Itinerary-Based Nesting Control with Upsell

Chan Seng Pun, Diego Klabjan, Fikri Karaesmen, Sergey Shebalov

Abstract

In order to accept future high-yield booking requests, airlines protect seats from low-yield passengers. More seats may be reserved when passengers faced with closed fare classes can upsell to open higher fare classes. We address the airline revenue management problem with capacity nesting and customer upsell, and formulate this problem by a stochastic optimization model to determine a set of static protection levels for each itinerary. We apply an approximate dynamic programming framework to approximate the objective function by piecewise linear functions, whose slopes (marginal revenue) are iteratively updated and returned by an efficient heuristic that simultaneous handles both nesting and upsells. The resulting allocation policy is tested over a real airline network and benchmarked against the randomized linear programming bid-price policy under various demand settings. Simulation results suggest that the proposed allocation policy significantly outperforms when incremental demand or upsell probability are high. Structural analyses are also provided for special demand dependence cases.

Key words: network revenue management, capacity nesting, customer upsell, approximate dynamic programming

1 Introduction

Airline Revenue Management (RM) is about making a decision whether or not a booking request for a seat of a fare class should be accepted at a particular point in time. If the request is accepted, the revenue is immediately collected. Otherwise, the airline reserves the seat for an elite passenger who might book in the near future. Formally, RM is to maximize revenue by managing a capacity-constrained flight network. An RM control policy for such a purpose is often constructed based on the (primal and/or dual) solutions of a resource allocation problem. Constructing a good control policy has been an interesting topic to both practitioners and researchers for decades. Challenges are mainly due to the size of the flight network, the dynamic nature of the airline business, and the stochastic booking behaviors of customers. These challenges essentially drive the airline industry to the quest for simple and efficient heuristics.

At the beginning of the decision process, an underlying optimization problem allocates seats to passenger classes before passengers start booking (this optimization problem is typically resolved later during the booking process). By allocating the right amount of seats to each class, seats can be protected from low-yield passengers, who usually book their tickets earlier and are able to take seats that could later be sold to high-yield passengers. In practice, instead of using the allocation solution as is, allocations are nested over fare classes to set up protection levels, so that high-yield passengers with depleted allocations are allowed to book seats originally allocated to low-yield classes. A popular alternative to an allocation policy is the bid-price policy consisting of a threshold price (bid price) for each itinerary. A booking from a passenger paying a fare above the bid price is accepted given the itinerary is open. The optimal bid price is usually approximated by summing the (approximated) marginal value of a seat over all flights in the itinerary. Lastly, by assuming a relationship between the passenger population and fare level (price elasticity), the fare can be adjusted to achieve a similar capacity protection effect. For further details, we refer the reader to Williamson (1992) and Talluri and van Ryzin (1998) for bid prices, Bitran and Caldentey (2003) for pricing solutions, de Boer et al. (2002) for numerical experiments, and McGill and Van Ryzin (1999) for a comprehensive review of RM.

Although the bid-price policy has been extensively studied, the traditional seat allocation policy remains popular, owning to the fact that many existing RM systems are built to handle allocation policies due to their capability to include customer behavior (cancellation, no-show, and upsell) more intuitively and provide a more granular control over the network. With the ever tightening revenue margin nowadays, capturing customer behavior is vital to the prosperity of the airlines.

In practice, bid-prices are used not as a control policy but as a set of required inputs to prorate itinerary fares and to decompose the flight network during the pre-optimization phase, where the marginal value of a seat is approximated by a dual solution of a relaxed seat allocation model. After the fares are prorated, virtual classes are defined at the flight level and mapped to the original fare classes. The protection level for each virtual class is then determined by an efficient leg-based heuristic that captures customer behavior. Our work simplifies existing allocation-based RM systems by eliminating the use of the proration scheme and the need of defining virtual classes, and provides an allocation policy that requires no changes in management practice.

In this paper, we model a network RM problem by a stochastic program under an assumption that bookings are in a low-to-high arrival order. The model first maximizes revenue by allocating seats to each itinerary subject to flight capacity. Given an itinerary-level allocation and a demand sample, the problem then maximizes sales by distributing available seats to each fare class while capturing capacity nesting and customer upsell. We apply approximate dynamic programming (ADP) to approximate the objective function by piecewise linear functions. The slopes for each piecewise linear function are estimated by an efficient heuristic that locally adjusts class-level seat allocations and are iteratively updated given new demand information. Several other sophisticated algorithms are developed to reduce the running time of the heuristic. In addition, some special cases of demand dependence are considered: when demands are independent, we show that our model is the same as the model in Curry (1990), and the nested allocation policy, similar to the partitioned allocation policy, enjoys the asymptotic optimality in Cooper (2002). When demands depend only among fare classes but not across itineraries, we analyze a sufficient condition under which the objective function is concave. Simulation results on a medium airline network using a real-world dataset are discussed. Sensitivity analyses are conducted to access the performance of the nested allocation policy by varying demand magnitudes and upsell probabilities.

Our contributions are as follows.

1. We provide an innovative stochastic programming formulation to model the network RM problem which considers both capacity nesting and customer upsell at the itinerary level.
2. We are the first to solve the network RM problem in question and develop a parallelizable ADP algorithm that approximates the complicated objective function. The ADP algorithm allows us to closely approximate the problem by a linear program that can be solved using any efficient linear programming technique. Furthermore, it does not require techniques that introduce additional layers of suboptimality, e.g. linear relaxations, network decomposition by fare proration, or the use of virtual classes, and returns a static allocation policy that can be easily stored and implemented.
3. We devise a sophisticated heuristic to be served as the core of the ADP algorithm. It efficiently determines the set of protection levels and approximates the revenue and seat margin. We numerically show that it significantly outperforms the path-independent heuristic by Gallego et al. (2009) when the number of fare classes is high.
4. We justify the use of the nested allocation policy by revealing its asymptotic optimality when capacity and demand are high and demands are independent.
5. We generalize the results of Brumelle and McGill (1993) and the positive regression dependence in Cooper and Gupta (2006) to multiple classes when demands are discrete and only depend among fare classes.
6. We benchmark our allocation policy against RLP bid-prices to provide a comprehensive numerical study on the performance of the nested allocation policy when demands are scaled and when upsell probabilities cannot be accurately forecasted, a common problem in practice.

We outline our paper as follows. Section 2 provides a general overview on several well-known seat allocation models. Section 3 presents our itinerary-based nesting model with upsell, and Section 4 elaborates details on the approximate dynamic programming algorithm that we apply to solve our model. Several other heuristics and algorithms are also presented. Section 5 discusses asymptotic optimality of the nested allocation policy when demands are independent, and the structure of the objective function when demands depend only among fare classes. Section 6 reports simulation results, and Section 7 concludes the paper.

