem, the integrality of the assignments of customers to alternatives is ensured by $y_{kj} \in \mathbb{Z}^+$ in (4). However, when projecting out the distribution variables by means of FME, we cannot keep this constraint and need a formulation that includes only $\leq$ constraints:

$$\sum_k \sum_{j \in \mathcal{M}_k} a_{hj} \cdot y_{kj} \leq c_h \qquad \forall h \tag{5}$$

$$\sum_{j \in \mathcal{M}_k} -y_{kj} \leq -y_k \qquad \forall k \tag{6}$$

$$-y_{kj} \leq 0 \qquad \forall k, j \in \mathcal{M}_k \tag{7}$$

Constraints (5)–(7) are the linear relaxation of (2)–(4). Whereas (5) and (7) correspond directly to (2) and (the relaxed) (4), constraints (6) may be less obvious. They follow from rewriting (3) as $\sum_{j \in \mathcal{M}_k} y_{kj} \geq y_k \; \forall k$, which is equivalent, because the feasible region defined by (2) and (the relaxed) (4) with regard to y_{kj} is a convex polytope including $\mathbf{0}$. In order to ensure that (2)–(4) can technically be replaced with (5)–(7), we claim that the following condition needs to hold:

Condition 1: If the linear relaxation (5)–(7) has an arbitrary solution, there exists also an integer solution (i.e., a solution given the same number of commitments which (additionally) satisfies (4)).

Please note that Condition 1 is satisfied in most applications. For example, it is a sufficient condition for Condition 1 to be valid that the left-hand side coefficient matrix of the feasibility problem (2)–(3) (or, equivalently, (5)–(6)) is totally unimodular, since adding an identity matrix as in (7) would preserve this property (see, e.g., Martin (1999), Chapter 14.2). Total unimodularity is, at least for relevant problem sizes, easy to check (see, e.g., Walter and Truemper (2013)). Please note that total unimodularity in our case refers to the complete feasibility problem including (3) and thus to a different matrix than in the common discussion in the revenue management literature, where it refers to the left-hand side matrix of the well-known deterministic linear program (DLP; see, e.g., Talluri and van Ryzin (2004b)), that is, (2) without (3) but with an additional identity matrix resulting from demand constraints. It is well-known that total unimodularity is satisfied, for example, in problems for which corresponding network flow formulations can be constructed. For our feasibility problem, the construction of such network flow formulations can be performed analogously to the construction for the DLP (see, e.g., Glover et al. (1982), as well as Bertsimas and Popescu (2003) for the construction of network flow formulations in origin-destination networks, and Chen (1998) for hotel networks). Even more, recall that total unimodularity is only a sufficient condition, and there are also many other settings without total unimodularity which satisfy Condition 1.

Note that in the case that Condition 1 does not hold, network instances whose capacity is slightly overestimated could potentially result from the linear relaxation (see also Proposition 1 and the note thereafter).

Now, we can project the distribution variables y_{kj} out of (5)–(7) by using FME. The classical FME idea can be summarized as follows: Consider that we want to project variable x out of the inequality system $LB_i \leq a_i \cdot x \; \forall i$, $b_j \cdot x \leq UB_j \; \forall j$. This inequality system has a feasible solution if and only if $\max_i \frac{LB_i}{a_i} \leq \min_j \frac{UB_j}{b_j}$, which is equal to the system of linear inequalities $b_j \cdot LB_i \leq a_i \cdot UB_j \; \forall i, j$. Therefore, the initial inequality system can be replaced equivalently by the latter constraints.

We can now apply this idea to our setting, considering one distribution variable after another. The important point here is that we also treat the state of the selling process as variables. To formalize this approach, let **LHS** and **RHS** be the left-hand side and right-hand side coefficient matrices of (5)–(7), respectively. With the $\sum_k |\mathcal{M}_k| \times 1$ vector of distribution variables denoted by $\boldsymbol{y}^{kj} = \left(y_{kj}\right)_{\forall k, j \in \mathcal{M}_k}$, (5)–(7) can be rewritten as:

$$\mathbf{LHS} \cdot \boldsymbol{y}^{kj} \leq \mathbf{RHS} \cdot (\boldsymbol{c}^T | \boldsymbol{y}^T | \mathbf{0}^T)^T \tag{8}$$

Please note that **LHS** and **RHS** only depend on the network structure $(\boldsymbol{A}, \boldsymbol{\mathcal{M}})$ and are the coefficients. Now, Algorithm 1 creates a projection of (8) by applying a sequence of FMEs to the distribution variables. It can briefly be explained as follows: The distribution variables are projected out step-wise (line 1), considering one column (i.e., distribution variable) of **LHS** after another. In an iteration, the row indices of the current feasibility problem are partitioned by their coefficient of **LHS** into sets of rows with positive, negative, and null coefficients (lines 2–5). Based on this, the new number of rows (after eliminating the current distribution variable) is determined (line 6), and an arbitrary indexation of these rows is introduced (line 7). Finally, in lines 8–15, the current distribution variable is projected out. Please note that $\mathbf{LHS}_{row}$ ($\mathbf{RHS}_{row}$) refers to the row-th row of **LHS** (**RHS**).

Algorithm 1: Elimination of distribution variables from the feasibility problem		
1	**for** $d = 1$ **to** $\sum_k \lvert\mathcal{M}_k\rvert$ **do**	▷ consider one distribution variable y_{kj}
2	$nRows \longleftarrow$ number of rows of $\mathbf{LHS}$	
3	$Pos \longleftarrow \{row \in \{1, \dots, nRows\}: \text{lhs}_{row,d} > 0\}$	▷ partition row indices
4	$Neg \longleftarrow \{row \in \{1, \dots, nRows\}: \text{lhs}_{row,d} < 0\}$	
5	$Null \longleftarrow \{row \in \{1, \dots, nRows\}: \text{lhs}_{row,d} = 0\}$	
6	$nRows \longleftarrow \lvert Null \cup (Pos \times Neg)\rvert$	▷ number of rows after eliminating the current distribution variable
7	Let $biject$ be a bijection that maps $\{1, \dots, nRows\}$ onto $Null \cup (Pos \times Neg)$	▷ $biject$ is an arbitrary indexation of the new rows after the elimination
8	**for** $row = 1$ **to** $nRows$ **do**	▷ construct a new row
9	**if** $biject(row) \in Null$ **then**	▷ copy row without change
10	$\mathbf{LHS}^{new}_{row} \longleftarrow \mathbf{LHS}_{biject(row)}$ and $\mathbf{RHS}^{new}_{row} \longleftarrow \mathbf{RHS}_{biject(row)}$	
11	**else**	▷ $biject(row) \in Pos \times Neg$ and add these rows
12	$(pos, neg) \longleftarrow biject(row)$	
13	$\mathbf{LHS}^{new}_{row} \longleftarrow \text{lhs}_{neg,d} \cdot \mathbf{LHS}_{pos} + \text{lhs}_{pos,d} \cdot \mathbf{LHS}_{neg}$	
14	$\mathbf{RHS}^{new}_{row} \longleftarrow \text{lhs}_{neg,d} \cdot \mathbf{RHS}_{pos} + \text{lhs}_{pos,d} \cdot \mathbf{RHS}_{neg}$	
15	Set $\mathbf{LHS} \longleftarrow \mathbf{LHS}^{new}$ and $\mathbf{RHS} \longleftarrow \mathbf{RHS}^{new}$	
16	**return RHS**	▷ return only **RHS**, because $\mathbf{LHS} = \mathbf{0}$

Additionally, as pointed out by Bertsimas and Tsitsiklis (1997), Chapter 2.8, redundant rows should be regularly eliminated while performing FME (see, e.g., Paulraj and Sumathi (2010) on finding redundant constraints in linear inequality systems).

Example (cont'd): Returning to the example from Section 3.2, the algorithm is illustrated by Table 2. The first set of rows (i)–(vii) refers to the initial feasibility problem, the second (i')–(vii') to the feasibility problem after projecting out y_{11}, and the third (i'')–(viii'') to the transformed feasibility problem after projecting out y_{12}. The last column refers to the operation performed to obtain the row.

In the first set of rows, there are only $\leq$-constraints according to (5)–(7). Now, to project out y_{11}, the rows with null coefficient ($Null = \{$(iii),(iv),(vii)$\}$) are copied without change (resulting in rows (i')–(iii')). The rows with positive coefficient ($Pos = \{$(i),(ii)$\}$ and negative coefficient ($Neg = \{$(v),(vi)$\}$) are added according to lines 13–14 of the algorithm (resulting in rows (iv')–(vii')). The second iteration is performed analogously, finally leading to the transformed feasibility problem given by rows (i'')–(viii''). The transformed feasibility problem obviously consists of non-negativity constraints of regular resources' remaining capacity, i.e., $0 \leq c_h \; \forall h = 1, \dots, 4$, as well as four additional constraints that each check non-negativity of the sum of two regular resources' remaining capacity less the commitments for the flexible product:

$$\begin{pmatrix}0\\0\\0\\0\end{pmatrix} \leq \begin{pmatrix}1\\1\\0\\0\end{pmatrix} \cdot c_1 + \begin{pmatrix}0\\0\\1\\1\end{pmatrix} \cdot c_2 + \begin{pmatrix}1\\0\\1\\0\end{pmatrix} \cdot c_3 + \begin{pmatrix}0\\1\\0\\1\end{pmatrix} \cdot c_4 - \begin{pmatrix}1\\1\\1\\1\end{pmatrix} \cdot y_1$$

Row	y_{11}	y_{12}		Right-hand side	Operation
(i)	1		$\leq$	c_1	
(ii)	1		$\leq$	c_2	
(iii)		1	$\leq$	c_3	
(iv)		1	$\leq$	c_4	
(v)	-1	-1	$\leq$	$-y_1$	
(vi)	-1		$\leq$	0	
(vii)		-1	$\leq$	0	
(i')		1	$\leq$	c_3	(iii)
(ii')		1	$\leq$	c_4	(iv)
(iii')		-1	$\leq$	0	(vii)
(iv')		-1	$\leq$	$c_1 - y_1$	(i) + (v)
(v')			$\leq$	c_1	(i) + (vi)
(vi')		-1	$\leq$	$c_2 - y_1$	(ii) + (v)
(vii')			$\leq$	c_2	(ii) + (vi)
(i'')			$\leq$	c_1	(v')
(ii'')			$\leq$	c_2	(vii')
(iii'')			$\leq$	c_3	(i') + (iii')
(iv'')			$\leq$	c_4	(ii') + (iii')
(v'')			$\leq$	$c_1 + c_3 - y_1$	(i') + (iv')
(vi'')			$\leq$	$c_1 + c_4 - y_1$	(ii') + (iv')
(vii'')			$\leq$	$c_2 + c_3 - y_1$	(i') + (vi')
(viii'')			$\leq$	$c_2 + c_4 - y_1$	(ii') + (vi')

Table 2: Algorithm 1 in running example

In general, we obtain the following transformed feasibility problem (9) and (10) which has the same form as the feasibility check in the traditional setting without flexible products, in which only the non-negativity of the remaining capacity is checked (see, e.g., Talluri and van Ryzin (2004b), Chapter 3.2):

$$0 \leq c_h \qquad \forall h \in \mathcal{H} \tag{9}$$

$$0 \leq \sum_h \tilde{f}_{ih} \cdot c_h - \sum_k \tilde{b}_{ik} \cdot y_k \qquad \forall i \in \widetilde{\mathcal{H}} = \{1, \dots, \widetilde{m}\} \tag{10}$$

Constraints (9), in which the non-negativity of the regular resources' remaining capacity c_h is checked, are in fact identical. Constraints (10) can be interpreted as analogous conditions that require the non-negativity of some additional, artificial resources

$\tilde{\mathcal{H}} = \{1, \ldots, \tilde{m}\}$. Please note that the number $\tilde{m}$ of artificial resources as well as the values of $\tilde{f}_{ih}$ and $\tilde{b}_{ik}$ are determined by FME. In Section 4.3, we investigate the number $\tilde{m}$ subject to different network types.

Each artificial resource i has a capacity of $\tilde{c}_i = \sum_h \tilde{f}_{ih} \cdot c_h - \sum_k \tilde{b}_{ik} \cdot y_k$ and can be considered a pool of several regular resources that captures their alternative usage: It pools the capacity of some resources (those with $\tilde{f}_{ih} = 1$) and is required by regular products needing these resources, as well as by one or more flexible products that use these resources alternatively (those with $\tilde{b}_{ik} = 1$).

To ease notation in the following, we group the coefficients $\tilde{f}_{ih}$ and $\tilde{b}_{ik}$ into $\tilde{\boldsymbol{f}}_h = \tilde{\boldsymbol{f}}_h(\boldsymbol{A}, \boldsymbol{\mathcal{M}})$ and $\tilde{\boldsymbol{b}}_k = \tilde{\boldsymbol{b}}_k(\boldsymbol{A}, \boldsymbol{\mathcal{M}})$. Furthermore, we group the artificial resources represented by the right hand side of (10) into $\tilde{\boldsymbol{c}} = \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) = \sum_h \tilde{\boldsymbol{f}}_h(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot c_h - \sum_k \tilde{\boldsymbol{b}}_k(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot y_k$. Finally, we define $\tilde{\boldsymbol{a}}_j = \tilde{\boldsymbol{a}}_j(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) = \sum_h \tilde{\boldsymbol{f}}_h(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot a_{hj}$, which can be interpreted as a regular product's capacity consumption of artificial resources. Given these definitions, (9) and (10) can be abbreviated to

$$\boldsymbol{c} \geq \boldsymbol{0} \tag{11}$$

$$\tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) \geq \boldsymbol{0}, \tag{12}$$

and we can state the following result:

Proposition 1: Given Condition 1 holds, $(\boldsymbol{c}, \boldsymbol{y}) \in \mathcal{Z}_{(\boldsymbol{A}, \boldsymbol{\mathcal{M}})}$ (that is, the feasibility problem (2)–(4) has a solution) if and only if $\left(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y})\right) \geq \boldsymbol{0}$ (that is, (11)–(12) has a solution).

Proof: See Appendix A.

Note that if Condition 1 does not hold, the transformed feasibility problem in (11)–(12) becomes heuristic, and the artificial resources slightly overestimate capacity for flexible product k by, at most, $|\mathcal{M}_k|$ capacity units.

Example (cont'd): To illustrate the notation in our example, there are $\tilde{m} = 4$ artificial resources. Resource $h = 1$ is included in artificial resources $i = 1$ and $i = 2$, ($\tilde{\boldsymbol{f}}_1 = (1,1,0,0)^T$) and so on, that is, $\tilde{\boldsymbol{f}}_2 = (0,0,1,1)^T$, $\tilde{\boldsymbol{f}}_3 = (1,0,1,0)^T$, $\tilde{\boldsymbol{f}}_4 = (0,1,0,1)^T$. The flexible product consumes capacity on all artificial resources ($\tilde{\boldsymbol{b}}_1 = (1,1,1,1)^T$).