1.1 Literature Review

We briefly discuss previous works closely related to our materials. Starting with two passenger classes defined by their fares, Littlewood (1972) derives the optimality condition to determine the optimal protection level for the lowest fare class when passengers book first. Brumelle and McGill (1993), Curry (1990), and Wollmer (1992) independently generalize the optimality condition to multiple classes. While Wollmer (1992) handles discrete demand, Curry (1990) assumes continuous demand, and Brumelle and McGill (1993) is applicable to both. Furthermore, Curry (1990) proposes a two-step optimization procedure to obtain protection levels at the itinerary level. While we also consider the RM problem at the itinerary level, it is formulated as a stochastic program that can be solved efficiently with upsell.

Another distinction is in Curry's assumption of independent demand, while we allow demand to be correlated for our structural analysis of the revenue function.

Adding to the classical results of Littlewood (1972), Brumelle et al. (1990) analyze the effect of dependent demand on optimal booking limits for the case with two fare classes. Two special cases are considered. The first case is about correlated demands. The second case is about demand dependence on seat availability. They show that the policy structure for these two special cases is monotonic under a mild condition on the structure of the demand distribution. We extend their results for the first case to handle multiple fare classes.

A stochastic approximation algorithm is proposed by van Ryzin and McGill (2000) to approximate optimal protection levels. Their assumptions are the same as those in Brumelle and McGill (1993) that demands are independent and arrive in an low-to-high order. The algorithm does not rely on an apriori distribution and provably converges to optimality. However, their algorithm is single leg, and hence, does not capture the network effect.

Higle (2007) models the seat allocation problem by two-stage stochastic programming. Her model captures demand at the origin-destination level and computes protection levels at the flight level. Our two-stage stochastic model is similar, but we consider demand and protection levels at the itinerary level, and capture both the low-to-high order and customer upsell. Also, we do not require classes to be defined at the network level (across itineraries), which is nontrivial and required in her model for her flight-level nesting scheme to work.

Taking the two-stage framework one step further, Chen and Homem-de-Mello (2010) model the network RM problem by multi-stage stochastic programming. However, doing so rises tractability concerns, let alone they do not capture upsell.

Recent attentions have been given to integrating customer upsell with traditional revenue management models. Fiig et al. (2010) derive a fare adjustment scheme to handle discrete choice models. The scheme was tested by a passenger origin-destination simulator. Gallego et al. (2009) develop choice-based EMSR methods for a problem with multinomial logit (MNL) demand. They show superior performances over both EMSR-b with upsell in Belobaba and Weatherford (1996) and an adapted version of Fiig et al. (2010). We compare our algorithm directly with the path independent algorithm in Gallego et al. (2009) which has been the most promising heuristic, and achieve similar results when the number of classes is no more than three, but our algorithm significantly outperforms theirs when the number of fare classes exceeds three.

Zhang and Adelman (2009) propose the use of approximation dynamic programming to solve the network RM problem with customer choice. They provided multiple bounding results and a column generation algorithm to handle the MNL choice model with disjoint consideration sets studied in Liu and van Ryzin (2008). The ADP framework they applied is similar to ours. However, they estimate the set of products to sell instead of a set of seats to protect and do not consider upsell.

2 Overview of Revenue Management

Let T be the set of time periods (reading days), F be the set of flights, I be the set of itineraries, I_f be the set of itineraries that use flight f, $C_i = \{1, 2, \ldots, |C_i|\}$ be the set of fare classes on itinerary i ordered by fares $r_{i1} \geq \cdots \geq r_{i|C_i|}$, $J = \{(i, c)\}_{c \in C_i, i \in I}$ be the set of products (itinerary-fare combinations), J_f be the set of products that use flight f, Π_j be the nested allocation (protection level) of product j, K_f be the number of available seats on flight f, y_i be the number of seats allocated to itinerary i, x_{ic} for $j = (i, c)$ be the number of seats allocated to product j, z_{ic} be the number of empty seats extracted from all classes lower or equal to class c on itinerary i, D_{jt} be a random variable that represents the number of booking requests for product j at time t, and d_{jt} be its realization. If the demands are aggregated over all time periods, or each class of demands has a designated arrival time period, then the subscript to time t is ignored.

Any multidimensional quantity is denoted in bold. A superscript * denotes the optimal value of a problem. A subscript refers to the sliced set in the subscripted dimension, e.g. $\mathbf{\Pi}_i$ is the set of protection levels over all classes on itinerary i. Minimum and maximum operation is assumed component-wise, and $(\cdot)^+$ represents $\max\{\cdot, 0\}$. For ease of notation, we define the set of protection levels for classes not lower than c by $\mathbf{\Pi}_i^c = \{\Pi_{i1}, \ldots, \Pi_{ic}\}$, the set of feasible itinerary-level allocations by $\mathcal{I}(\mathbf{K}) = \{\mathbf{y} : \sum_{i \in I_f} y_i \leq K_f \text{ for } f \in F \text{ and } y_i \in \mathbb{N} \text{ for } i \in I\}$, and the set of feasible product-level allocation by $\mathcal{J}(\mathbf{K}) = \{\mathbf{x} : \sum_{j \in J_f} x_j \leq K_f \text{ for } f \in F \text{ and } x_j \in \mathbb{N} \text{ for } j \in J\}$. The summations in $\mathcal{I}(\mathbf{K})$ and $\mathcal{J}(\mathbf{K})$ represent the requirement that the total allocation to itinerary or product cannot exceed the number of seats available. Additionally, we define $\mathcal{P}(\mathbf{\Pi}_i, y_i) = \{\mathbf{x}_i : x_{ic} = \min\{y_i, \Pi_{ic}\} - \Pi_{ic-1} \text{ for } c \in C_i\}$ a policy function that maps a set of protection levels (nested allocations) to a set of partitioned (non-nested) allocations given the total number of seats available for an itinerary. We define its inverse function by $\mathcal{N}(\mathbf{x}_i) = \{(\mathbf{\Pi}_i, y_i) : \Pi_{ic} = \sum_{c' \leq c} x_{ic'}$ for $c \in C_i, y_i = \sum_{c \in C_i} x_{ic}\}$.

Starting with the dynamic programming (DP) formulation of the RM problem with independent demand described in Talluri and van Ryzin (1998), we discuss several tractable approximation models to the DP value function in each time period. The DP accurately models the problem if the probability of having more than one arrival in between two time periods is negligible, or as a special case, if arrivals in between two time periods only belong to the same class (see Robinson (1995)). Mathematically, the optimality equation is

$$v_t(\mathbf{K}) = \mathbb{E}\left[\max_{\substack{\mathbf{x} \in \mathcal{J}(\mathbf{K}) \\ 0 \leq x_j \leq D_{jt}, j \in J}} \sum_{j \in J} r_j x_j + v_{t-1}\left(\left\{K_f - \sum_{j \in J_f} x_j\right\}_{f \in F}\right)\right] \tag{1}$$

with $v_0(\cdot) = 0$. For each time period t, a decision has to be made about the number of accepted products. In the end, the DP returns a table of dynamic controls to indicate which classes are open for each possible demand scenario over all time periods. Major challenges include the curse of dimensionality (see Powell (2007)) and tremendous storage requirements of the dynamic controls. Promising techniques have been developed to cope with these challenges by approximating the value function in a way that a set of static controls can be efficiently retrieved. Several related models are presented in sequel.

The stochastic seat allocation model $SP^*(\mathbf{K}) = \max_{\mathbf{x} \in \mathcal{J}(\mathbf{K})} \sum_{j \in J} \mathbb{E}[r_j \min\{x_j, D_j\}]$ is widely used. It aggregates demand for the remaining time periods and aims to maximize the expected revenue by allocating available seats to each product. Revenue can only be extracted given that both bookings and allocations exist. While this model is intuitive, it assumes a high-to-low booking arrival order and yields a set of partitioned seat allocations that do not consider nesting. Its continuous relaxation is known as the probabilistic nonlinear programming model (PNLP), and the deterministic version of PNLP is known as the deterministic linear programming model (DLP) (see Talluri and van Ryzin (2004)).

To incorporate demand stochasticity while keeping the simplicity of DLP, Talluri and Van Ryzin (1999) propose the use of a randomized linear program $RLP^*(\mathbf{K}) = \mathbb{E}[\max_{\mathbf{x} \in \mathcal{J}(\mathbf{K})} \sum_{j \in J} r_j \min\{x_j, D_j\}]$. It has been theoretically proven that RLP's bid-prices outperform DLP bid-prices, which are dual prices.

To capture nesting over multiple fare classes, Curry (1990) derives an itinerary-based allocation model, which yields a set of static protection levels for each itinerary. It is optimal if the arrival order is low-to-high and can be solved efficiently by a two-stage procedure. Let ξ be the number of remaining seats given to an itinerary. The model with independent demand reads

$$IP^*(\mathbf{K}) = \max_{\mathbf{x} \in \mathcal{J}(\mathbf{K})} \left\{\sum_{i \in I} R_{i|C_i|}(\mathbf{\Pi}_i, y_i) \middle| (\mathbf{\Pi}_i, y_i) \in \mathcal{N}(\mathbf{x}_i) \text{ for } i \in I\right\}$$

and

$$\begin{aligned} R_{ic}(\mathbf{\Pi}_i^{c-1}, \xi) = &\int_0^{\xi - \Pi_{ic-1}} \left[r_{ic} d_{ic} + R_{ic-1}\left(\mathbf{\Pi}_i^{c-2}, \xi - d_{ic}\right) f_{ic}(d_{ic})\right] dd_{ic} \\ &+ \left(r_{ic}(\xi_i - \Pi_{ic-1}) + R_{ic-1}(\mathbf{\Pi}_i^{c-2}, \Pi_{ic-1})\right) \int_{\xi_i - \Pi_{ic-1}}^{\infty} f_{ic}(d_{ic}) dd_{ic}, \end{aligned} \tag{2}$$

where $R_{i0} = 0, \Pi_{i0} = 0$, and $f_j(\cdot)$ is the demand density function for product j. The revenue function (2) is recursive and has a state space of protection levels and remaining empty seats. It collects revenue by accepting booking requests for class c and adjusts the remaining seats before proceeding to class $c - 1$. Note that since booking requests arrive in a low-to-high arrival order, time index can be ignored. The discrete version of (2) can be found in Wollmer (1992).

All the optimization models discussed above are rather intuitive and well-studied. In the following sections, we first discuss how to extend $IP(\mathbf{K})$ to capture customer upsell when demands are multinomial logit. Our solution method is then proposed.

3 Itinerary-based Nested Allocation Model with Upsell

In this section, we develop a network model that captures both capacity nesting and customer upsell by adding upsells to $IP(\mathbf{K})$ and assuming a low-to-high arrival order. Furthermore, we show that it has a stochastic formulation with a structure that allows us to develop an efficient algorithm.

For capacity nesting, a customer with its corresponding class closed may be given a seat from any lower classes. For customer upsell, it is the opposite. A customer may be willing to pay more to obtain a seat if its corresponding fare class is closed. The former is a choice of the airline with passengers accepting the requests. The later is a choice of the customer with a probability that is usually assumed multinomial logit (MNL). For the MNL demand model, the upsell probability for class c is determined by its attractiveness $a_c = exp(\beta_s s_c + \beta_r r_c)$, where β_s and β_r are the elasticities of the schedule and fare, and s_c and r_c are the schedule quality and fare of a class-c ticket. The upsell probability from class c to c' is computed based on the proportion of the attractiveness of higher classes, e.g. $p_{cc'} = a_{c'}/(\sum_{l=1}^{c} a_l + a_{|C_i|})$, where $a_{|C_i|}$ is the attractiveness of alternative options. For more information about MNL, we refer the reader to Gallego et al. (2009).

For an itinerary i, let $U_{icc'}$ be the number of upsells from class c to class c', $u_{icc'}$ be its realization, $\boldsymbol{\eta}_i$ be a vector of accumulated upsells of size $|C_i|$, $\mathbf{p}_i(c) = (p_{c1}, \dots, p_{c|C_i|})$ be the vector of upsell probabilities for class c with $p_{cc} = \dots = p_{c|C_i|-1} = 0$ and $p_{c|C_i|}$ being the probability of selecting an alternative carrier. For a given number of rejected bookings n, let $\mathbf{q}(n, \mathbf{p}(c), c) = (U_{ic1}, \dots, U_{ic|C_i|})$ be the vector of upsells for class c based on a multinomial probability distribution $\mathcal{B}(n, \mathbf{p}(c))$. Note that for each realization $\mathbf{u}_{ic}$, we have $\sum_{c' \in C_i} u_{icc'} = n$ and $\sum_{c' \in C_i} p_{icc'} = 1$. The itinerary-based nested allocation model with upsell is

$$U'^*(\mathbf{K}) = \max_{\mathbf{x} \in \mathcal{J}(\mathbf{K})} \left\{ \sum_{i \in I} V_{i|C_i|}(\mathbf{\Pi}_i, y_i, \mathbf{0}) \middle| (\mathbf{\Pi}_i, y_i) \in \mathcal{N}(\mathbf{x}_i) \text{ for } i \in I \right\},$$