4.2 Reformulation as a standard revenue management problem

We next consider the dynamic revenue management problem again and show how Proposition 1 allows for managing the sales process of flexible products. A straightforward application of the transformed feasibility problem would be to replace (2)–(4) repeatedly to check which products can be offered for sale. More precisely, consider the check whether a regular product j (a flexible product k) can be sold given the current state $(\boldsymbol{c},\boldsymbol{y})$. One could, of course, reduce the regular resources' remaining capacity to $\boldsymbol{c}-\boldsymbol{a}_j$ (increase the commitments to $\boldsymbol{y}+\boldsymbol{e}_k$), then apply FME, and finally check whether $\left(\boldsymbol{c},\tilde{\boldsymbol{c}}\left(\boldsymbol{A},\boldsymbol{\mathcal{M}},\boldsymbol{c}-\boldsymbol{a}_j,\boldsymbol{y}\right)\right)\geq\boldsymbol{0}$ $\left(\left(\boldsymbol{c},\tilde{\boldsymbol{c}}(\boldsymbol{A},\boldsymbol{\mathcal{M}},\boldsymbol{c},\boldsymbol{y}+\boldsymbol{e}_k)\right)\geq\boldsymbol{0}\right)$ has a solution. However, it is not necessary to repeat FME so frequently throughout the booking horizon, which we will show in the following.

Proposition 2: Let $(\boldsymbol{c},\boldsymbol{y})\in\mathcal{Z}_{(\boldsymbol{A},\boldsymbol{\mathcal{M}})}$ be an arbitrary state of DP-flex. Then, we have

(a) $\left(\boldsymbol{c}-\boldsymbol{a}_j,\boldsymbol{y}\right)\in\mathcal{Z}_{(\boldsymbol{A},\boldsymbol{\mathcal{M}})}$ if and only if $\left(\boldsymbol{c}-\boldsymbol{a}_j,\tilde{\boldsymbol{c}}\left(\boldsymbol{A},\boldsymbol{\mathcal{M}},\boldsymbol{c}-\boldsymbol{a}_j,\boldsymbol{y}\right)\right)\geq\boldsymbol{0}\ \forall j$ and

$(\boldsymbol{c},\boldsymbol{y}+\boldsymbol{e}_k)\in\mathcal{Z}_{(\boldsymbol{A},\boldsymbol{\mathcal{M}})}$ if and only if $\left(\boldsymbol{c},\tilde{\boldsymbol{c}}(\boldsymbol{A},\boldsymbol{\mathcal{M}},\boldsymbol{c},\boldsymbol{y}+\boldsymbol{e}_k)\right)\geq\boldsymbol{0}\ \forall k$

(b) $\left(\boldsymbol{c}-\boldsymbol{a}_j,\tilde{\boldsymbol{c}}\left(\boldsymbol{A},\boldsymbol{\mathcal{M}},\boldsymbol{c}-\boldsymbol{a}_j,\boldsymbol{y}\right)\right)=\left(\boldsymbol{c}-\boldsymbol{a}_j,\tilde{\boldsymbol{c}}(\boldsymbol{A},\boldsymbol{\mathcal{M}},\boldsymbol{c},\boldsymbol{y})-\tilde{\boldsymbol{a}}_j(\boldsymbol{A},\boldsymbol{\mathcal{M}})\right)\ \forall j$ and

$\left(\boldsymbol{c},\tilde{\boldsymbol{c}}(\boldsymbol{A},\boldsymbol{\mathcal{M}},\boldsymbol{c},\boldsymbol{y}+\boldsymbol{e}_k)\right)=\left(\boldsymbol{c},\tilde{\boldsymbol{c}}(\boldsymbol{A},\boldsymbol{\mathcal{M}},\boldsymbol{c},\boldsymbol{y})-\tilde{\boldsymbol{b}}_k(\boldsymbol{A},\boldsymbol{\mathcal{M}})\right)\ \forall k$

Proof: Expression (a) obviously follows from Proposition 1. Appendix B.1 provides the proof of expression (b).

Expression (a) implies that the decision whether a regular product j (a flexible product k) can be offered in state $(\boldsymbol{c},\boldsymbol{y})\in\mathcal{Z}_{(\boldsymbol{A},\boldsymbol{\mathcal{M}})}$ need not be made by the original feasibility check (5)–(7), but can be equivalently made by using (11)–(12) instead. Expression (b) implies that, after a sale of a regular product j (of a flexible product k), FME need not be repeated. Instead, the regular and artificial resources capacity $\left(\boldsymbol{c},\tilde{\boldsymbol{c}}(\boldsymbol{A},\boldsymbol{\mathcal{M}},\boldsymbol{c},\boldsymbol{y})\right)$ can simply be reduced by $\left(\boldsymbol{a}_j,\tilde{\boldsymbol{a}}_j(\boldsymbol{A},\boldsymbol{\mathcal{M}})\right)$ (by $\left(\boldsymbol{0},\tilde{\boldsymbol{b}}_k(\boldsymbol{A},\boldsymbol{\mathcal{M}})\right)$), in order to obtain the transformed feasibility problem of the following state.

By exploiting the previous results, we can derive an alternative DP formulation. Consider an arbitrary instance of the transformed feasibility problem given by regular and artificial resources' remaining capacity $\boldsymbol{c}$ and $\tilde{\boldsymbol{c}}$, respectively, as well as by the coefficients $\tilde{\boldsymbol{f}}_h$, $\tilde{\boldsymbol{b}}_k$, and $\tilde{\boldsymbol{a}}_j$. We define the DP's state space directly as $(\boldsymbol{c}, \tilde{\boldsymbol{c}})$. Let $V_t^{surr}(\boldsymbol{c}, \tilde{\boldsymbol{c}})$ denote the optimal expected revenue-to-go with t periods left, which can be computed recursively using the following Bellman equation (**DP-surr**):

$$V_t^{surr}(\boldsymbol{c}, \tilde{\boldsymbol{c}}) = \max_S \Big\{ \sum_j \lambda \cdot P_j^{reg}(S) \cdot \left(r_j^{reg} + V_{t-1}^{surr}(\boldsymbol{c} - \boldsymbol{a}_j, \tilde{\boldsymbol{c}} - \tilde{\boldsymbol{a}}_j) \right)$$
$$+ \sum_k \lambda \cdot P_k^{flex}(S) \cdot \left(r_k^{flex} + V_{t-1}^{surr}(\boldsymbol{c}, \tilde{\boldsymbol{c}} - \tilde{\boldsymbol{b}}_k) \right)$$
$$+ (\lambda \cdot P_0(S) + 1 - \lambda) \cdot V_{t-1}^{surr}(\boldsymbol{c}, \tilde{\boldsymbol{c}}) \Big\} \tag{13}$$

with the boundary conditions $V_t^{surr}(\boldsymbol{c}, \tilde{\boldsymbol{c}}) = -\infty$ if $(\boldsymbol{c}, \tilde{\boldsymbol{c}}) \ngeq \boldsymbol{0}$ and $V_0^{surr}(\boldsymbol{c}, \tilde{\boldsymbol{c}}) = 0 \; \forall (\boldsymbol{c}, \tilde{\boldsymbol{c}}) \geq \boldsymbol{0}$. Subsequently, we can formulate the following result:

Proposition 3: $V_t(\boldsymbol{c}, \boldsymbol{y}) = V_t^{surr}\big(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y})\big)$ for all t, $(\boldsymbol{c}, \boldsymbol{y})$.

Proof: See Appendix B.2.

Propositions 2 and 3 imply that DP-flex (1) is indeed fully equivalent to DP-surr (13). The optimal expected value from the initial state is identical, and both DP formulations can be thought of as being carried out in parallel. More precisely, they allow the same decision options (due to Proposition 2 (a)) and virtually make the same decision (due to Proposition 2 (b) and Proposition 3). Thus, DP-surr can be used instead of DP-flex, we can drop the commitments from the state space, and, instead, track regular and artificial resources' capacity.

Accordingly, it suffices to apply FME only once to a given network, namely at the beginning of the booking horizon. In doing so, we fully retain the supply-side flexibility by postponing the assignment of flexible products. From a technical point of view, flexible products now correspond directly to resources. A flexible product is treated like a regular one; that is, the remaining capacity of artificial resources i is immediately reduced by $\tilde{b}_{ik}$, and the regular resources are left unchanged. In comparison, a regular product j requires—apart from the standard consumption $\boldsymbol{a}_j$—one unit of capacity of artificial resource i for every resource pooled in i and used by j; that is, $\tilde{a}_{ij} = \sum_h \tilde{f}_{hi} \cdot a_{hj}$. Thus, DP-surr (13) has the same form as a standard revenue management problem without flexible products. This finally enables the application of standard solution approaches like DPD.

We call the network of products and resources underlying DP-surr the *surrogate* network of the original network underlying DP-flex. The surrogate networks are not only a technical output of FME, but are usually quite intuitive.

Example (cont'd): In the running example, the artificial resources are interpretable as upper limits on the total amount of the flexible product that can be sold. Obviously, the number of sales is restricted by the legs out of A ($\tilde{c}_1 = c_1 + c_3$) and into B ($\tilde{c}_3 = c_2 + c_4$). Additionally, remember that passengers travel either in the morning (legs 1 and 2) or in the afternoon (legs 3 and 4). If legs 1 and 4 are fully booked, no additional flexible product can be sold. This restriction is captured by $\tilde{c}_2 = c_1 + c_4$. Similarly, legs 2 and 3 may be the bottleneck ($\tilde{c}_4 = c_2 + c_3$). Together, these four artificial resources consider in an meaningful way all four combinations of bottlenecks for the number of sales that may occur and restrict sales of the flexible product to $\min\{c_1 + c_2\} + \min\{c_3 + c_4\}$.

Later, we also give an analogous interpretation of the example networks used in the numerical experiments (see Sections 5.2.2 and 5.3.2).

4.3 Network types and size of the surrogate networks

In the previous subsection, we have shown how an arbitrary revenue management problem with flexible products can be reformulated as an equivalent standard revenue management problem without flexible products. In this subsection, we focus on the size of the resulting surrogate network. This is an important issue, since FME can, in general, add a large number of constraints and there are examples where the number of added constraints is exponential in the problem size. Clearly, such examples also exist in the context of revenue management. When formulating the surrogate network, two or more of the m regular resources form an artificial resource, which a subset of the flexible products then uses. Thus, we have a maximum of $\tilde{m} = 2^m$ artificial resources. Examples that reach this upper bound can be easily constructed.

However, the networks (or, often, subnetworks), in which flexible products actually occur in practice, usually have a structure in which the number of artificial resources is mostly far less and stays polynomially bounded. It is important to observe that the number of artificial resources only depends on the structure of flexible products; that is, their alternatives $\mathcal{M}_k$. This number is completely independent of the regular products and also independent of flexible products' prices. That is, it does not increase if, besides an existing flexible product k, a second flexible product k' is added with identical alternatives $\mathcal{M}_{k'} = \mathcal{M}_k$, but a different revenue and/or demand. Moreover, if arbitrary flexible products are deleted from a network, the number of artificial resources never increases. In the following, we thus focus on flexible products' structure and consider several network types whose flexible products are frequently used along with the resulting number of artificial resources.

Network type 1 *(fully flexible parallel resources)*

This network type consists of m parallel resources and a single flexible product that may be assigned to each of the m resources: $\boldsymbol{A} = [\boldsymbol{E}_{m\times m}|\cdot]$, $\mathcal{M}_1 = \{1, \dots, m\}$, where the columns to the right of $\boldsymbol{E}_{m\times m}$ denote the arbitrary resource consumption of regular products.

In practice, this network type occurs, for example, in the travel industry. Many tour operators offer travel roulette, which assigns customers to one of several similar hotels in their destination area.

In terms of the surrogate network, there is only one artificial resource that pools the capacity of all regular resources; that is, $\tilde{c}_1 = \sum_{h=1}^{m} c_h$. This resource is used by all products.

Proposition 4: In network type 1, the number of artificial resources is $\widetilde{m} = 1$ (and thus constant in the number of regular resources m).

We show Proposition 4 by induction in Appendix C.1.

Network type 2 (*pairwise flexible parallel resources*)

This network type consists of m parallel resources and $m-1$ flexible products. Flexible product k may be assigned to resource k or $k+1$: $\boldsymbol{A} = [\boldsymbol{E}_{m\times m}|\cdot]$, $\mathcal{M}_k = \{k, k+1\}\ \forall k$.

This network type arises, for example, in manufacturing. In a chain of factories, each factory k might be able to produce products $k-1$ and k. When quantifying the benefit of flexibility, Jordan and Graves (1995) find that such a chain of factories yields nearly the same output as a set of fully flexible factories, that can each produce every product. However, we leave out the last link in the chain here; there is no product that can be produced alternatively in factory m or 1. Another example is upgrading to the next higher resource in revenue management with parallel resources, for example, single-resource airline revenue management. Among others, Gallego and Stefanescu (2009) consider this upgrading "limited-cascading upgrading," while Shumsky and Zhang (2009) consider it "single-step upgrading." Moreover, the flexibility can also relate to time if, for example, guests' alternative stays on a cruise ship are considered.

Regarding the surrogate network, for all $\underline{h} \in \{1, \dots, m-1\}$ and $\overline{h} \in \{\underline{h}+1, \dots, m\}$, there is an artificial resource $\tilde{c}_{\underline{h}\overline{h}} = \sum_{h=\underline{h}}^{\overline{h}} c_h$ that pools the capacity of the adjacent regular resources $\underline{h}$ to $\overline{h}$. Artificial resource $\tilde{c}_{\underline{h}\overline{h}}$ is jointly used by flexible products $\underline{h}$ to $\overline{h}-1$.

Proposition 5: In network type 2, the number of artificial resources is $\widetilde{m} = \frac{(m-1)\cdot m}{2}$ (and thus polynomial in the number of regular resources m).

We show Proposition 5 by induction in Appendix C.2.

Network type 3 (*adjacent flexible parallel resources*)

This is a generalization of network type 2 and consists of m parallel resources, but $\frac{(m-1)\cdot m}{2}$ flexible products. More precisely, for all $\underline{h} \in \{1, \dots, m-1\}$ and $\overline{h} \in \{\underline{h}+1, \dots, m\}$, there is a flexible product that may be assigned to the adjacent resources $\underline{h}$ to $\overline{h}$. Therefore, the flexible product k may be described in terms of the topologically first alternative $\underline{h}_k$ and last alternative $\overline{h}_k$, and we have: $\boldsymbol{A} = [\boldsymbol{E}_{m\times m} | \cdot]$, $\mathcal{M}_k = \{h | \underline{h}_k \leq h \leq \overline{h}_k\} \, \forall k$.

In practice, this network type arises in generalizations of type 2, where, for example, a factory can produce more than two products; that is, some factories h are able to produce products $h-2, h-1$, and h. In single-resource revenue management, Gallego and Stefanescu (2009) have termed this generalization "full-cascading upgrading," while Shumsky and Zhang (2009) call it "multi-step upgrading."

We obtain the same set of artificial resources as in network type 2. However, the artificial resource $\tilde{c}_{\underline{h}\overline{h}}$ that pools capacity from $\underline{h}$ to $\overline{h}$ is now shared by all flexible products whose first, as well as last alternative, is between $\underline{h}$ and $\overline{h}$ ($\mathcal{M}_k \subseteq \{\underline{h}, \underline{h}+1, \dots, \overline{h}\}$).

Proposition 6: In network type 3, the number of artificial resources is $\widetilde{m} = \frac{(m-1)\cdot m}{2}$ (and thus polynomial in the number of regular resources m).

The structure of the proof is similar to the proof of Proposition 5 and omitted.

Network type 4 (*independent flexible block wise resources*)

This network type has a block structure, which consists of resource blocks $bl \in \mathcal{BL} = \{1, \dots, n^{block}\}$ with (w.l.o.g.) an identical number of m^{block} resources in each block, such that there are $m = m^{block} \cdot n^{block}$ resources in total. A flexible product k simultaneously uses resources from an arbitrary subset of blocks $\mathcal{BL}_k \subseteq \mathcal{BL}$. If k uses a block bl, the flexibility there is defined according to network type 3 (types 1 and 2 are also possible): $\boldsymbol{A}^{bl} = [\boldsymbol{E}_{m^{block} \times m^{block}} | \cdot]$, $\mathcal{M}_k^{bl} = \{h | \underline{h}_{k,bl} \le h \le \overline{h}_{k,bl}\} \, \forall k$. The key point here is that the blocks are independent in the sense that the final assignment of flexible product k to a regular resource in one block is independent of its assignment in other blocks; that is, an alternative $j \in \mathcal{M}_k$ combines arbitrary $\mathcal{M}_k^{bl}$ for all $bl \in \mathcal{BL}_k$ and $|\mathcal{M}_k| = \prod_{bl \in \mathcal{BL}_k} |\mathcal{M}_k^{bl}|$.

In practice, this setting occurs, for example, in multi-stage production processes (see, e.g., Leachman and Carmon (1992)), in which the blocks correspond to the stages and the resources in a stage correspond to its machines. Another example is upgrading in network airline revenue management. Here, the blocks correspond to legs in the network and a ticket (i.e., a flexible product) encompasses one or more legs. On each leg, the passenger may be arbitrarily upgraded to a higher compartment (i.e., another resource in this block), for example, from economy to business class (see, e.g., Gönsch and Steinhardt (2015)).

In respect of the surrogate network, each resource block can be considered independently. In each block bl, we obtain $\frac{(m^{block}-1)\cdot m^{block}}{2}$ artificial resources. As in network types 2 and 3, each resource $\tilde{c}_{\underline{h}\overline{h}}$ pools the capacity of the adjacent regular resources $\underline{h}$ to $\overline{h}$ for all $\underline{h} \in \{1, \dots, m^{block} - 1\}$ and $\overline{h} \in \{\underline{h} + 1, \dots, m^{block}\}$.

Proposition 7: In network type 4, the total number of artificial resources is $\widetilde{m} = n^{blocks} \cdot \frac{(m^{block}-1)\cdot m^{block}}{2} = \frac{m\cdot(m^{block}-1)}{2}$ (and thus polynomial in the number of regular resources m).

The proof is similar to the proof of Proposition 5 and omitted.

Network type 5 (*dependent flexible block wise resources*)

Like network type 4, this network type has a block structure, which consists of resource blocks $bl \in \mathcal{BL} = \{1, \dots, n^{block}\}$. Again, a flexible product k simultaneously uses resources from an arbitrary subset of blocks $\mathcal{BL}_k \subseteq \mathcal{BL}$.

In contrast to network type 4, we now consider resource types. There are the same resource types $h \in \{1, \dots, n^{rt}\}$ in each resource block. Let the tuple (h, bl) refer to a resource of type h from a specific block bl, and let $c_{h,bl}$ denote its capacity, such that there are $m = n^{block} \cdot n^{rt}$ resources in total. The resource types follow a nested upgrade hierarchy. A higher index indicates a higher position in the hierarchy; that is, a more versatile resource.

The flexible product k is associated with resource type h_k; that is, it can be assigned to h_k or upgraded to any $h > h_k$. Regarding block bl, we have: $\boldsymbol{A}^{bl} = [\boldsymbol{E}_{n^{rt}\times n^{rt}} | \cdot]$, $\mathcal{M}_{kh}^{bl} = \{h\}\ \forall k, h \geq h_k$. The important point here is that the assignment of product k must be the same $h \geq h_k$ for all blocks; that is, an alternative $j \in \mathcal{M}_k$ combines the $\mathcal{M}_{kh}^{bl}$ for all $bl \in \mathcal{BL}_k$ and one $h \geq h_k$. Thus, we have $|\mathcal{M}_k| = n^{rt} - h_k + 1$.

The car rental industry is one of the most important users of such upgrades (see, e.g., Geraghty and Johnson (1997), Pachon et al. (2003), and Fink and Reiners (2006)). There, the blocks correspond to the days of the planning horizon and the resource types in a block correspond to different car types following a given upgrade hierarchy (e.g., economy, compact, and full-size car types). Another example is upgrading in the hotel industry, where the resource types correspond to different room types.

Regarding the surrogate network, we cannot consider the resource blocks independently anymore, because the assignments must be the same for all blocks. In the following, for all $h \in \{1, \dots, n^{rt} - 1\}$, we define
$\mathcal{W}_h = \left\{\{(h, bl_h), (h+1, bl_{h+1}), \dots, (n^{rt}, bl_{n^{rt}})\} | bl_{h'} \in \mathcal{BL}\ \forall h' \in \{h, \dots, n^{rt}\}\right\}$. Each

element $w \in \mathcal{W}_h$ refers to a set of resources with exactly one resource $(h', bl_{h'})$ of each type $h' \geq h$ from an arbitrary block $bl_{h'}$. Thus, w contains $n^{rt} - h + 1$ resources. As all combinations of blocks are considered, $\mathcal{W}_h$ contains $(n^{block})^{n^{rt}-h+1}$ sets of resources. Now, there is one artificial resource $\tilde{c}_{hw}$ for each $w \in \mathcal{W}_h$ and $h \in \{1, \dots, n^{rt} - 1\}$ that simply adds up the capacity of the resources in w:

$$\tilde{c}_{hw} = \sum_{(h',bl) \in w} c_{h',bl} \quad \forall w \in \mathcal{W}_h, h \in \{1, \dots, n^{rt} - 1\} \quad (14)$$

Let $\mathcal{BL}^{w}$ denote the set of resource blocks from which resources are contained in $w \in \mathcal{W}_h$. The artificial resource $\tilde{c}_{hw}$ is shared by all products with $h_k \geq h$ and $\mathcal{BL}_k \supseteq \mathcal{BL}^{w}$.

Proposition 8: In network type 5, the total number of artificial resources is $\widetilde{m} = \sum_{k=2}^{n^{rt}} (n^{block})^k$ (and thus polynomial in the number of blocks and exponential in the number of resource types).

The structure of the proof is similar to the proof of Proposition 5 and is omitted.

Please note that the number of resource types is relatively small and constant in most practical applications. For example, in the car rental industry, there are often three to five car types in the upgrade hierarchy. In contrast, the number of resource blocks considered varies across rental stations and is often subject to the individual decision maker. Thus, also in this setting, the problem stays polynomially bounded in the relevant, potentially scalable problem parameters, that is, the resource blocks.

5 Computational experiments

In this section, we evaluate the revenue performance of the surrogate approach from Section 4. We use two airline networks introduced by Liu and van Ryzin (2008), which became de facto standard test instances for choice-based revenue management (see, e.g., Miranda Bront et al. (2009); Meissner and Strauss (2012)). We describe the experimental setup in Section 5.1, and we evaluate the approaches' revenue performance in detail in Sections 5.2 and 5.3, separately for the two networks under consideration.

5.1 Experimental setup

We summarize the implemented revenue management methods in Section 5.1.1. Please note that the technical details are provided in Appendix D. Furthermore, we describe the customer choice behavior in Section 5.1.2 and explain the consideration of forecast uncertainty in Section 5.1.3.

5.1.1 Implemented revenue management methods

Our main method is *DPD-surr*, which implements the surrogate approach described in the previous section. In this method, the surrogate reformulation is solved with the DPD approach of Liu and van Ryzin (2008). Details can be found in Appendix D.4. As benchmarks, we implemented the two methods *DPD-ah* and *CDLP-surr* as well as an upper bound (*UB*) on the optimal expected revenue of DP-flex (1):

- *DPD-ah* is a DPD approach that forgoes flexibility and immediately assigns flexible products (ad hoc) after sale (see Appendix D.2). Several studies report a good revenue performance of this approach in settings with independent demand (see Section 2). We incorporated this ad hoc assignment into the DPD approach of Liu and van Ryzin (2008).
- *CDLP-surr* refers to the optimal primal solution of the corresponding CDLP formulation (D.1.8)–(D.1.12) that gives us the time a set S should be offered during the booking horizon (see Appendix D.3). This is in line with a benchmark used by Liu and van Ryzin (2008).
- *UB* is the upper bound obtained from the optimal objective value of the CDLP (see Appendix D.1). This value can be obtained by either solving the CDLP model with flexible products (CDLP-flex (D.1.1)–(D.1.6); see Gallego et al. (2004)) or by using the surrogate reformulation in the standard CDLP formulation without flexible products (CDLP-surr (D.1.8)–(D.1.12); see, e.g., Liu and van Ryzin (2008) and Miranda Bront et al. (2009)).

All algorithms were implemented in MATLAB (Version 8, Release R2013a). Linear programs were solved by the function linprog from the Optimization Toolbox. We use

Monte Carlo simulation to evaluate the described methods and report values averaged over 200 customer streams for each problem instance.

5.1.2 Customer choice behavior

We assume the same choice behavior as Liu and van Ryzin (2008). Therefore, the choice model and the notation required to describe our computational experiments are only summarized in brief. Each customer belongs to a segment $l \in \mathcal{L}$, and customers from l are only interested in a subset of the entire product set, namely their consideration set C_l. Furthermore, the consideration sets are disjoint for customers belonging to different segments. With probability λ_l, a customer from segment l arrives. Her segment-specific purchase probabilities, that is, $P_{lj}^{reg}(S)$ for regular product j, $P_{lk}^{flex}(S)$ for flexible product k, and $P_{l0}(S)$ for the no-purchase alternative, are given by the standard multinomial logit model. They are computed using her product-specific preference weights, denoted by the parameters v_{lj}^{reg}, v_{lk}^{flex}, and v_{l0} for regular product j, flexible product k, and the no-purchase alternative, respectively. Then, for this choice model, the purchase probability is computed by

$$P_{lj}^{reg}(S) = \frac{v_{lj}^{reg}}{\sum_{j \in C_l \cap S} v_{lj}^{reg} + \sum_{k \in C_l \cap S} v_{lk}^{flex} + v_{l0}} \qquad (15)$$

for a regular product. For a flexible product and the no-purchase alternative, only the numerator changes. Please note that, because of the assumption of a multinomial logit model and disjoint consideration sets, the large number of (column generation) subproblems arising in *DPD-surr*, *DPD-ah*, *CDLP-surr*, and *UB* (i.e., the problems determining the offer set) can be solved efficiently by a simple ranking procedure (see Liu and van Ryzin (2008)).

Regarding the segment probabilities' temporal distribution, we consider two arrival patterns. The first one models *time-homogenous* demand. In the second arrival pattern, called *mixed*, we consider that low-value demand tends to arrive earlier. This pattern is obtained by assuming that 50% of demand is time-homogenous, and 50% arrives according to the classical low-before-high assumption. We straightforwardly adapt our models by introducing time-dependent arrival probabilities in the DPD approaches and in the CDLP approximations used therein.

5.1.3 Forecast uncertainty

To incorporate forecast uncertainty, we implemented stochastic forecast errors as studied in Petrick et al. (2012) to disturb the regular products' preference weights. The forecast errors are itinerary-based. All forecasted preference weights concerning a specific itinerary are disturbed by the same factor. Therefore, a uniformly distributed random number $\hat{\delta} \in \mathcal{U}(-\delta, +\delta)$ is drawn within every simulation run for each of the regular products' itineraries, and the corresponding preference weights are multiplied by the factor $(1 + \hat{\delta})$. The size of the error is controlled by the error bound $\delta \in [0,1]$.

5.2 Network 1: Parallel flights

In the following sections, we explain how we modified the first example from Liu and van Ryzin (2008) to include flexible products (Section 5.2.1) and interpret the corresponding surrogate reformulation (Section 5.2.2). We then evaluate the approaches' revenue performance in detail (Section 5.2.3).

5.2.1 Network description

Network 1 consists of three parallel legs with capacity $\boldsymbol{c} = (30, 50, 40)^T$ that can be thought of as flights on the same route at different times of day. On each leg, the firm offers a high fare class regular product (products 1–3) and a low fare class regular product (products 4–6). The prices are given by $\boldsymbol{r}^{reg} = (800, 1000, 600, 400, 500, 300)^T$. In addition, we consider a flexible product (product f) which can be sold at a price of $r_f^{flex} = 240$. The flexible product guarantees transportation on one of the three legs.

There is a high fare class customer segment H (consideration set $C_H = \{1,2,3\}$), a low fare class segment L (consideration set $C_L = \{4,5,6\}$) and a flexible segment F (consideration set $C_F = \{f\}$). The preference vectors are given by $v_H = (5,10,1)^T$, $v_L = (5,1,10)^T$, and $v_F = (10)$.

As usual in revenue management experiments, we tested different network loads. We varied the scarcity of capacity using a capacity factor $\alpha \in \{0.4, 0.5, \ldots, 1.2\}$. Different customer attitudes were captured by four no-purchase preference vectors $v_0 =$

$(v_{H0}, v_{L0})^T$, that is, $(0.01, 0.01)^T$, $(1, 5)^T$, $(5, 10)^T$, and $(10, 20)^T$. The flexible customer segment's preference weight for the no-purchase alternative is 0.01, and the number of periods is set to 300.

In the *time-homogenous* arrival pattern, customers of segments H, L, and F arrive with probabilities $\lambda_H = 0.2$, $\lambda_L = 0.3$, and $\lambda_F = 0.1$, respectively. Accordingly, in the *mixed* arrival pattern, the probabilities $(\lambda_H, \lambda_L, \lambda_F)^T$ are $(0.1, 0.15, 0.35)^T$, $(0.1, 0.45, 0.05)^T$, and $(0.4, 0.15, 0.05)^T$ in periods 300–251, 250–101, and 100–1, respectively.

5.2.2 Surrogate reformulation

The surrogate network consists of four resources: the three regular resources and one artificial resource with capacity $c_1 + c_2 + c_3$. It can readily be interpreted. The artificial resource pools the capacity of all resources that can potentially be used to fulfill the flexible product. More precisely, it represents the maximum amount of flexible and regular products that can be sold, because they all jointly use the capacity of the three legs. Accordingly, if a product is actually sold, the artificial resource's capacity is reduced. A flexible product needs capacity on this artificial resource alone, because the number of flexible product sales is constrained only by the joint capacity of the three legs. By contrast, a regular product requires one unit of capacity on 'its' regular resource and one unit of capacity on the artificial resource. The consumption of the regular resource reflects that one seat fewer is now available on this leg. The consumption of the artificial resource reflects that the seat can be used neither for a flexible nor a regular product.

5.2.3 Performance evaluation

Figure 2 shows the average revenues of *DPD-surr, DPD-ah*, and *CDLP-surr* relative to *UB* in all scenarios subject to the capacity factor α. Each column relates to a specific no-purchase preference vector, and each row represents one of the two arrival patterns (*time-homogenous* or *mixed*). Forecast errors are not considered here.