and the revenue function is

$$V_{ic}(\mathbf{\Pi}_i^{c-1}, \xi, \boldsymbol{\eta}_i) = \begin{cases} \mathbb{E}\left[r_{ic} \min\{\xi - \Pi_{ic-1}, D_{ic} + \eta_{ic}\} \right. \\ \quad + V_{ic-1}\left(\mathbf{\Pi}_i^{c-2}, \xi - \min\{\xi - \Pi_{ic-1}, D_{ic} + \eta_{ic}\}, \right. \\ \quad\quad \left.\left. \boldsymbol{\eta}_i + \mathbf{q}(D_{ic} - (\xi - \Pi_{ic-1} - \eta_{ic})^+, \mathbf{p}_i(c), c) \right) \right] & \text{if } \xi \geq \Pi_{ic-1}, \\ \mathbb{E}[V_{ic-1}(\mathbf{\Pi}_i^{c-2}, \xi, \boldsymbol{\eta}_i + \mathbf{q}(D_{ic}, \mathbf{p}_i(c), c))] & \text{otherwise,} \end{cases}$$

where $\mathbf{0}$ is a vector of zeros of size $|C_i|$. When the number of the remaining seats is larger than the number of the protected seats for higher classes, e.g. $\xi \geq \Pi_{ic-1}$, the revenue for class c is computed based on the minimum of the available seats and the total demands (demands for class c plus upsells). If no seat is available for class c, all demands for class c are declined, and the resulting upsells are added to $\boldsymbol{\eta}_i$. The main differences between $IP(\mathbf{K})$ and $U'(\mathbf{K})$ are that the vector of the observed upsells $\boldsymbol{\eta}_i$ is now a part of the state space, and η_{ic} the total number of upsells to class c is added whenever demands are considered.

Proposition 1. *Problem $U'(\mathbf{K})$ exhibits the following stochastic formulation:*

$$U^*(\mathbf{K}) = \max_{\mathbf{x} \in \mathcal{J}(\mathbf{K})} \sum_{i \in I} \mathbb{E}\left[Q_i(\mathbf{x}_i, \mathbf{D}_i) \right]. \tag{3}$$

The revenue function for each itinerary i is

$$Q_i(\mathbf{x}_i, \mathbf{d}_i) = \max_{\mathbf{z}} \mathbb{E}\left[\sum_{c \in C_i} r_{ic} \min\left\{ x_{ic} + z_{ic+1}, d_{ic} + \psi_{ic} \right\} \right] \tag{4}$$

subject to

$$z_{ic} = (x_{ic} + z_{ic+1} - d_{ic} - \psi_{ic})^+ \qquad c \in C_i \tag{5}$$

$$(U_{ic1}, \dots, U_{ic|C_i|}) = \mathbf{q}\left(d_{ic} - (x_{ic} + z_{ic+1} - \psi_{ic})^+, \mathbf{p}_i(c), c \right) \qquad c \in C_i \tag{6}$$

$$\sum_{c'=c-1}^{|C_i|-1} U_{ic'c} = \psi_{ic} \qquad c \in C_i$$

$$z_{ic} \in \mathbb{N} \qquad c \in C_i$$

where the first expectation is taken over demands, and the second expectation is taken over upsells given a demand sample.

Proof. We rely on the relationships $\xi - \Pi_{ic-1} = x_{ic} + z_{ic+1}$ and $\eta_{ic} = \psi_{ic}$ and prove the proposition by induction. For a given itinerary i and $c = 1$, we have $V_{i1}(\emptyset, \xi, \boldsymbol{\eta}_i) = \mathbb{E}[r_{i1} \min\{\xi, D_{i1} + \eta_{i1}\}] = \mathbb{E}[r_{i1} \min\{x_{i1} + z_{i2}, D_{i1} + \psi_{i1}\}]$. Suppose $V_{ic-1}(\boldsymbol{\Pi}_i^{c-2}, \xi, \boldsymbol{\eta}_i)$ and $\mathbb{E}[\sum_{c'=1}^{c-1} r_{ic'} \min\{x_{ic'} + z_{ic'+1}, D_{ic'+1} + \psi_{ic'}\}]$ are equal. We have

$$
\begin{aligned}
&V_{ic}(\boldsymbol{\Pi}_i^{c-1}, \xi, \boldsymbol{\eta}_i)\\
&=\mathbb{E}[r_{ic} \min\{\xi - \Pi_{ic-1}, D_{ic} + \eta_{ic}\}\\
&\qquad + V_{ic-1}(\boldsymbol{\Pi}_i^{c-2}, \xi - \min\{\xi - \Pi_{ic-1}, D_{ic} + \eta_{ic}\}, \boldsymbol{\eta}_i + \mathbf{q}(D_{ic} - (\xi - \Pi_{ic-1} - \eta_{ic})^+), \mathbf{p}_i(c), c)]\\
&=\mathbb{E}\Bigg[r_{ic} \min\left\{x_{ic} + z_{ic+1}, D_{ic} + \psi_{ic}\right\}\\
&\qquad + V_{ic-1}\Bigg(\boldsymbol{\Pi}_i^{c-2}, \sum_{c'>c} x_{ic'} + x_{ic} + z_{ic+1} - \min\left\{x_{ic} + z_{ic+1}, D_{ic} + \psi_{ic}\right\},\\
&\qquad\qquad \boldsymbol{\eta}_i + \mathbf{q}\left(d_{ic} - \left(x_{ic} + z_{ic+1} - \psi_{ic}\right)^+, \mathbf{p}_i(c), c\right)\Bigg)\Bigg]\\
&=\mathbb{E}\left[r_{ic} \min\left\{x_{ic} + z_{ic+1}, D_{ic} + \psi_{ic}\right\} + V_{ic-1}\left(\boldsymbol{\Pi}_i^{c-2}, \sum_{c'>c} x_{ic'} + z_{ic}, \boldsymbol{\eta}_i + (U_{ic1}, \ldots, U_{ic|C_i|})'\right)\right]\\
&=\mathbb{E}\left[\sum_{c'=1}^{c} r_{ic'} \min\left\{x_{ic'} + z_{ic'+1}, D_{ic'} + \psi_{ic'}\right\}\right].
\end{aligned}
$$

The third equality is by the definition of (5) and (6). $\square$

The problem (3) first maximizes the expected revenue by allocating seats to each product. Once the set of allocated seats and a demand sample is given, sales are maximized while considering capacity nesting and customer upsell. Constraints (5) and (6) are the definitions of z_{ic} and $U_{icc'}$ for $c' \in C$. Note that if the expectation over upsell is ignored, the revenue function $Q_i(\mathbf{x}_i, \mathbf{d}_i)$ is similar to the objective function of DLP but with variable z_{ic+1} to account for the accumulated empty seats from lower classes and the term ψ_{ic} to handles upsells to class c. This alternative formulation of the problem has a structure that allows us to develop an efficient algorithm, which is discussed next.

4 Solution Methodology

We employ an ADP framework described in Powell et al. (2004) to solve our problem. It is specifically designed for a two-stage stochastic problem with the following properties: the objective function is separable in the first stage decision, stochastic information can be easily collected, and subgradient to the objective function can be computed.