In general, all three methods' revenue performance is rather good, as expected from the literature on DPD without flexible products (see, e.g., Miranda Bront et al. (2009)). There seems to be no major impact of the arrival pattern. *DPD-ah* usually yields 94%–

98% of *UB* and *CDLP-surr* yields 96%–98%. *DPD-surr* attains even higher revenues of 97%–99% of *UB*. As usual, revenue management is relatively easy for extreme network load factors. If capacity is very scarce ($\alpha = 0.4$ and $(v_{0H}, v_{0L})^T = (0.01, 0.01)^T$), only the high fare class products are offered. Similarly, revenue management becomes more or less obsolete when all products are offered in case of ample capacity ($\alpha \geq 1.0$), and all methods yield revenues close to *UB*. But for intermediate capacity, where revenue management is most relevant, considerable differences can be observed. Here, *DPD-surr* shows a very stable revenue performance, whereas *DPD-ah* yields considerably lower revenues in many cases. This is most obvious for preference weights of $(v_{0H}, v_{0L})^T = (0.01, 0.01)^T$. In both the *time-homogeneous* and the *mixed* arrival patterns, *DPD-ah*'s revenue falls to under 90% at $\alpha = 0.8$, whereas *DPD-surr* still attains about 98% and *CDLP-surr* remains at 96%.

Next, we focus on the relative performance of the two DPD methods and consider forecast errors. Figure 3 shows the revenue gain of *DPD-surr* over *DPD-ah*, subject to the upper error bound δ on the forecast uncertainty. To keep the figure simple, we only depict the most relevant capacity factors ($\alpha \in \{0.5, 0.6, 0.7, 0.8\}$). Furthermore, we tested whether these revenue gains are significant at the 99% level of confidence. We calculated the revenue difference together with the empirical standard deviation on a per-stream basis and conducted a standard paired t-test. If the 99% confidence interval of the revenue difference does not include zero, the gain is significant. For reasons of clarity and because all confidence intervals are similar in size, we only included error bars for the top and bottom lines in the plots of Figure 3.

For the original setting without forecast errors ($\delta = 0$), we observe revenue gains of around 1% and 2% in the majority of cases. In general, the gains increase with higher forecast uncertainty. The higher the δ, the more important it is to use flexible products to mitigate demand uncertainty. Obviously, *DPD-surr* can benefit considerably from retaining full flexibility of the requests already accepted. This is in line with an observation from Petrick et al. (2010) who obtained similar results regarding linear programming-based heuristics that retained flexibilities to varying degrees.

At first glance, it seems strange that there is almost no influence of the forecast error in network 1 for preference weights of $(v_{0H}, v_{0L})^T = (0.01, 0.01)^T$. This is due to the special demand structure in this standard setting where the probability for the no-purchase alternative is almost zero, as long as a product can be bought ($P_{l0}(S) \approx 0 \; \forall S \cap C_l \neq \emptyset$). The customers' preference weights, which are disturbed by the forecast error, essentially do not influence whether a customer buys, they only influence which product she buys. However, correctly anticipating this decision is not important, because the products in a customer's consideration set have similar revenues, and if a leg is fully booked, only the other legs' products are offered and bought with probability one.

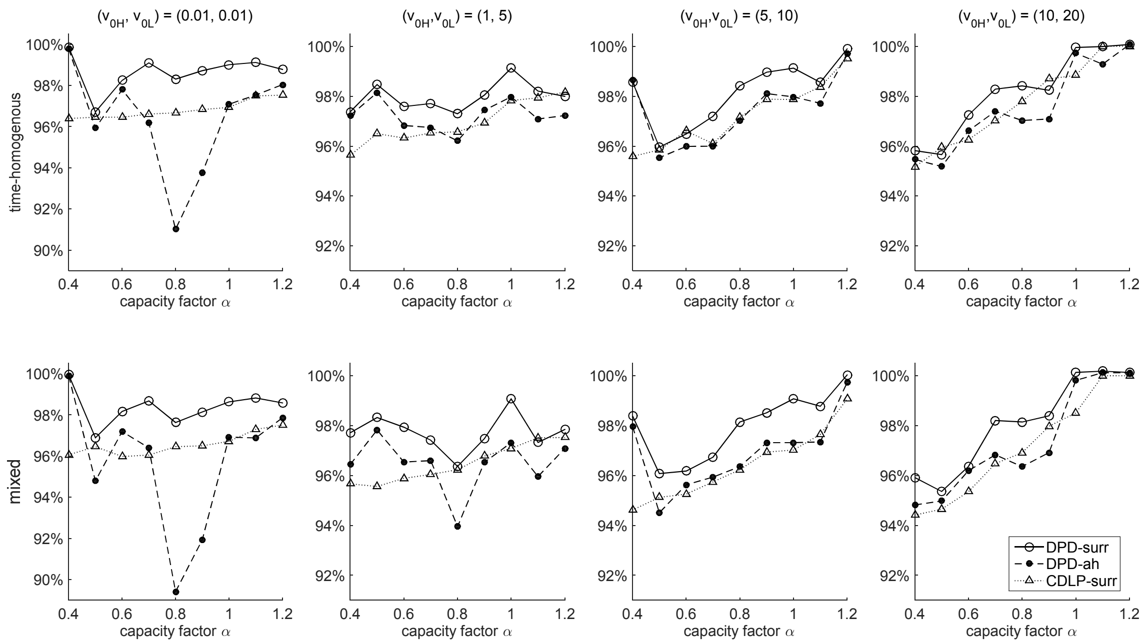

Figure 2: Average revenues of *DPD-surr*, *DPD-ah*, and *CDLP-surr* relative to *UB* in network 1

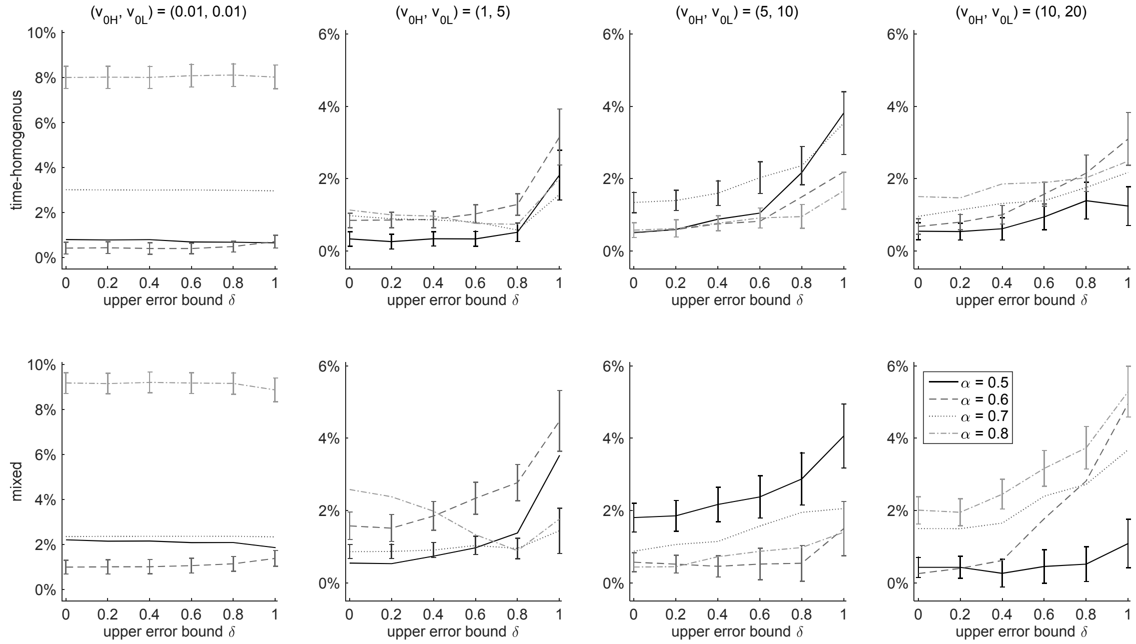

Figure 3: Revenue gains of *DPD-surr* over *DPD-ah* in network 1

5.3 Network 2: Small hub-and-spoke network

Again, we first describe the specific product and demand data of the second problem instance (Section 5.3.1) and interpret the corresponding surrogate reformulation (Section 5.3.2). We then turn to the computational results (Section 5.3.3).

5.3.1 Network description

Network 2 consists of seven flight legs connecting the four cities A, B, C, and H (see Figure 4). There are 11 itineraries, and on each itinerary, the airline offers a high fare class and a low fare class product. Details on prices and capacity consumption of these 22 regular products are identical to Liu and van Ryzin (2008) and provided in Table E.1 in Appendix E. In addition, we consider five flexible products (products $f1$–$f5$). The first one offers transportation from A to B, either on leg 1, legs 2 and 4, or legs 3 and 5. The second flexible product is from A to C on either (2, 6) or (3, 7). The last three flexible products guarantee short-haul transportation on one of the two possible legs from A to H, H to B, and H to C. The prices of these five products are given by $\boldsymbol{r}^{flex} = (240, 280, 160, 120, 200)^T$. We consider 15 customer segments: one high fare class and one low fare class segment interested in regular products for each of the origin-destination pairs AB, AH, HB, HC, and AC, and one segment for each of the five flexible products. Details on consideration sets, preference vectors, and segment probabilities can be found in Table E.2 in Appendix E. Different customer attitudes are again captured by the four values of the no-purchase preference vector already used in network 1. The flexible product segments' weights for the no-purchase alternative are fixed at 0.01. Analogously to network 1, we consider the two arrival patterns *time-homogenous* and *mixed* as well as a capacity factor.

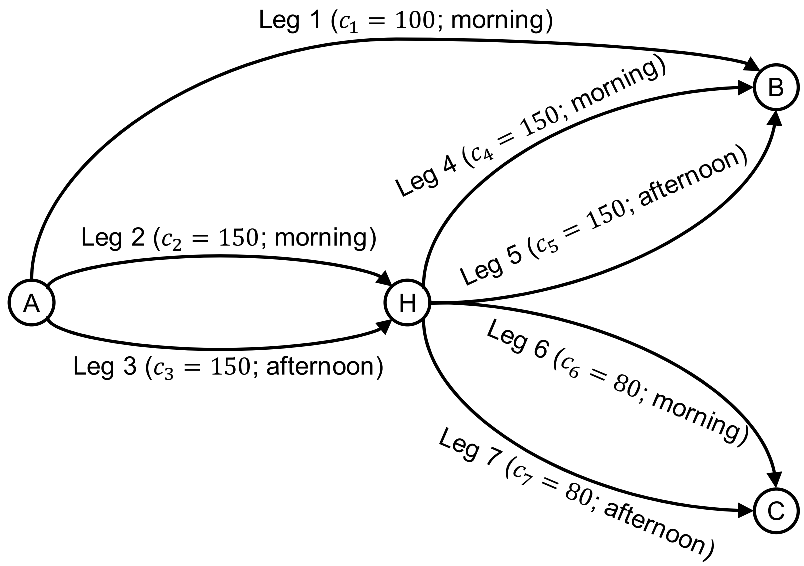

Figure 4: Small hub-and-spoke network (Network 2; see Liu and van Ryzin (2008))

5.3.2 Surrogate reformulation

The surrogate network comprises 11 artificial resources. The artificial resources and flexible products' capacity consumption are shown in Table 3.

Regular products require one unit of capacity of the corresponding regular resource(s) (see Table E.1) and one unit of capacity of each artificial resource containing the regular resource.

Again, the artificial resources are interpretable. For example, sales of $f4$ (H to B) are limited by the joint capacity of legs 4 and 5. This restriction is captured by the artificial resource $\tilde{c}_5 = c_4 + c_5$. Note that the restriction imposed by artificial resource $\tilde{c}_9 = c_1 + c_4 + c_5$ is obviously weaker and never limiting for $f4$ sales, but it is necessary to capture an interaction with $f1$, which will be described later.

Product $f1$ (A to B) is a bit more tedious. Similar to the running example from Section 4, the number of sales is restricted by the joint capacity of the legs out of A ($\tilde{c}_{11} = c_1 + c_2 + c_3$) and into B ($\tilde{c}_9 = c_1 + c_4 + c_5$). Furthermore, legs 2 and 5 ($\tilde{c}_1 = c_1 + c_2 + c_5$) or legs 3 and 4 ($\tilde{c}_2 = c_1 + c_3 + c_4$) may be the bottleneck. Together, these four artificial resources restrict sales of $f1$ to $c_1 + \min\{c_2 + c_4\} + \min\{c_3 + c_5\}$.

Besides the restrictions on sales for individual flexible products, the capacity shared by multiple flexible products has to be taken into consideration. Artificial resource 9 is a simple example: Capacity on legs 4 and 5 used by $f4$ customers cannot be used by $f1$ customers. Thus, this artificial resource – which was derived above as an individual restriction for $f1$ – is in fact not only used by $f1$ but also by $f4$. Furthermore, there can also be additional artificial resources that are only required because of such interactions between flexible products. Artificial resource 6 ($\tilde{c}_6 = c_1 + c_2 + c_5 + c_7$) is an example of this: It restricts joint sales of $f1$ and $f2$, because customers going from A to B ($f1$) or A to C ($f2$) either leave in the morning (legs 1 and 2) or arrive in the afternoon (legs 5 and 7). Note that this interaction between $f1$ and $f2$ is not captured by artificial resources 1, 2, 9, and 11 described above, because the routing of $f2$ may also be restricted by capacity on legs 6 and 7. An analogous example is $\tilde{c}_7 = c_1 + c_3 + c_4 + c_6$.

Artificial resource		**Flexible product**				
Index i	Capacity $\tilde{c}_i$	$f1$	$f2$	$f3$	$f4$	$f5$
1	$c_1 + c_2 + c_5$	X				
2	$c_1 + c_3 + c_4$	X				
3	$c_2 + c_7$		X			
4	$c_3 + c_6$		X			
5	$c_4 + c_5$				X	
6	$c_1 + c_2 + c_5 + c_7$	X	X			
7	$c_1 + c_3 + c_4 + c_6$	X	X			
8	$c_2 + c_3$		X	X		
9	$c_1 + c_4 + c_5$	X			X	
10	$c_6 + c_7$		X			X
11	$c_1 + c_2 + c_3$	X	X	X		

Table 3: Artificial resources of small hub-and-spoke network

5.3.3 Performance evaluation

Analogously to network 1, Figure 5 and Figure 6 show the average revenues of *DPD-surr*, *DPD-ah*, and *CDLP-surr*, as well as the revenue gain of *DPD-surr* over *DPD-ah* in network 2.

Compared with network 1, average revenues in the standard setting without forecast errors (Figure 5) are slightly higher in network 2 for both *DPD-ah* (97.5%–99%) and *DPD-surr* (99%–100%). Again, the performance of *DPD-surr* is more stable without any outliers, while *DPD-ah*'s revenue often falls below 98% of *UB* for capacity factors of $\alpha = 0.8$. By contrast, *CDLP-surr* performs considerably worse with many revenues around 93% and only a few values exceeding 96%.

For $(v_{0H}, v_{0L})^T = (0.01, 0.01)^T$, the revenue gain (Figure 6) of *DPD-surr* is considerably smaller at around 1%–2%. With the other three preference weights of $(v_{0H}, v_{0L})^T$ considered, the gain is more or less the same as in network 1. However, especially for $(v_{0H}, v_{0L})^T = (10,20)^T$ and time-homogeneous demand, considerably higher gains are observed.

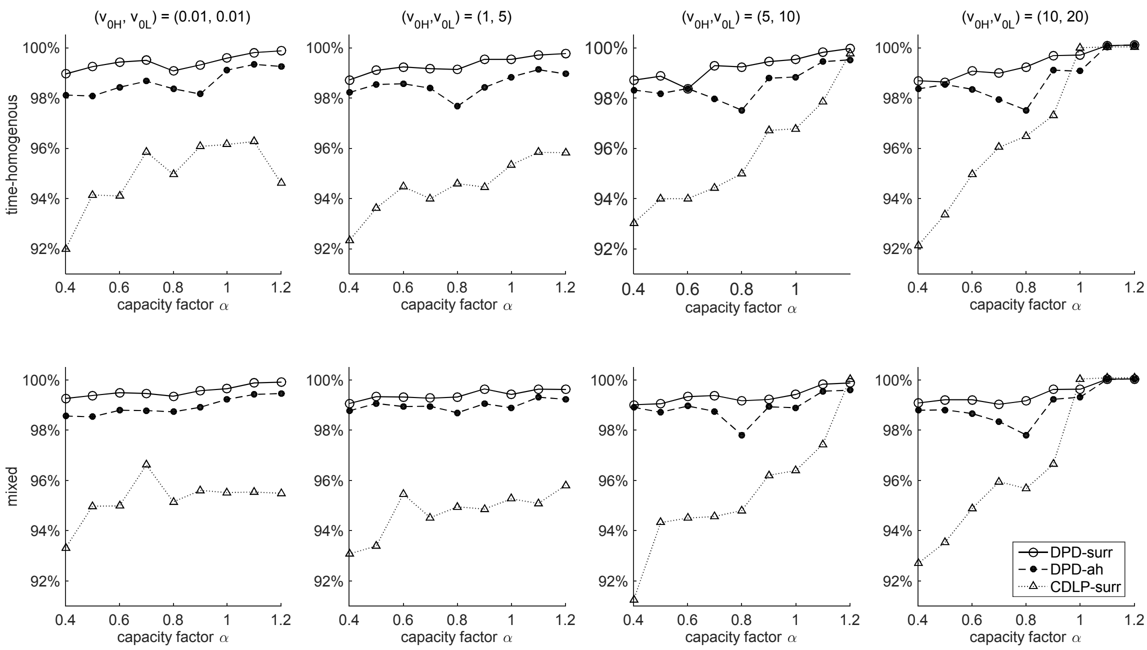

Figure 5: Average revenues of *DPD-surr*, *DPD-ah*, and *CDLP-surr* relative to *UB* in network 2

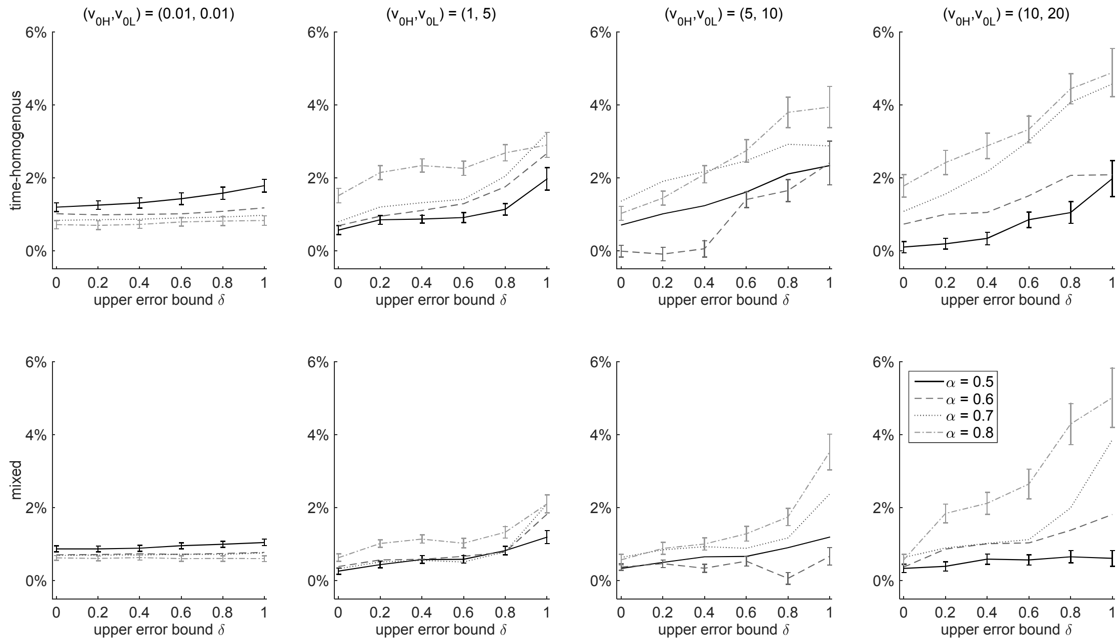

Figure 6: Revenue gains of *DPD-surr* over *DPD-ah* in network 2

6 Discussion and future research

Several managerial implications follow from our work. Most importantly, the inclusion of flexible products no longer excludes the use of standard dynamic programming techniques. We presented a novel generic way to overcome the commitment-based state space and the feasibility problem inherent in network revenue management problems regarding flexible products. In the surrogate approach, the problem is reformulated by applying FME to the feasibility problem, and an equivalent standard revenue management problem is obtained. This allows the direct use of standard DPD. Moreover, it allows the continued use of arbitrary methods and existing software systems, albeit with modified input data.

In a large number of numerical experiments, we compared the approach with a benchmark approach adapted from the literature that forgoes flexibility and obtains a resource-based state space by immediately assigning flexible products (ad hoc) after sale. The surrogate approach consistently obtains the highest revenues, which are close to the theoretical upper bound. In test instances with intermediate capacity, this approach increases revenues by up to 8% compared to the ad hoc approach. Moreover, revenue gains increase when forecast errors are considered. Thus, we think the surrogate approach should be the first choice when incorporating flexible products into revenue management.

Moreover, the difference between the revenue of the surrogate approach and that obtained with the ad hoc approach can also be roughly interpreted as the supply-side benefit of offering flexible products instead of opaque products. Flexible products should be offered if this benefit outweighs their demand-side disadvantages (customers usually prefer an opaque product where they are immediately informed of what they get). However, there is no clear advice here. The difference is marginal in some cases (extreme capacity situations, low forecast errors) and considerable in others (intermediate capacity, medium to high forecast errors).

We think that our results are promising, and we encourage future work on this topic. First, research could focus on problem instances where our transformed feasibility problem is heuristic. In this respect, a starting point might be projection for integer problems (see, e.g., Williams and Hooker (2016)). Second, our results indicate that heuristics restricting flexible products' flexibility can also yield a good revenue performance. Consequently, we think it is promising to develop approximate dynamic programming techniques tailored to flexible products that retain more flexibility than the ad hoc approach. For example, the linear programming approach for approximate dynamic programming (see, e.g., Adelman (2007) for the traditional revenue management setting; Tong and Topaloglu (2014), as well as Vossen and Zhang (2015a) for refinements) could be extended to flexible products.

Acknowledgements

We thank the Associate Editor, as well as two anonymous reviewers, for the valuable feedback that helped improve the paper.

References

Adelman D (2007) Dynamic bid prices in revenue management. *Operations Research* 55(4): 647-661.

Bartodziej P, Derigs U, Zils M (2006) O&D revenue management in cargo airlines - A mathematical programming approach. *OR Spectrum* 29(1): 105-121.

Belobaba PP (1987) Air travel demand and airline seat inventory management. Dissertation, Massachusetts Institute of Technology, Cambridge.

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

Bertsimas D, Popescu I (2003) Revenue management in a dynamic network environment. *Transportation Science* 37(3): 257-277.

Bertsimas D, Tsitsiklis JN (1997) *Introduction to linear optimization,* 3 (Athena Scientific, Belmont, Massachusetts).

Birbil Şİ, Frenk JBG, Gromicho JAS, Zhang S (2014) A network airline revenue management framework based on decomposition by origins and destinations. *Transportation Science* 48(3): 313-333.

Chen D (1998) Network flows in hotel yield management. Working paper, Cornell University, New York.

Chen S, Gallego G, Li MZ, Lin B (2010) Optimal seat allocation for two-flight problems with a flexible demand segment. *European Journal of Operational Research* 201(3): 897-908.

Cooper WL, Homem-de-Mello T (2007) Some decomposition methods for revenue management. *Transportation Science* 41(3): 332-353.

Davis JM, Gallego G, Topaloglu H (2014) Assortment optimization under variants of the nested logit model. *Operations Research* 62(2): 250-273.

Erdelyi A, Topaloglu H (2009) Separable approximations for joint capacity control and overbooking decisions in network revenue management. *Journal of Revenue and Pricing Management* 8(1): 3-20.

Erdelyi A, Topaloglu H (2010) A dynamic programming decomposition method for making overbooking decisions over an airline network. *INFORMS Journal on Computing* 22(3): 443-456.

Fink A, Reiners T (2006) Modeling and solving the short-term car rental logistics problem. *Transportation Research Part E: Logistics and Transportation Review* 42(4): 272-292.

Gallego G, Iyengar G, Phillips RL, Dubey A (2004) Managing flexible products on a network. Working paper, Columbia University, New York.

Gallego G, Phillips RL (2004) Revenue management of flexible products. *Manufacturing & Service Operations Management* 6(4): 321-337.

Gallego G, Ratliff R, Shebalov S (2015) A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research* 63(1): 212-232.

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

Geraghty MK, Johnson E (1997) Revenue management saves national car rental. *Interfaces* 27(1): 107-127.

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

Gönsch J, Steinhardt C (2013) Using dynamic programming decomposition for revenue management with opaque products. *BuR - Business Research* 6(1): 94-115.

Gönsch J, Steinhardt C (2015) On the incorporation of upgrades into airline network revenue management. *Review of Managerial Science* 9(4): 635-660.

Hosseinalifam M (2014) A mathematical programming framework for network capacity control in customer choice-based revenue management. Dissertation, École polytechnique de montréal, Montréal.

Jordan WC, Graves SC (1995) Principles on the benefits of manufacturing process flexibility. *Management Science* 41(4): 577-594.

Kemmer P, Strauss AK, Winter T (2011) Dynamic simultaneous fare proration for large-scale network revenue management. *Journal of the Operational Research Society* 63(10): 1336-1350.

Kimms A, Müller-Bungart M (2007) Revenue management for broadcasting commercials: the channel's problem of selecting and scheduling the advertisements to be aired. *International Journal of Revenue Management* 1(1): 28-44.

Kunnumkal S, Topaloglu H (2008) A tractable revenue management model for capacity allocation and overbooking over an airline network. *Flexible Services and Manufacturing Journal* 20(3): 125-147.

Kunnumkal S, Topaloglu H (2010) A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. *Production and Operations Management* 19(5): 575-590.

Lautenbacher CJ, Stidham S (1999) The underlying markov decision process in the single-leg airline yield-management problem. *Transportation Science* 33(2): 136-146.

Leachman RC, Carmon TF (1992) On capacity modeling for production planning with alternative machine types. *IIE Transactions* 24(4): 62-72.

Lee TC, Hersh M (1993) A model for dynamic airline seat inventory control with multiple seat bookings. *Transportation Science* 27(3): 252-265.

Littlewood K (1972) Forecasting and control of passengers. *Proceedings of the 12th AGIFORS symposium,* Nathanya, Israel): 95-117.

Liu Q, van Ryzin G (2008) On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management* 10(2): 288-310.

Martin RK (1999) *Large Scale Linear and Integer Optimization: A Unified Approach* (Kluwer, Boston).

Meissner J, Strauss A (2012) Network revenue management with inventory-sensitive bid prices and customer choice. *European Journal of Operational Research* 216(2): 459-468.

Meissner J, Strauss A, Talluri KT (2013) An enhanced concave program relaxation for choice network revenue management. *Production and Operations Management* 22(1): 71-87.

Miranda Bront JJ, Méndez-Díaz I, Vulcano G (2009) A column generation algorithm for choice-based network revenue management. *Operations Research* 57(3): 769-784.

Oosten M (2004) Revenue management with flexible due dates. Presentation, Conference of POMS College of Service Operations, New York.

Pachon JE, Iakovou E, Ip C, Aboudi R (2003) A synthesis of tactical fleet planning models for the car rental industry. *IIE Transactions* 35(9): 907-916.

Paulraj S, Sumathi P (2010) A comparative study of redundant constraints identification methods in linear programming problems. *Mathematical Problems in Engineering* 2010(1): 1-16.

Petrick A, Gönsch J, Steinhardt C, Klein R (2010) Dynamic control mechanisms for revenue management with flexible products. *Computers & Operations Research* 37(11): 2027-2039.

Petrick A, Steinhardt C, Gönsch J, Klein R (2012) Using flexible products to cope with demand uncertainty in revenue management. *OR Spectrum* 34(1): 215-242.

Schrijver A (1998) *Theory of Linear and Integer Programming* (Wiley, Chichester).

Shumsky RA, Zhang F (2009) Dynamic capacity management with substitution. *Operations Research* 57(3): 671-684.

Steinhardt C, Gönsch J (2012) Integrated revenue management approaches for capacity control with planned upgrades. *European Journal of Operational Research* 223(2): 380-391.

Strauss AK, Talluri KT (2015) Tractable consideration set structures for network revenue management. Working paper, University of Warwick, Warwick.

Talluri KT (2001) Airline revenue management with passenger routing control: A new model with solution approaches. *International Journal of Services Technology and Management* 2(1/2): 102-115.

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

Talluri KT, van Ryzin G (2004a) Revenue management under a general discrete choice model of consumer behavior. *Management Science* 50(1): 15-33.

Talluri KT, van Ryzin G (2004b) *The Theory and Practice of Revenue Management* (Springer, New York).

Tong C, Topaloglu H (2014) On the approximate linear programming approach for network revenue management problems. *INFORMS Journal on Computing* 26(1): 121-134.

Train K (2009) *Discrete Choice Methods with Simulation,* 2nd ed. (Cambridge University Press, Cambridge).

Vossen TWM, Zhang D (2015a) A dynamic disaggregation approach to approximate linear programs for network revenue management. *Production and Operations Management* 24(3): 469-487.

Vossen TWM, Zhang D (2015b) Reductions of approximate linear programs for network revenue management. *Operations Research* 63(6).

Walter M, Truemper K (2013) Implementation of a unimodularity test. *Mathematical Programming Computation* 5(1): 57-73.

Williams HP, Hooker JN (2016) Integer programming as projection. *Appears in: Discrete Optimization*.

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

Zhang D, Adelman D (2009) An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science* 43(3): 381-394.

Dynamic programming decomposition for choice-based revenue management with flexible products

Sebastian Koch, Jochen Gönsch, Claudius Steinhardt

Online Appendix

Appendix A: Proof of Proposition 1

Proposition 1: Given Condition 1 holds, $(\boldsymbol{c}, \boldsymbol{y}) \in \mathcal{Z}_{(\boldsymbol{A},\boldsymbol{\mathcal{M}})}$ (that is, the feasibility problem (2)–(4) has a solution) if and only if $\left(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y})\right) \geq \boldsymbol{0}$ (that is, (11)–(12) has a solution).

Proof: First, (2)–(4) is reformulated as (8) using Condition 1. This is transformed into (11)–(12) by projecting out one distribution variable after the other, using Algorithm 1. To show that the whole algorithm keeps equivalence, it is sufficient to show that the inequality system before an iteration implies the inequality system after the iteration and vice versa. In the following, we show this for an arbitrary iteration d. Please note that, due to the construction of the algorithm, d always refers to the first column of the current **LHS**, which contains at least one nonzero coefficient.