The idea is to approximate the originally complicated objective function by basis functions which can be easily encoded. In the algorithm, the first-stage problem is approximated by using the approximated objective function. The first-stage solution together with some observed stochastic information are fed to the second-stage problem to estimate the required slopes for updating the basis functions. As this procedure iterates and more information is observed, the original objective function can be approximated arbitrarily closely under some conditions. Powell et al. (2004) show under such conditions that the algorithm converges.

Beside some theoretical guarantees, there are two major practical benefits from applying this ADP framework: 1) the first-stage problem is often easier to solve when the original objective function is replaced by some simple basis functions, and 2) as the problem is separable in the first-stage decision, the second-stage problem can be parallelized.

This ADP framework especially suits our problem as 1) the objective function (3) can be separable in itineraries, 2) demands and upsells can be easily simulated, and 3) slopes can be estimated directly based on the recursive structures of (5) and (6). In our application, we use piecewise linear functions as the basis functions. To have the objective function (3) separable in itineraries, we further decompose our problem into two sub-problems by adding an auxiliary variable y_i to represent the number of seats allocated to each itinerary, where the first sub-problem is

$$U^{sp1}(\mathbf{K}) = \max_{\mathbf{y} \in \mathcal{I}(\mathbf{K})} \sum_{i \in I} U_i^{sp2}(y_i),$$

and the second sub-problem is

$$U_i^{sp2}(y_i) = \max_{\mathbf{x}} \left\{ \mathbb{E}\left[Q_i(\mathbf{x}_i, \mathbf{D}_i)\right] : \sum_{c \in C_i} x_{ic} = y_i \text{ and } x_j \in \mathbb{N} \text{ for } j \in J \right\}.$$

A side benefit from this modification is that by decoupling the class-level allocations from the network-level capacity constraints, we can focus on obtaining class-level allocation and accommodating upsells locally and independently for each itinerary.

Let $S_i = \min_{f \ni i}\{K_f\}$ be an upper bound on the number of seats that can be allocated to itinerary i, and v_{is} be the marginal revenue of allocating one more seat up to level s. The ADP algorithm approximates $U_i^{sp2}(y_i)$ by a piecewise linear function $\mathcal{Q}_i(y_i) = \sum_{s=1}^{l} v_{is} + v_{is+1}(y_i - l)$ for a unique integer l satisfying $l \leq y_i < l + 1$. Note that the function has S_i many breakpoints, and we assume $\mathcal{Q}_i(0) = 0$. With the piecewise linear functions, the ADP algorithm easily approximates $U^{sp1}(\mathbf{K})$ and obtains a set of itinerary-level allocations. For a given itinerary allocation $\bar{y}_i$, an upsell heuristic (the core of the ADP algorithm, presented later) is run to approximate $U_i^{sp2}(\bar{y}_i)$. It returns an approximated marginal revenue of $U_i^{sp2}(\cdot)$ at $\bar{y}_i$. The margin is, in turn, used to refine the accuracy of $\mathcal{Q}_i(\cdot)$. In the algorithm, a set of class-level allocations is also iteratively collected and updated for each itinerary allocation level ever encountered. The complete ADP algorithm is presented in Algorithm 1.

Algorithm 1 ADP Algorithm to approximate $U^{sp1}(K)$

1: Initialize v_{is} for $s = 1, \ldots, S_i$ and $i \in I$.
2: **while** stopping criteria are not met **do**
3: Solve $\bar{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{I}(\mathbf{K})} \{\sum_{i \in I} \mathcal{Q}_i(y_i)\}$
4: **for all** $i \in I$ **do**
5: Run the upsell heuristic to obtain $\{\hat{x}_{ic}\}_{c \in C_i}$ the partitioned allocations and $\hat{v}_i$ the marginal revenue.
6: **for all** $c \in C_i$ **do**
7: Update $\bar{x}_{ic}(\bar{y}_i)$ by $(1 - \beta_{ic})\bar{x}_{ic}(\bar{y}_i) + \beta_{ic}\hat{x}_{ic}$
8: Update stepsize β_{ic} by McClain's formula (see Powell, 2007, chap. 6).
9: **end for**
10: Update $\bar{v}_{i\bar{y}_i}$ by $(1 - \alpha_i)v_{i\bar{y}_i} + \alpha_i \hat{v}_i$.
11: Update stepsize α_i by bias-adjusted Kalman filter stepsize rule (see Powell, 2007, chap. 6).
12: Set $\mathbf{v}_i = \arg\min\{\sum_{s=1}^{S_i}(\delta_{is} - \bar{v}_{is})^2 | \delta_{is+1} \leq \delta_{is} \text{ for } s = 1, \ldots, S_i\}$.
13: **end for**
14: **end while**
15: **return** $\bar{x}_{ic}(\bar{y}_i)$ for $c \in C_i$ and $i \in I$.

Line 1 of the ADP algorithm initializes the marginal seat revenue (slope) for each possible allocation level over all itineraries. Line 2 stops the algorithm when changes of the slopes are negligible. Line 3 approximates $U^{sp1}(\mathbf{K})$ with its objective function replaced by the sum of the piecewise linear functions and returns the number of seats allocated to each itinerary. Note that $\mathcal{Q}_i(\cdot)$ is concave, as its slopes $\{v_{is}\}$ are decreasing in s, a property ensured by the projection operation in Line 12. This property enables us to linearize the approximation model and rewrite it into the following equivalent linear program:

$$\max_{\substack{\mathbf{y} \in \mathcal{I}(\mathbf{K}) \\ \boldsymbol{\rho}}} \left\{ \sum_{i \in I} \sum_{s=1}^{S_i} v_{is}\rho_{is} : \sum_{s=1}^{S_i} \rho_{is} = y_i \text{ for } i \in I, 0 \leq \rho_{is} \leq 1 \text{ for } s = 1, \ldots, S_i \right\}.$$

Given $\bar{y}_i$, Line 5 computes a set of new partitioned allocations and returns the seat margin at $\bar{y}_i$ using the upsell heuristic, which we developed to solve $U^{sp2}(\bar{y}_i)$. Line 7 updates the partitioned allocations based on $\{\hat{x}_{ic}\}_{c \in C}$, and Line 8 updates β_{ic} the stepsize for updating the allocation. Line 10 updates the slope using $\hat{v}_i$ from Line 5 while Line 11 updates α_i the stepsize for updating the slope. Note that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are in fact state-dependent, but the dependence is dropped in Algorithm 1 for ease of notation. After the slope is updated, $\mathcal{Q}_i(y)$ may not be concave in y. Line 12 then imposes concavity on each piecewise linear function by finding the closest concave piecewise linear function, where the distance is measured by the two-norm. An efficient projection algorithm can be found in Powell et al. (2004). With the well-approximated objective function, Line 15 returns a set of partitioned allocations given the latest allocation for each itinerary from Line 3.