W.l.o.g., assume that the coefficients in the first column are 1, -1, or 0. The system before the iteration starts is given by

$$1 \cdot y_d + \sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \text{lhs}_{pos,col} \cdot y_{col} \leq \mathbf{RHS}_{pos} \text{ for all } pos \in Pos \tag{A.1}$$

$$(-1) \cdot y_d + \sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \text{lhs}_{neg,col} \cdot y_{col} \leq \mathbf{RHS}_{neg} \text{ for all } neg \in Neg \tag{A.2}$$

$$\sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \text{lhs}_{null,col} \cdot y_{col} \leq \mathbf{RHS}_{null} \text{ for all } null \in Null \tag{A.3}$$

and the system after the iteration by

$$\sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \text{lhs}_{null,col} \cdot y_{col} \leq \mathbf{RHS}_{null} \text{ for all } null \in Null \tag{A.4}$$

$$\sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \left(\text{lhs}_{pos,col} + \text{lhs}_{neg,col}\right) \cdot y_{col} \leq \mathbf{RHS}_{pos} + \mathbf{RHS}_{neg} \quad \text{for all } (pos, neg) \in Pos \times Neg. \tag{A.5}$$

Now, note that (A.1) and (A.2) imply

$$-\mathbf{RHS}_{neg} + \sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \mathrm{lhs}_{neg,col} \cdot y_{col} \leq y_d \leq \mathbf{RHS}_{pos} - \sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \mathrm{lhs}_{pos,col} \cdot y_{col}$$
$$\text{for all } (pos, neg) \in Pos \times Neg, \tag{A.6}$$

which itself implies

$$-\mathbf{RHS}_{neg} + \sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \mathrm{lhs}_{neg,col} \cdot y_{col} \leq \mathbf{RHS}_{pos} - \sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \mathrm{lhs}_{pos,col} \cdot y_{col}$$
$$\text{for all} (pos, neg) \in Pos \times Neg. \tag{A.7}$$

The proof that (A.1)–(A.3) implies (A.4)–(A.5) is particularly easy. Consider an arbitrary solution $\left(y_d, \ldots, y_{|\sum_k \mathcal{M}_k|}\right)^T$ from (A.1)–(A.3). Then, (A.4) has a solution, because it is equivalent to (A.3). Moreover, because (A.1) and (A.2) hold, (A.7) holds and, therefore, also (A.5), which is (A.7) slightly reformulated.

Now, in order to show that (A.4)–(A.5) implies (A.1)–(A.3), consider an arbitrary solution $\left(y_{d+1}, \ldots, y_{|\sum_k \mathcal{M}_k|}\right)^T$ from (A.4)–(A.5). (A.3) has a solution, because it is equivalent to (A.4). Moreover, because (A.5) and its reformulation (A.7) have a solution, we can construct a y_d for which $y_d \geq \max_{neg \in Neg}\left\{-\mathbf{RHS}_{neg} + \sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \mathrm{lhs}_{neg,col} \cdot y_{col}\right\}$ and $y_d \leq \min_{pos \in Pos}\left\{\mathbf{RHS}_{pos} - \sum_{col=d+1}^{|\sum_k \mathcal{M}_k|} \mathrm{lhs}_{pos,col} \cdot y_{col}\right\}$ holds. Therefore (A.2) and (A.1) hold, too. □

Appendix B: Equivalence of DP-flex and DP-surr

Appendix B.1: Proof of Proposition 2

Proposition 2: Let $(\boldsymbol{c}, \boldsymbol{y}) \in \mathcal{Z}_{(\boldsymbol{A},\boldsymbol{\mathcal{M}})}$ be an arbitrary state of DP-flex. Then, we have

(a) $\left(\boldsymbol{c} - \boldsymbol{a}_j, \boldsymbol{y}\right) \in \mathcal{Z}_{(\boldsymbol{A},\boldsymbol{\mathcal{M}})}$ if and only if $\left(\boldsymbol{c} - \boldsymbol{a}_j, \tilde{\boldsymbol{c}}\left(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c} - \boldsymbol{a}_j, \boldsymbol{y}\right)\right) \geq \boldsymbol{0}\ \forall j$ and

$(\boldsymbol{c}, \boldsymbol{y} + \boldsymbol{e}_k) \in \mathcal{Z}_{(\boldsymbol{A},\boldsymbol{\mathcal{M}})}$ if and only if $\left(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y} + \boldsymbol{e}_k)\right) \geq \boldsymbol{0}\ \forall k$

(b) $\left(\boldsymbol{c} - \boldsymbol{a}_j, \tilde{\boldsymbol{c}}\left(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c} - \boldsymbol{a}_j, \boldsymbol{y}\right)\right) = \left(\boldsymbol{c} - \boldsymbol{a}_j, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) - \tilde{\boldsymbol{a}}_j(\boldsymbol{A}, \boldsymbol{\mathcal{M}})\right)\ \forall j$ and

$\left(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y} + \boldsymbol{e}_k)\right) = \left(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) - \tilde{\boldsymbol{b}}_k(\boldsymbol{A}, \boldsymbol{\mathcal{M}})\right)\ \forall k$

Proof: Expression (a) obviously follows from Proposition 1, because we can simply define the state $\left(\boldsymbol{c} - \boldsymbol{a}_j, \boldsymbol{y}\right) \coloneqq (\boldsymbol{c}', \boldsymbol{y})$ (and $(\boldsymbol{c}, \boldsymbol{y} + \boldsymbol{e}_k) \coloneqq (\boldsymbol{c}, \boldsymbol{y}')$) and apply Proposition 1 to the so-defined state.

To show expression (b), first consider the upper case; that is, the sale of a regular product j. Regarding the first term in both brackets ($\boldsymbol{c} - \boldsymbol{a}_j$, i.e., the regular resources), the equality is trivial. Regarding the second term in the brackets (i.e., the artificial resources), we have:

$$\tilde{\boldsymbol{c}}\left(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c} - \boldsymbol{a}_j, \boldsymbol{y}\right) =$$

$$= \sum_h \tilde{\boldsymbol{f}}_h(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot \left(c_h - a_{hj}\right) - \sum_k \tilde{\boldsymbol{b}}_k(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot y_k =$$

$$= \sum_h \tilde{\boldsymbol{f}}_h(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot c_h - \sum_k \tilde{\boldsymbol{b}}_k(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot y_k - \sum_h \tilde{\boldsymbol{f}}_h(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot a_{hj} =$$

$$= \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) - \tilde{\boldsymbol{a}}_j(\boldsymbol{A}, \boldsymbol{\mathcal{M}})$$

The first equality simply follows from the definition of the function $\tilde{\boldsymbol{c}}(\cdot)$ with reduced capacity, the second is algebra, and the third uses the definitions of $\tilde{\boldsymbol{c}}(\cdot)$ and $\tilde{\boldsymbol{a}}_j(\cdot)$.

Next, consider the lower case of Proposition 2 (b); that is, the sale of a specific flexible product k (in the summations, flexible products are denoted as k' in the following). Similarly to the considerations above, we only have to consider the second term in brackets:

$$\tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y} + \boldsymbol{e}_k) =$$

$$= \sum_h \tilde{\boldsymbol{f}}_h(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot c_h - \sum_{k' \neq k} \tilde{\boldsymbol{b}}_{k'}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot y_{k'} - \tilde{\boldsymbol{b}}_k(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot (y_k + 1) =$$

$$= \sum_h \tilde{\boldsymbol{f}}_h(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot c_h - \sum_{k'} \tilde{\boldsymbol{b}}_{k'}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot y_{k'} - \tilde{\boldsymbol{b}}_k(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \cdot 1 =$$

$$= \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) - \tilde{\boldsymbol{b}}_k(\boldsymbol{A}, \boldsymbol{\mathcal{M}})$$

The first and the third equality follow from the definition of $\tilde{\boldsymbol{c}}(\cdot)$ with increased commitments and the second is algebra. □

Appendix B.2: Proof of Proposition 3

Proposition 3: $V_t(\boldsymbol{c}, \boldsymbol{y}) = V_t^{surr}\big(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y})\big)$ for all t, $(\boldsymbol{c}, \boldsymbol{y})$.

Proof: The equality is shown by induction over t. It holds for $t = 0$, because, from the boundary conditions, we have $V_0(\boldsymbol{c}, \boldsymbol{y}) = V_0^{surr}\big(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y})\big) = 0$ for $(\boldsymbol{c}, \boldsymbol{y}) \in \mathcal{Z}_{(\boldsymbol{A}, \boldsymbol{\mathcal{M}})} \Leftrightarrow \big(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y})\big) \geq \boldsymbol{0}$ and $V_0(\boldsymbol{c}, \boldsymbol{y}) = V_0\big(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y})\big) = -\infty$ otherwise, where the equivalence is Proposition 1.

Now, assume that the result holds for $t - 1$. In respect of t, two cases have to be distinguished again. If the boundary condition $(\boldsymbol{c}, \boldsymbol{y}) \notin \mathcal{Z}_{(\boldsymbol{A}, \boldsymbol{\mathcal{M}})} \Leftrightarrow \big(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y})\big) \ngeq \boldsymbol{0}$ applies, we have $V_t(\boldsymbol{c}, \boldsymbol{y}) = V_t^{surr}\big(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y})\big) = -\infty$. Otherwise, we have

$$\begin{aligned}
V_t(\boldsymbol{c}, \boldsymbol{y}) = \max_S \Big\{ & \textstyle\sum_j \lambda \cdot P_j^{reg}(S) \cdot \left(r_j^{reg} + V_{t-1}(\boldsymbol{c} - \boldsymbol{a}_j, \boldsymbol{y}) \right) \\
& + \textstyle\sum_k \lambda \cdot P_k^{flex}(S) \cdot \left(r_k^{flex} + V_{t-1}(\boldsymbol{c}, \boldsymbol{y} + \boldsymbol{e}_k) \right) \\
& + (\lambda \cdot P_0(S) + 1 - \lambda) \cdot V_{t-1}(\boldsymbol{c}, \boldsymbol{y}) \} \\
= \max_S \Big\{ & \textstyle\sum_j \lambda \cdot P_j^{reg}(S) \cdot \left(r_j^{reg} + V_{t-1}^{surr}\left(\boldsymbol{c} - \boldsymbol{a}_j, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c} - \boldsymbol{a}_j, \boldsymbol{y}) \right) \right) \\
& + \textstyle\sum_k \lambda \cdot P_k^{flex}(S) \cdot \left(r_k^{flex} + V_{t-1}^{surr}\left(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y} + \boldsymbol{e}_k) \right) \right) \\
& + (\lambda \cdot P_0(S) + 1 - \lambda) \cdot V_{t-1}^{surr}\left(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) \right) \Big\} \\
= \max_S \Big\{ & \textstyle\sum_j \lambda \cdot P_j^{reg}(S) \cdot \left(r_j^{reg} + V_{t-1}^{surr}\left(\boldsymbol{c} - \boldsymbol{a}_j, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) - \tilde{\boldsymbol{a}}_j(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \right) \right) \\
& + \textstyle\sum_k \lambda \cdot P_k^{flex}(S) \cdot \left(r_k^{flex} + V_{t-1}^{surr}\left(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) - \tilde{\boldsymbol{b}}_k(\boldsymbol{A}, \boldsymbol{\mathcal{M}}) \right) \right) \\
& + (\lambda \cdot P_0(S) + 1 - \lambda) \cdot V_{t-1}^{surr}\left(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) \right) \Big\} \\
= V_t^{surr} & \left(\boldsymbol{c}, \tilde{\boldsymbol{c}}(\boldsymbol{A}, \boldsymbol{\mathcal{M}}, \boldsymbol{c}, \boldsymbol{y}) \right)
\end{aligned}$$

The first equality is simply the definition of DP-flex (1), the second uses the induction hypothesis, the third equality follow from Proposition 2 (b), and the fourth is the definition of DP-surr (13). □

Appendix C: Derivation of artificial resources for network types 1 and 2

Appendix C.1: Proof of Proposition 4

Proposition 4: In network type 1, the number of artificial resources is $\widetilde{m} = 1$ (and thus constant in the number of regular resources m).

Proof: Network type 1 consists of m parallel resources and one flexible product that may be assigned to the m resources. Thus, the feasibility problem (5)–(7) is given by

$$y_{1h} \leq c_h \ \forall h = 1, \dots, m \tag{C.1.1}$$

$$\sum_{h=1}^{m} -y_{1h} \leq -y_1 \tag{C.1.2}$$

$$-y_{1h} \leq 0 \ \forall h = 1, \dots, m \tag{C.1.3}$$

Throughout the proof, we refer to the FME-steps as given by Algorithm 1. W.l.o.g., we assume that the elimination of y_{1h} by Algorithm 1 is done iteratively in increasing order of h. We index the iterations of Algorithm 1 with $i = 1, \dots, m + 1$, each referring to the feasibility problem before eliminating the distribution variable y_{1i} (and after eliminating $y_{1,i-1}$). Please note that iteration $i = 1$ corresponds to the initial feasibility problem and that the dummy iteration $i = m + 1$ gives us the feasibility problem after eliminating all the distribution variables.

Now, by induction over i, we show the following:

Induction hypothesis: The feasibility problem in iteration i is given by

$$y_{1h} \leq c_h \ \forall h = i, \dots, m \tag{C.1.4}$$

$$-y_{1h} \leq 0 \ \forall h = i, \dots, m \tag{C.1.5}$$

$$0 \leq c_h \ \forall h = 1, \dots, i - 1 \tag{C.1.6}$$

$$-\sum_{h=i}^{m} y_{1h} \leq \sum_{h=1}^{i-1} c_h - y_1 \tag{C.1.7}$$

Induction basis: The induction hypothesis holds for $i = 1$, because (C.1.4), (C.1.7), and (C.1.5) equal (C.1.1), (C.1.2), and (C.1.3), respectively, and because (C.1.6) drops out ($\forall h = 1, \dots, 0$).

Induction step: Assume that the hypothesis holds for i. We next show that it will then also hold for $i+1$. The feasibility problem of iteration $i+1$ is obtained by applying FME on y_{1i} in (C.1.4)–(C.1.7). The constraints/rows with null coefficients for y_{1i} stay the same according to lines 9 and 10 of Algorithm 1:

$y_{1h} \leq c_h \; \forall h = i+1, \dots, m$ (second constraint to last constraint of (C.1.4)) (C.1.8)

$-y_{1h} \leq 0 \; \forall h = i+1, \dots, m$ (second constraint to last constraint of (C.1.5)) (C.1.9)

$0 \leq c_h \; \forall h = 1, \dots, i-1$ (constraints (C.1.6))

The constraints with coefficient $+1$ or -1 for y_{1i} are given by

$y_{1i} \leq c_i$ (first constraint of (C.1.4)) (C.1.10)

$-y_{1i} \leq 0$ (first constraint of (C.1.5)) (C.1.11)

$-\sum_{h=i}^{m} y_{1h} \leq \sum_{h=1}^{i-1} c_h - y_1$ (constraint (C.1.7))

Consequently, adding (C.1.10) and (C.1.11), as well as (C.1.10) and (C.1.7), according to lines 13 and 14 of Algorithm 1 leads to

$0 \leq c_i$ (C.1.12)

$-\sum_{h=i+1}^{m} y_{1h} \leq c_i + \sum_{h=1}^{i-1} c_h - y_1 = \sum_{h=1}^{i} c_h - y_1$ (C.1.13)

Thus, in total, we obtain the following set of constraints from iteration i:

$y_{1h} \leq c_h \; \forall h = i+1, \dots, m$ (constraints (C.1.8))

$-y_{1h} \leq 0 \; \forall h = i+1, \dots, m$ (constraints (C.1.9))

$0 \leq c_h \; \forall h = 1, \dots, i$ (consisting of constraints (C.1.6) and (C.1.12))

$-\sum_{h=i+1}^{m} y_{1h} \leq \sum_{h=1}^{i} c_h - y_1$ (constraint (C.1.13)

These constraints equal (C.1.4)–(C.1.7) with increased $i \coloneqq i+1$, which concludes the induction step.

Given this result, we subsequently consider the feasibility problem after completely executing Algorithm 1, that is, after eliminating all the distribution variables. The resulting problem is given by (C.1.6) and (C.1.7) with $i = m+1$, because (C.1.4) and (C.1.5) drop out ($\forall h = m+1, \dots, m$):

$0 \leq c_h \ \forall h = 1, \dots, m$ (regular resources, constraints (C.1.6))

$0 \leq \sum_{h=1}^{m} c_h - y_1$ (artificial resource, constraint (C.1.7))

Therefore, we obtain a total of $\widetilde{m} = 1$ artificial resource. ◻

Appendix C.2: Proof of Proposition 5

Proposition 5: In network type 2, the number of artificial resources is $\widetilde{m} = \frac{(m-1) \cdot m}{2}$ (and thus polynomial in the number of regular resources m).

Proof: Network type 2 consists of m parallel resources and $m - 1$ flexible products. Flexible product k may be assigned to resource k or $k + 1$. Thus, the feasibility problem (5)–(7) is given by

$$y_{11} \leq c_1 \tag{C.2.1}$$

$$y_{k2} + y_{k+1,1} \leq c_{k+1} \ \forall k = 1, \dots, m-2 \tag{C.2.2}$$

$$y_{m-1,2} \leq c_m \tag{C.2.3}$$

$$-y_{k1} - y_{k2} \leq -y_k \ \forall k = 1, \dots, m-1 \tag{C.2.4}$$

$$-y_{k1} \leq 0 \ \forall k = 1, \dots, m-1 \tag{C.2.5}$$

$$-y_{k2} \leq 0 \ \forall k = 1, \dots, m-1 \tag{C.2.6}$$

W.l.o.g., we assume that the $2 \cdot (m-1)$ distribution variables are eliminated in the order of $y_{11}, y_{12}, y_{21}, \dots, y_{m-1,2}$. For this purpose, we conduct the iterations $i = 1, \dots, m-1$, with i referring to the feasibility problem after eliminating $y_{i-1,2}$ (for $i > 1$) and before eliminating y_{i1}. Please note that dummy iteration $i = 1$ corresponds to the initial feasibility problem, that each iteration comprises eliminating both distribution variables of flexible product i (and thus, two of the iterations of Algorithm 1), and that $i = m$ refers to the feasibility problem after eliminating all the distribution variables, that is, after iteration $m - 1$.

Now, by induction over i, we show the following:

Induction hypothesis: The feasibility problem in iteration $i = 1, \dots, m-1$ is given by

$$y_{i1} \le c_i \tag{C.2.7}$$

$$y_{k2} + y_{k+1,1} \le c_{k+1} \; \forall k = i, \dots, m-2 \tag{C.2.8}$$

$$y_{m-1,2} \le c_m \tag{C.2.9}$$

$$-y_{k1} - y_{k2} \le -y_k \; \forall k = i, \dots, m-1 \tag{C.2.10}$$

$$-y_{k1} \le 0 \; \forall k = i, \dots, m-1 \tag{C.2.11}$$

$$-y_{k2} \le 0 \; \forall k = i, \dots, m-1 \tag{C.2.12}$$

$$0 \le \textstyle\sum_{h=\underline{h}}^{\overline{h}} c_h - \sum_{h=\underline{h}}^{\overline{h}-1} y_h \;\; \forall \underline{h} = 1, \dots, i-1, \forall \overline{h} = \underline{h}, \dots, i-1 \tag{C.2.13}$$

$$y_{i1} \le \textstyle\sum_{h=\underline{h}}^{i} c_h - \sum_{h=\underline{h}}^{i-1} y_h \;\; \forall \underline{h} = 1, \dots, i-1 \tag{C.2.14}$$

Induction basis: The induction hypothesis holds for $i = 1$, because (C.2.7)–(C.2.12) equal (C.2.1)–(C.2.6), respectively, and because (C.2.13) and (C.2.14) drop out ($\forall \underline{h} = 1, \dots, 0$).

Induction step: Assume that the hypothesis holds for i. We now show that it will then also hold for $i + 1$. We first eliminate y_{i1} by performing one iteration of Algorithm 1. The constraints/rows with null coefficients for y_{i1} stay the same according to lines 9 and 10 of Algorithm 1:

$y_{k2} + y_{k+1,1} \le c_{k+1} \; \forall k = i, \dots, m-2$ (constraints (C.2.8))
$y_{m-1,2} \le c_m$ (constraint (C.2.9))
$-y_{k1} - y_{k2} \le -y_k \; \forall k = i+1, \dots, m-1$ (second constraint to last constraint of (C.2.10)) (C.2.15)

$-y_{k1} \le 0 \; \forall k = i+1, \dots, m-1$ (second constraint to last constraint of (C.2.11)) (C.2.16)

$-y_{k2} \le 0 \; \forall k = i, \dots, m-1$ (constraints (C.2.12))
$0 \le \sum_{h=\underline{h}}^{\overline{h}} c_h - \sum_{h=\underline{h}}^{\overline{h}-1} y_h \;\; \forall \underline{h} = 1, \dots, i-1, \forall \overline{h} = \underline{h}, \dots, i-1$ (constraints (C.2.13))

The constraints with coefficient $+1$ or -1 for y_{i1} are given by

$y_{i1} \le c_i$ (constraint (C.2.7))

$y_{i1} \le \sum_{h=\underline{h}}^{i} c_h - \sum_{h=\underline{h}}^{i-1} y_h \;\; \forall \underline{h} = 1, \dots, i-1$ (constraints (C.2.14))

$-y_{i1} - y_{i2} \leq -y_i$ (first constraint of (C.2.10)) (C.2.17)

$-y_{i1} \leq 0$ (first constraint of (C.2.11)) (C.2.18)

Consequently, adding (C.2.7) and (C.2.17), (C.2.7) and (C.2.18), (C.2.14) and (C.2.17), as well as (C.2.14) and (C.2.18), according to lines 13 and 14 of Algorithm 1 leads to

$-y_{i2} \leq c_i - y_i$ (C.2.19)

$0 \leq c_i$ (C.2.20)

$-y_{i2} \leq \sum_{h=\underline{h}}^{i} c_h - \sum_{h=\underline{h}}^{i-1} y_h - y_i = \sum_{h=\underline{h}}^{i} c_h - \sum_{h=\underline{h}}^{i} y_h \;\; \forall \underline{h} = 1, \dots, i-1$ (C.2.21)

$0 \leq \sum_{h=\underline{h}}^{i} c_h - \sum_{h=\underline{h}}^{i-1} y_h \;\; \forall \underline{h} = 1, \dots, i-1$ (C.2.22)

Thus, in total, we obtain the following set of constraints after the elimination of y_{i1}:

$y_{k2} + y_{k+1,1} \leq c_{k+1} \; \forall k = i, \dots, m-2$ (constraints (C.2.8))

$y_{m-1,2} \leq c_m$ (constraint (C.2.9))

$-y_{k1} - y_{k2} \leq -y_k \; \forall k = i+1, \dots, m-1$ (constraints (C.2.15))

$-y_{k1} \leq 0 \; \forall k = i+1, \dots, m-1$ (constraints (C.2.16))

$-y_{k2} \leq 0 \; \forall k = i, \dots, m-1$ (constraints (C.2.12))

$0 \leq \sum_{h=\underline{h}}^{\overline{h}} c_h - \sum_{h=\underline{h}}^{\overline{h}-1} y_h \;\; \forall \underline{h} = 1, \dots, i, \forall \overline{h} = \underline{h}, \dots, i$ (consisting of constraints (C.2.13), (C.2.20), and (C.2.22)) (C.2.23)

$-y_{i2} \leq \sum_{h=\underline{h}}^{i} c_h - \sum_{h=\underline{h}}^{i} y_h \;\; \forall \underline{h} = 1, \dots, i$ (consisting of constraints (C.2.19) and (C.2.21)) (C.2.24)

For this set of constraints, we show that, when applying another iteration of Algorithm 1 to eliminate y_{i2}, we obtain the feasibility problem of iteration $i+1$; that is (C.2.7)–(C.2.14) with increased $i \coloneqq i+1$. The constraints/rows with null coefficients for y_{i2} stay the same according to lines 9 and 10 of Algorithm 1:

$y_{k2} + y_{k+1,1} \leq c_{k+1} \; \forall k = i+1, \dots, m-2$ (second constraint to last constraint of (C.2.8); drops out in case that $i = m-2$) (C.2.25)

$y_{m-1,2} \leq c_m$ (constraint (C.2.9); note that $i < m-1$ according to the hypothesis; thus this constraint always has null coefficients for y_{i2})

$-y_{k1} - y_{k2} \leq -y_k \; \forall k = i+1, \dots, m-1$ (constraints (C.2.15))

$-y_{k1} \leq 0 \; \forall k = i+1, \dots, m-1$ (constraints (C.2.16))

$-y_{k2} \leq 0 \; \forall k = i+1, \dots, m-1$ (second constraint to last constraint of (C.2.12)) (C.2.26)

$0 \leq \sum_{h=\underline{h}}^{\overline{h}} c_h - \sum_{h=\underline{h}}^{\overline{h}-1} y_h \;\; \forall \underline{h} = 1, \dots, i, \forall \overline{h} = \underline{h}, \dots, i$ (constraints (C.2.23))

The rows with coefficient $+1$ or -1 are

$y_{i2} + y_{i+1,1} \leq c_{i+1}$ (first constraint of (C.2.8); note that $i < m-1$ according to the hypothesis, thus this constraint always exists) (C.2.27)

$-y_{i2} \leq 0$ (first constraint of (C.2.12)) (C.2.28)

$-y_{i2} \leq \sum_{h=\underline{h}}^{i} c_h - \sum_{h=\underline{h}}^{i} y_h \;\; \forall \underline{h} = 1, \dots, i$ (constraints (C.2.24))

Consequently, adding (C.2.27) and (C.2.28), as well as (C.2.27) and (C.2.24), according to lines 13 and 14 of Algorithm 1 leads to

$y_{i+1,1} \leq c_{i+1}$ (C.2.29)

$y_{i+1,1} \leq c_{i+1} + \sum_{h=\underline{h}}^{i} c_h - \sum_{h=\underline{h}}^{i} y_h = \sum_{h=\underline{h}}^{i+1} c_h - \sum_{h=\underline{h}}^{i} y_h \;\; \forall \underline{h} = 1, \dots, i$ (C.2.30)

Thus, in total, we obtain the following set of constraints after the elimination of y_{i2}:

$y_{i+1,1} \leq c_{i+1}$ (constraint (C.2.29))

$y_{k2} + y_{k+1,1} \leq c_{k+1} \; \forall k = i+1, \dots, m-2$ (constraints (C.2.25))

$y_{m-1,2} \leq c_m$ (constraint (C.2.9))

$-y_{k1} - y_{k2} \leq -y_k \; \forall k = i+1, \dots, m-1$ (constraints (C.2.15))

$-y_{k1} \leq 0 \; \forall k = i+1, \dots, m-1$ (constraints (C.2.16))

$-y_{k2} \leq 0 \; \forall k = i+1, \dots, m-1$ (constraints (C.2.26))

$0 \leq \sum_{h=\underline{h}}^{\overline{h}} c_h - \sum_{h=\underline{h}}^{\overline{h}-1} y_h \;\; \forall \underline{h} = 1, \dots, i, \forall \overline{h} = \underline{h}, \dots, i$ (constraints (C.2.23))

$y_{i+1,1} \leq \sum_{h=\underline{h}}^{i+1} c_h - \sum_{h=\underline{h}}^{i} y_h \;\; \forall \underline{h} = 1, \dots, i$ (constraints (C.2.30))

These constraints equal (C.2.7)–(C.2.14) with increased $i \coloneqq i+1$, which concludes the induction step. Given this result, we can now formally state the feasibility problem after performing $m-2$ iterations by simply setting $i = m-1$ in the hypothesis. We obtain

$y_{m-1,1} \leq c_{m-1}$

$y_{m-1,2} \leq c_m$

$-y_{m-1,1} - y_{m-1,2} \leq -y_{m-1}$

$-y_{m-1,1} \leq 0$

$-y_{m-1,2} \leq 0$

$0 \leq \sum_{h=\underline{h}}^{\overline{h}} c_h - \sum_{h=\underline{h}}^{\overline{h}-1} y_h \ \ \forall \underline{h} = 1, \dots, m-2, \forall \overline{h} = \underline{h}, \dots, m-2$

$y_{m-1,1} \leq \sum_{h=\underline{h}}^{m-1} c_h - \sum_{h=\underline{h}}^{m-2} y_h \ \ \forall \underline{h} = 1, \dots, m-2$

Finally, we perform the remaining $m-1$-th iteration, that is, two final iterations of Algorithm 1 to subsequently eliminate $y_{m-1,1}$ and $y_{m-1,2}$. After the elimination of $y_{m-1,1}$, we obtain the following set of constraints:

$$y_{m-1,2} \leq c_m$$

$$-y_{m-1,2} \leq 0$$

$$0 \leq \sum_{h=\underline{h}}^{\overline{h}} c_h - \sum_{h=\underline{h}}^{\overline{h}-1} y_h \ \ \forall \underline{h} = 1, \dots, m-1 \ \forall \overline{h} = \underline{h}, \dots, m-1$$

$$-y_{m-1,2} \leq \sum_{h=\underline{h}}^{m-1} c_h - \sum_{h=\underline{h}}^{m-1} y_h \ \ \forall \underline{h} = 1, \dots, m-1$$

After eliminating $y_{m-1,2}$ — that is, after eliminating all the distribution variables of the original feasibility problem — we obtain the constraints

$$0 \leq \sum_{h=\underline{h}}^{\overline{h}} c_h - \sum_{h=\underline{h}}^{\overline{h}-1} y_h \ \ \forall \underline{h} = 1, \dots, m \ \forall \overline{h} = \underline{h}, \dots, m$$

which may be rewritten as

$0 \leq c_h \ \forall h = 1, \dots, m$ (regular resources)

$0 \leq \sum_{h=\underline{h}}^{\overline{h}} c_h - \sum_{h=\underline{h}}^{\overline{h}-1} y_h \ \ \forall \underline{h} = 1, \dots, m-1, \forall \overline{h} = \underline{h} + 1, \dots, m$ (artificial resources)

Therefore, we have a total of $\widetilde{m} = \sum_{h=1}^{m-1} h = \frac{(m-1)m}{2}$ artificial resources. □

Appendix D: Technical details of the implemented methods

In Section 5, we evaluated the average revenues obtained by using our surrogate approach (*DPD-surr*) in comparison with two revenue management methods (*DPD-ah* and *CDLP-surr*), as well as an upper bound (*UB*) on the optimal expected revenue of DP-flex (1). In the following, we provide the technical details.