We now discuss the upsell heuristic which estimates the marginal revenue for $U_i^{sp2}(\cdot)$ at $\bar{y}_i$ and yields a set of partitioned allocations to be updated by the ADP. The heuristic iteratively adjusts the existing partitioned allocations by removing seats from less-profitable classes to more-profitable classes. It first generates a set of demand scenarios, finds the highest class having a positive allocation, and sets it as the current class $\hat{c}$. The algorithm then iteratively reduces the number of allocated seats from class $\hat{c}$ and reallocates the extracted seats to lower classes that can most profitably utilize the seats. If no profitable lower class is found or no more allocation can be extracted, the algorithm repeats with the next highest class having a positive allocation. This procedure continues until the highest class having a positive allocation is also the lowest class. The algorithm is described in Algorithm 2.

Algorithm 2 Upsell heuristic to approximate $U_i^{sp2}(y_i)$

Require: y_i the total number of seats available.
1: Generate K demand samples $\{\zeta_{ic}^1, \dots, \zeta_{ic}^K\}$ for $c \in C_i$.
2: Initialize x_{ic} for $c \in C_i$ using the upsell-adjusted algorithm.
3: Set the current class $\hat{c}$ to the highest class with a positive allocation.
4: **while** class $\hat{c}$ is not the lowest class **do**
5:   Decrement $x_{i\hat{c}}$ by 1.
6:   Apply the revenue estimation algorithm to compute revenue r.
7:   Find c' a lower class that yields the highest margin $\delta_{c'}$ by the margin estimation algorithm.
8:   Increment $x_{ic'}$ by 1.
9:   **if** $\hat{c} = c'$ or $x_{i\hat{c}} = 0$ **then**
10:     Find $\tilde{c}$ the next highest class with a positive allocation.
11:     **if** class $\tilde{c}$ exists **then**
12:       Set $\hat{c} = \tilde{c}$
13:     **else**
14:       **return** $\mathbf{x}_i, r + \delta_{c'}, \delta_{c'}$.
15:     **end if**
16:   **end if**
17: **end while**

Line 1 generates demand samples used to estimate the total revenue and seat margin. Line 2 initializes class-level allocations using the upsell-adjusted algorithm designed to handle upsells when the rejection event $\{\sum_{c \in C_i} \zeta_{ic} > y_i\}$ is very likely. Line 3 finds $\hat{c}$ the highest class with a positive allocation. If $\hat{c}$ is not the lowest class, Line 5 subtracts a seat from class $\hat{c}$. Line 6 then applies the revenue estimation algorithm to compute the base revenue to which the new margin is to be added. Line 7 estimates the new margin using the margin estimation algorithm, which also returns c' the class that is associated with the new margin and can most profitably utilize the seat. After the extracted seat is added to class c', the upsell heuristic then finds the next highest class having a positive allocation at Line 10. If such a class exists, the algorithm starts the next iteration with that class. Otherwise, the algorithm returns the set of modified allocations, the associated revenue, and the seat margin at Line 14.

For completeness, we also briefly summarize here the revenue estimation algorithm (Algorithm 3), the margin estimation algorithm (Algorithm 4), and the upsell-adjusted algorithm (Algorithm 5) with their algorithmic details documented in Appendix B.1, B.2, and B.3 respectively.

The revenue estimation algorithm is basically an implementation-level verbatim of $Q_i(\mathbf{x}_i, \mathbf{d}_i)$. It takes $\mathbf{x}_i$ the number of seats allocated to each class, $\boldsymbol{\zeta}_i$ the set of generated demand samples, and returns r the average revenue over all demand samples along with all upsell information for the margin estimation algorithm to efficiently compute the seat margin. It heavily relies on the nested recursive structure of (5) and (6) to compute the revenue directly.

The margin estimation algorithm recovers our single-seat allocation decision to provide a what-if margin. It requires the same inputs as those for the revenue estimation algorithm, the upsell information, as well as the class c' that the extra seat is adding to. It starts with checking if there exists a rejected upsell from any lower classes. If a rejected upsell is found, it returns the class-c' fare. Otherwise, for any higher classes $l = 1, \dots, c'$, the algorithm searches for a rejected booking and its corresponding upsell. If both are found, the algorithm recovers the margin as if there was no upsell from class l. If a rejected booking exists but does not result in an upsell, the algorithm returns the class-l fare as the seat margin.

The upsell-adjusted algorithm is designed to handle situations when there are many rejected booking requests. It takes y_i the total number of available seats and $\boldsymbol{\zeta}_i$ the set of demand samples, and returns $\mathbf{x}_i$ a set of initial class-level allocations for the upsell heuristic. The algorithm basically subtracts upsells from lower classes and adds them to the observed demands for higher classes, and treats the total demand as if it is the original demand, e.g. approximating the MNL distribution with the distribution of the demand. The upsell-adjusted algorithm then iterates until no profitable upsell can be added to any higher classes over all demand samples.

5 Special Dependence Cases

In this section, we investigate structural properties of our model under special demand dependence cases. When demand is independent, we show that the nested allocation policy is asymptotically optimal. A comprehensive analysis on asymptotic optimality for partitioned allocation policy is given by Cooper (2002), yet a result for nested allocation policies has not been derived. The asymptotic optimality ensures that when both capacity and demand are sufficiently large, the allocation policy will perform arbitrarily closely to the optimal control policy. The idea is to scale both the mean of the demand and capacity without scaling their variabilities. After the scaled means are normalized, the variabilities are essentially eliminated. It results in having similar performance over a group of allocation policies from different approximation models to $v_{|T|}(\mathbf{K})$, the value function of the optimal DP.

In the second part of this section, we investigate the case when there are correlations between demands. We generalize the observation about the positive regression dependence in Brumelle et al. (1990) and Cooper and Gupta (2006), and extend the results on the optimality conditions for correlated demands in Brumelle and McGill (1993) to handle multiple classes. We do not only reveal the concavity of the revenue function under the positive regression dependence, but also a sufficient condition for determining and verifying the optimality of the protection levels given the total allocation to an itinerary.

5.1 Asymptotic Property

In this section, our goal is to show the asymptotically optimality of the nested allocation policy of $U(\mathbf{K})$ when demands are independent. Let us validate $P(\mathbf{K})$, the version of $U(\mathbf{K})$ without upsell by showing that it is equivalent to $IP(\mathbf{K})$ when demands are independent. Our problem $U(\mathbf{K})$ with independent demand is

$$P^*(\mathbf{K}) = \max_{\mathbf{x} \in \mathcal{J}(\mathbf{K})} \sum_{i \in I} \mathbb{E}\left[\mathcal{S}_i(\mathbf{x}_i, \mathbf{D}_i)\right],$$

and the second-stage problem is

$$\begin{aligned} \mathcal{S}_i(\mathbf{x}_i, \mathbf{d}_i) = \max_{\mathbf{z} \text{ integer}} & \sum_{c \in C_i} r_{ic} \min\{x_{ic} + z_{ic+1}, d_{ic}\} \\ z_{ic} = & (z_{ic+1} + x_{ic} - d_{ic})^+ \qquad c \in C_i \end{aligned} \tag{7}$$

At the first stage, seats are allocated to each product subject to flight capacity. At the second stage, bookings are accepted based on the first stage allocations given a realization of the demand.