Appendix D.1: Upper bound (*UB*)

As the upper bound on the optimal expected revenue of (1), we use the optimal objective value of the corresponding CDLP formulation, which Gallego et al. (2004) propose (**CDLP-flex**):

$$\text{Maximize } \sum_S t(S) \cdot \lambda \cdot \left(\sum_j P_j^{reg}(S) \cdot r_j^{reg} + \sum_k P_k^{flex}(S) \cdot r_k^{flex}\right) \tag{D.1.1}$$

subject to

$$\sum_S t(S) \cdot \lambda \cdot \sum_j P_j^{reg}(S) \cdot a_{hj} + \sum_k \sum_{j \in \mathcal{M}_k} a_{hj} \cdot y_{kj} \le c_h \ \forall h \tag{D.1.2}$$

$$\sum_S t(S) \cdot \lambda \cdot P_k^{flex} = \sum_{j \in \mathcal{M}_k} y_{kj} \ \forall k \tag{D.1.3}$$

$$\sum_S t(S) = T \tag{D.1.4}$$

$$t(S) \ge 0 \ \forall S \tag{D.1.5}$$

$$y_{kj} \ge 0 \ \forall k, j \in \mathcal{M}_k \tag{D.1.6}$$

In this model, the variable $t(S)$ denotes how long set S is offered. Please note that constraints (D.1.2), (D.1.3), and (D.1.6), after some minor rearrangements, equal the (relaxed) feasibility problem (5)–(7).

In order to solve CDLP-flex and obtain UB, we use column generation (see, e.g., Liu and van Ryzin (2008) and Miranda Bront et al. (2009) for an extensive description in the context of standard revenue management). We start with a reduced number of columns in CDLP-flex; that is, with only a subset of the possible offer sets. Let π_h^{red}, μ_k^{red}, and σ^{red} denote the optimal dual prices for restrictions (D.1.2), (D.1.3), and (D.1.4) of this reduced problem. Thereafter, we have to check whether there is an offer set with positive reduced costs that must be included. More precisely, a column corresponding to a new offer set is the optimal solution of the following column generation sub-problem:

$$\max_S\{\lambda \cdot \left(\sum_j P_j^{reg}(S) \cdot \left(r_j^{reg} - \sum_h a_{hj} \cdot \pi_h^{red}\right)\right.$$
$$\left.+ \sum_k P_k^{flex}(S) \cdot \left(r_k^{flex} - \mu_k^{red}\right)\right)\} - \sigma^{red} \quad \text{(D.1.7)}$$

The second term in the argument of the maximum function is due to the consideration of flexible products. The solution technique depends on the choice model used.

Please note that we can obtain *UB* alternatively by using the surrogate reformulation from Section 4. To see this, we apply the surrogate network on the standard CDLP formulation without flexible products (see, e.g., Liu and van Ryzin (2008)) and obtain the following formulation (**CDLP-surr**):

$$\text{Maximize } \sum_S t(S) \cdot \lambda \cdot \left(\sum_j P_j^{reg}(S) \cdot r_j^{reg} + \sum_{k \in S} P_k^{flex}(S) \cdot r_k^{flex}\right) \quad \text{(D.1.8)}$$

subject to

$$\sum_S t(S) \cdot \lambda \cdot \sum_j P_j^{reg}(S) \cdot a_{hj} \leq c_h \ \forall h \quad \text{(D.1.9)}$$

$$\sum_S t(S) \cdot \lambda \cdot \left(\sum_j P_j^{reg}(S) \cdot \tilde{a}_{ij} + \sum_k P_k^{flex}(S) \cdot \tilde{b}_{ik}\right) \leq \tilde{c}_i \ \forall i \quad \text{(D.1.10)}$$

$$\sum_S t(S) = T \quad \text{(D.1.11)}$$

$$t(S) \geq 0 \ \forall S \quad \text{(D.1.12)}$$

A comparison of CDLP-flex with CDLP-surr shows that they only differ in the constraints representing the feasibility problem: While CDLP-flex contains the original feasibility problem, CDLP-surr contains the transformed feasibility problem (constraints (D.1.9) and (D.1.10)). Thus, both CDLPs are equivalent. This result is intuitive, because both the original and the surrogate networks represent the same stochastic problem (represented by the DPs). In this sense, this result for the deterministic equivalent (given by the CDLP) of the stochastic problem parallels the result obtained in Section 4.2 in respect of the DPs. Please note that Condition 1 is not required for equivalence here.

Appendix D.2: Ad hoc assignment DPD approach (*DPD-ah*)

This method is based on an idea that was already investigated by Steinhardt and Gönsch (2012) in respect of the special case of upgrades and without customer choice. It forgoes flexibility and immediately assigns flexible products (ad hoc) after sale. As this assignment is irrevocable, we can immediately reduce the remaining capacity and do not need to store any commitments. Thus, a resource-based state space is obtained, and DPD by resources is possible. However, existing choice-based approaches do not include the described ad hoc assignment of flexible products and have to be modified appropriately. In the following, we carry out these modifications on the DPD approach of Liu and van Ryzin (2008).

The immediate assignment of a flexible product k to the current best of its alternatives $j \in \mathcal{M}_k$ is captured by the second line in the Bellman equation

$$\begin{aligned} V_t^{ah}(\boldsymbol{c}) &= \max_S \Big\{\textstyle\sum_j \lambda \cdot P_j^{reg}(S) \cdot \Big(r_j^{reg} + V_{t-1}^{ah}(\boldsymbol{c} - \boldsymbol{a}_j)\Big) \\ &+ \textstyle\sum_k \lambda \cdot P_k^{flex}(S) \cdot \max_{j\in\mathcal{M}_k}\{r_k^{flex} + V_{t-1}^{ah}(\boldsymbol{c} - \boldsymbol{a}_j)\} \\ &+ (\lambda \cdot P_0(S) + 1 - \lambda) \cdot V_{t-1}^{ah}(\boldsymbol{c})\} \end{aligned} \quad \text{(D.2.1)}$$

with the boundary conditions $V_t^{ah}(\boldsymbol{c}) = -\infty$ if $\boldsymbol{c} \ngeq \boldsymbol{0}$ and $V_0^{ah}(\boldsymbol{c}) = 0\ \forall \boldsymbol{c} \geq \boldsymbol{0}$.

The standard starting point of the decomposition is the corresponding CDLP formulation CDLP-flex (D.1.1)–(D.1.6). Let π_h^{ah} denote the optimal dual prices associated with the capacity of resource h (constraint (D.1.2)). We then obtain the following one-dimensional problem to assess the value of capacity of each resource $h' \in \mathcal{H}$:

$$\begin{aligned} V_t^{ah,h'}(c_{h'}) &= \max_S \Big\{\textstyle\sum_j \lambda \cdot P_j^{reg}(S) \cdot \Big(r_j^{reg} - \sum_{h\neq h'} a_{hj} \cdot \pi_h^{ah} + V_{t-1}^{ah,h'}(c_{h'} - a_{h'j})\Big) \\ &+ \textstyle\sum_k \lambda \cdot P_k^{flex}(S) \cdot \max_{j\in\mathcal{M}_k}\Big\{r_k^{flex} - \sum_{h\neq h'} a_{hj} \cdot \pi_h^{ah} + V_{t-1}^{ah,h'}(c_{h'} - a_{h'j})\Big\} \\ &+ (\lambda \cdot P_0(S) + 1 - \lambda) \cdot V_{t-1}^{ah,h'}(c_{h'})\Big\} \end{aligned} \quad \text{(D.2.2)}$$

with boundary conditions $V_t^{ah,h'}(c_{h'}) = -\infty$ if $c_{h'} < 0$ and $V_0^{ah,h'}(c_{h'}) = 0\ \forall c_{h'} \geq 0$.

During the booking horizon (that is, during the simulations), we approximate the opportunity cost of a regular product j and all flexible products' alternatives $j \in \mathcal{M}_k$, respectively, as the sum of the required resources' opportunity cost. More formally, with resource h's opportunity cost defined as

$$\Delta_h V_t^{ah,h}(c_h) := V_t^{ah,h}(c_h) - V_t^{ah,h}(c_h - 1), \quad \text{(D.2.3)}$$

the offer set is the optimal solution of

$$\begin{aligned} &\max_S\{\textstyle\sum_j \lambda \cdot P_j^{reg}(S) \cdot \big(r_j^{reg} - \sum_h a_{hj} \cdot \Delta_h V_{t-1}^{ah,h}(c_h)\big) \\ &+ \textstyle\sum_k \lambda \cdot P_k^{flex}(S) \cdot \max_{j\in\mathcal{M}_k}\{r_k^{flex} - \sum_h a_{hj} \cdot \Delta_h V_{t-1}^{ah,h}(c_h)\}\} \end{aligned} \quad \text{(D.2.4)}$$

Note that compared to the standard setting described, for example, in Liu and van Ryzin (2008), the second lines in equations (D.2.2) and (D.2.4) are extensions that are due to the consideration of flexible products. These modifications follow the lines of the modifications performed without customer choice in Steinhardt and Gönsch (2012) in respect of upgrades, and in Gönsch and Steinhardt (2013) in respect of opaque products. Similar to the column generation sub-problems used to solve CDLP-flex and CDLP-surr, the solution technique applied to find the optimal offer set S in (D.2.2) and (D.2.4) depends on the choice model used.

Appendix D.3: Primal solution of CDLP-surr (*CDLP-surr*)

This method operationalizes the optimal primal solution of CDLP-surr (D.1.8)–(D.1.12). Recall that the optimal solution gives us the time a set S is offered. Alternatively, the same solution is obtained by CDLP-flex (D.1.1)–(D.1.6).

We round fractional values of the decision variables $t(S)$ to the nearest integer. The sequence in which we offer the sets follows the lexicographic order in which the sets appear in the optimal solution. Please note that the offer sets are static over a number of periods. Thus, we have to check continuously (i.e., in each period) whether the capacity allows for offering the products contained in the static set. To do so, we use the capacity check of the surrogate network (i.e., constraints (11) and (12)). If the capacity is not sufficient to sell a product, it is removed from the set. Note that it is also possible to check capacity by solving the original feasibility problem (2)–(4).

Appendix D.4: Surrogate DPD approach (*DPD-surr*)

Our main method *DPD-surr* is obtained by applying the surrogate network to the DPD approach of Liu and van Ryzin (2008).

Analogously to *DPD-ah*, the starting point of the decomposition is the corresponding CDLP formulation; that is, CDLP-surr (D.1.8)–(D.1.12). Let π_h denote the optimal dual prices of regular resource h (constraint (D.1.9)). Furthermore, let $\tilde{\pi}_i$ denote the optimal dual prices of artificial resource i (constraint (D.1.10)).

We subsequently obtain the following two types of one-dimensional problems to assess the value of capacity of resources $h' \in \mathcal{H}$ and $i' \in \widetilde{\mathcal{H}}$:

$$V_t^{surr,h'}(c_{h'}) = \max_S\Big\{\sum_j \lambda \cdot P_j^{reg}(S) \cdot \big(r_j^{reg} - \sum_{h \neq h'} a_{hj} \cdot \pi_h - \sum_i \tilde{a}_{ij} \cdot \tilde{\pi}_i + \\ + V_{t-1}^{surr,h'}(c_{h'} - a_{h'j})\big) + (\lambda \cdot P_0(S) + 1 - \lambda) \cdot V_{t-1}^{surr,h'}(c_{h'})\Big\} \quad \text{(D.4.1)}$$

with boundary conditions $V_t^{surr,h'}(c_{h'}) = -\infty$ if $c_{h'} < 0$ and $V_0^{surr,h'}(c_{h'}) = 0 \; \forall c_{h'} \geq 0$ and

$$\tilde{V}_t^{surr,i'}(\tilde{c}_{i'}) = \max_S\Big\{\sum_j \lambda \cdot P_j^{reg}(S) \cdot \big(r_j^{reg} - \sum_h a_{hj} \cdot \pi_h - \sum_{i \neq i'} \tilde{a}_{ij} \cdot \tilde{\pi}_i \\ + \tilde{V}_{t-1}^{surr,i'}(\tilde{c}_{i'} - \tilde{a}_{i'j})\big) + \sum_k \lambda \cdot P_k^{flex}(S) \cdot \big(r_k^{flex} - \sum_{i \neq i'} \tilde{b}_{ik} \cdot \tilde{\pi}_i \\ + \tilde{V}_{t-1}^{surr,i'}(\tilde{c}_{i'} - \tilde{b}_{i'k})\big) + (\lambda \cdot P_0(S) + 1 - \lambda) \cdot \tilde{V}_{t-1}^{surr,i'}(\tilde{c}_{i'})\Big\} \quad \text{(D.4.2)}$$

with boundary conditions $\tilde{V}_t^{surr,i'}(\tilde{c}_{i'}) = -\infty$ if $\tilde{c}_{i'} < 0$ and $\tilde{V}_0^{surr,i'}(\tilde{c}_{i'}) = 0 \; \forall \tilde{c}_{i'} \geq 0$.

During the booking horizon, we approximate the opportunity cost as the sum of the required resources' opportunity cost. More formally, let regular resource h's opportunity cost be defined as

$$\Delta_h V_t^{surr,h}(c_h) := V_t^{surr,h}(c_h) - V_t^{surr,h}(c_h - 1) \quad \text{(D.4.3)}$$

and artificial resource i's opportunity cost be defined as

$$\widetilde{\Delta}_i V_t^{surr,i}(c_h) := \tilde{V}_t^{surr,i}(c_i) - \tilde{V}_t^{surr,i}(c_i - 1). \quad \text{(D.4.4)}$$

Then the offer set is the optimal solution of

$$\max_S\Big\{\sum_j \lambda \cdot P_j^{reg}(S) \cdot \big(r_j^{reg} - \sum_h a_{hj} \cdot \Delta_h V_{t-1}^{surr,h}(c_h) - \sum_i \tilde{a}_{ij} \cdot \widetilde{\Delta}_i V_{t-1}^{surr,i}(c_h)\big) \\ + \sum_k \lambda \cdot P_k^{flex}(S) \cdot \big(r_k^{flex} - \sum_i \tilde{b}_{ik} \cdot \widetilde{\Delta}_i V_{t-1}^{surr,i}(c_h)\big)\Big\} \quad \text{(D.4.5)}$$

Appendix E: Detailed values for products and segments in network 2

Regarding network 2, Table E.1 and Table E.2 summarize regular products' capacity consumption and revenues, as well as the segments' arrival probabilities, consideration sets, and preference weights.

Product	Legs	Revenue	Product	Legs	Revenue
1	1	1000	12	1	500
2	2	400	13	2	200
3	3	400	14	3	200
4	4	300	15	4	150
5	5	300	16	5	150
6	6	500	17	6	250
7	7	500	18	7	250
8	(2,4)	600	19	(2,4)	300
9	(3,5)	600	20	(3,5)	300
10	(2,6)	700	21	(2,6)	350
11	(3,7)	700	22	(3,7)	350

Table E.1: Description of regular products for network 2

Segment	Class	Probability	Consideration set	Preference vector
1	H	0.08	{1,8,9}	(10,5,5)
2	L	0.16	{12,19,20}	(10,10,5)
3	H	0.05	{2,3}	(10,10)
4	L	0.16	{13,14}	(10,10)
5	H	0.10	{4,5}	(10,10)
6	L	0.12	{15,16}	(10,5)
7	H	0.02	{6,7}	(10,5)
8	L	0.04	{17,18}	(10,10)
9	H	0.02	{10,11}	(10,5)
10	L	0.04	{21,22}	(10,10)
11	–	0.05	$\{f1\}$	(10)
12	–	0.02	$\{f2\}$	(10)
13	–	0.05	$\{f3\}$	(10)
14	–	0.04	$\{f4\}$	(10)
15	–	0.02	$\{f5\}$	(10)

Table E.2: Descriptions of customer segments for network 2