Proposition 2. *Assuming independent and continuous demand with a low-to-high arrival order, $P(\mathbf{K})$ and $IP(\mathbf{K})$ are equivalent.*

Proof. By exploring the structure of (7) and expressing expectations by integrals, we can easily recover the recursive structure of $IP(\mathbf{K})$. See Appendix A.3 for details. □

In fact, the assumption about continuous demand can be dropped, since Li and Oum (2002) establish the equivalence between $IP(\mathbf{K})$ and the model in Wollmer (1992). Thus, our model also works with discrete demand.

After we have verified that $P(\mathbf{K})$ is in fact the correct model with independent demand, we next study the bounding property of different approximation models to $v_{|T|}(\mathbf{K})$. Specifically, we show that $\overline{P}(\mathbf{K})$, the version of the nested allocation model with deterministic demands, yields the same objective value as $\overline{SP}(\mathbf{K})$, the version of the stochastic seat allocation model with deterministic demands, and the nested allocation policy of $P(\mathbf{K})$ is provably better than the partitioned allocation policy of $SP(\mathbf{K})$. With these two observations, we can directly apply the result from Cooper (2002) to obtain asymptotic optimality of the protection levels.

Lemma 1. *We have* $\overline{SP^*}(\mathbf{K}) = \overline{P^*}(\mathbf{K})$.

Proof. By substituting stochastic demand by its expectation, we can rewrite the recursive structure of (2) by a revenue function equivalent to that of $\overline{SP}(\mathbf{K})$. See Appendix A.1 for details. □

Let us denote by $\mathcal{R}^\pi$ the revenue obtained by implementing the allocation policy constructed based on the solution of problem π given a demand sample.

Lemma 2. *We have* $\mathbb{E}\mathcal{R}^{P(\mathbf{K})} \geq \mathbb{E}\mathcal{R}^{SP(\mathbf{K})} \geq \mathbb{E}\mathcal{R}^{\overline{SP}(\mathbf{K})}$ *for any arrival order.*

Proof. The proof is based on the observation that applying the nested allocation policy under the worst arrival order (low-to-high) yields a higher revenue than applying the partitioned allocation policy, which is arrival-order independent, and the fact that the partitioned allocation policy from $SP(\mathbf{K})$ is optimal under stochastic demand. See Appendix A.2 for details. □

Proposition 3. *We have* $\overline{SP^*}(\mathbf{K}) = \overline{P^*}(\mathbf{K}) \geq v_{|T|}(\mathbf{K}) \geq \mathbb{E}\mathcal{R}^{P(\mathbf{K})} \geq \mathbb{E}\mathcal{R}^{SP(\mathbf{K})} \geq \mathbb{E}\mathcal{R}^{\overline{SP}(\mathbf{K})}$.

Proof. ' Inequality $v_{|T|}(\mathbf{K}) \geq \mathbb{E}\mathcal{R}^{P(\mathbf{K})}$ follows from optimality of (1). Inequality $\overline{SP^*}(\mathbf{K}) \geq v_{|T|}(\mathbf{K})$ is from Cooper (2002). The proof is then completed by applying Lemma 1 and Lemma 2. □

Proposition 3 states that in expectation, we do not need nesting, and the objective value of $\overline{P}(\mathbf{K})$ can be used as an upper bound on $v_{|T|}(\mathbf{K})$. We can then directly apply Proposition 2 in Cooper (2002) to obtain asymptotic optimality of the nested allocation policy of $P(\mathbf{K})$. We apply superscript k to problem π to represent the version of π with its demand and capacity being k times larger. Furthermore, we denote by $D^k_{jt} = kD_{jt}$ the k-time larger demand for product j at time t.

Proposition 4. *If the normalized k-time linearly-scaled arrivals converge in distribution to their unscaled means, e.g.* $\frac{1}{k}\sum_{t\in T} D^k_{jt} \xrightarrow{\mathcal{D}} \sum_{t\in T}\mathbb{E}D_{jt}$, *the nested allocation policy from* $P^k(\mathbf{K})$ *is asymptotically optimal as* $k \to \infty$.

Proof. By Proposition 3, we have $\mathbb{E}\mathcal{R}^{\overline{SP^k}(\mathbf{K})}/v^k_{|T|}(\mathbf{K}) \leq \mathbb{E}\mathcal{R}^{P^k(\mathbf{K})}/v^k_{|T|}(\mathbf{K}) \leq 1$. From Proposition 2 in Cooper (2002), it follows $\mathbb{E}\mathcal{R}^{\overline{SP^k}(\mathbf{K})}/v^k_{|T|}(\mathbf{K}) \xrightarrow{k\to\infty} 1$, and in turn, we have $\mathbb{E}\mathcal{R}^{P^k(\mathbf{K})}/v^k_{|T|}(\mathbf{K}) \xrightarrow{k\to\infty} 1$ as required. □

If demands are not independent to each others, the arrival order affects the magnitude of the demands. As a result, Lemma 2 does not necessary hold (it is now possible to have $\mathbb{E}\mathcal{R}^{P(\mathbf{K})} < \mathbb{E}\mathcal{R}^{\overline{SP}(\mathbf{K})}$), and Proposition 4 is invalid. The above result is not as obvious as it seems. Since the revenue function is not concave in general, the deterministic upper bound on $v_{|T|}(\mathbf{K})$ cannot be simply implied by Jensen's inequality. To see this, suppose we have only two classes. The objective function of $\overline{P}(\mathbf{K})$ becomes $r_2 \min\{\xi - \Pi_1, \mathbb{E}D_2\} + r_1 \min\{\Pi_1 + (\xi - \Pi_1 - \mathbb{E}D_2)^+, \mathbb{E}D_1\}$. Its stochastic version is $\mathbb{E}[r_2 \min\{\xi - \Pi_1, D_2\}] + r_1\mathbb{E}[\mathbb{E}[\min\{\Pi_1 + (\xi - \Pi_1 - D_2)^+, D_1\}|D_2]]$. We can ignore the first term because of Jensen's inequality. Without relying on Lemmas 1 and 2, it is necessary to show $\min\{\Pi_1 + (\xi - \Pi_1 - \mathbb{E}D_2)^+, \mathbb{E}D_1\} \geq \mathbb{E}[\mathbb{E}[\min\{\Pi_1 + (\xi - \Pi_1 - D_2)^+, D_1\}|D_2]]$ for $\overline{P^*}(K) \geq \mathbb{E}\mathcal{R}^{P(K)}$ to hold, and this condition is not true in general even when demands are independent. Since the function is not concave and is non-increasing in D_2, the conditions for Jensen's inequality over this multivariate function are not satisfied. In the next section, we study some more structural properties of the revenue function when demands are correlated.

5.2 Optimality Condition for Correlated Demands

In this section, we show a sufficient condition that allows the objective function to be concave in the number of remaining seats, and optimality condition for determining protection levels. Let $\mathbf{d}^c_i = \{d_{ic}, \ldots, d_{i|C_i|}\}$ be a vector of the realized demands for classes $c, \ldots, |C_i|$. The modified $IP(\mathbf{K})$ that captures dependent demands among fare classes is:

$$\max_{\mathbf{x}\in\mathcal{J}(\mathbf{K})} \left\{ \sum_{i\in I} \mathcal{G}_{i|C_i|}(\mathbf{\Pi}_i, y_i, \emptyset) \middle| (\mathbf{\Pi}_i, y_i) \in \mathcal{N}(\mathbf{x}_i) \right\}$$

and for each itinerary $i \in I$, the revenue function is

$$\mathcal{G}_{ic}(\mathbf{\Pi}_i^{c-1}, \xi, \mathbf{d}_i^{c+1}) = \begin{cases} r_{ic}\mathbb{E}[\min\{\xi - \Pi_{ic-1}, D_{ic}\}|\mathbf{d}_i^{c+1}] & \text{if } \xi \geq \Pi_{ic-1} \\ \quad +\mathbb{E}[\mathcal{G}_{ic-1}(\mathbf{\Pi}_i^{c-2}, \xi - \min\{\xi - \Pi_{ic-1}, D_{ic}\}, \mathbf{d}_i^{c})|\mathbf{d}_i^{c+1}] & \\ \mathbb{E}[\mathcal{G}_{ic-1}(\mathbf{\Pi}_i^{c-2}, \xi, \mathbf{d}_i^{c})|\mathbf{d}_i^{c+1}] & \text{Otherwise.} \end{cases} \tag{8}$$

In general, the revenue function $\mathcal{G}_{ic}(\cdot, \xi, \cdot)$ is not concave in ξ with a reason similar to the example provided in the last paragraph of Section 5.1. However, with the following assumption, it is possible to show that $\mathcal{G}_{ic}(\cdot, \xi, \cdot)$ is in fact concave. For ease of notation, index to itinerary i is ignored from now on.

Assumption A1. *The probability of event E occurs given $Y \geq \phi$ is nondecreasing in ϕ.*

Assumption A1 is the generalized version of the positive regression dependence described in Cooper and Gupta (2006). The intuition can be seen from a two-class example: if fewer seats are allocated to the lower class, the probability of spilling booking requests from the higher class is higher, e.g. $P[d_1 \geq x - \phi | d_2 \geq \phi] \geq P[d_1 \geq \xi - \phi - 1 | d_2 \geq \phi - 1]$, where $\phi = \xi - \Pi_1$. By applying this assumption, we can generalize the results in Brumelle and McGill (1993) with demand dependency only among fare classes. Let $\mathbf{d}^c(E) = \mathbf{d}^{c+1} \cup \{D_c \in E\}$ be the set of observed demands given $\mathbf{d}^{c+1}$ and $\{D_c \in E\}$. We have the following concavity result.

Proposition 5. *Under Assumption A1, and if for all $l \leq c-1$, the protection level Π_l satisfies*

$$r_l \in \left[\delta^+_{x=\Pi_{l-1}} \mathcal{G}_{l-1}(\mathbf{\Pi}^{l-2}, x, \mathbf{d}^l(D_c \geq \xi - \Pi_{l-1})), \delta^-_{x=\Pi_{l-1}} \mathcal{G}_{l-1}(\mathbf{\Pi}^{l-2}, x, \mathbf{d}^l(D_c > \xi - \Pi_{l-1}))\right], \tag{9}$$

then $\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1})$ is concave in ξ.

Proposition 5 shows the concavity of (8) in ξ given the sub-differential condition on the expected revenue function and a set of protection levels that satisfies the condition. This sub-differential condition is similar to the one in Brumelle and McGill (1993), but now depends on the observed demands of the lower classes.

Proposition 6. *Condition (9) is the optimality condition for the protection level Π_{c-1} given any $\mathbf{\Pi}_{c-2}$ that satisfy Proposition 5.*

The proofs of Propositions 5 and 6 can be found in Appendix A.4. Proposition 6 ensures that the set of protection levels that satisfies the sub-differential condition in Proposition 5 is indeed optimal. These two results provide a way to verify the optimality of $\mathbf{\Pi}$ if demand correlation exists, although the protection levels now depend on the observed demands for lower classes.

6 Computational Experiments

In Section 4, we have discussed the ADP algorithm and a set of heuristics that we develop to solve the network RM problem with upsell. In this section, we split our discussion in two parts. The first part focuses on accessing the performance of the upsell heuristic (Algorithm 2), while the second part is devoted to the ADP algorithm (Algorithm 1). The reason to analyze the upsell heuristic separately is that the heuristic by itself is the most important part of the ADP algorithm which returns a set of class-level allocations and a seat margin to update the slope of the approximating function. Hence, its performance is vital to the accuracy of the ADP. The second part establishes the performance of the ADP framework for solving the RM problem at the network level. Both parts include simulation details and output comparisons.

In the first part of this section, the nested allocation policy from the upsell heuristic is benchmarked against the solutions of the algorithm in Wollmer (1992), the path independent choice-based algorithm in Gallego et al. (2009), and the optimal dynamic program with upsell. The algorithm in Wollmer (1992) is chosen because it is developed directly based on the optimality condition of the revenue function (2) assuming independent demand. It allows us to observe the degree of revenue improvement by incorporating upsell. The path independent algorithm in Gallego et al. (2009) is selected simply due to its unparalleled solution quality and efficiency[a]. It relies on a simplified optimality

[a]Although Gallego et al. (2009) has a path dependent algorithm that provides slightly better performance ($< 1\%$ in average), its implementation is more complicated and requires more running time. As our upsell heuristic significantly outperforms the path independent algorithm, we consider the path independent algorithm adequate for our benchmarking purpose.

condition of the revenue function (2) with aggregated demand and upsell. The optimal dynamic program with upsell is used to obtain optimality gaps for the algorithms. It evaluates all possible sets of the protection levels by computing their revenues over a large set of demand samples. In the end, we also investigate the accuracy of the marginal revenue returned by the upsell heuristic and the effect of the projection operation from Line 12 of the ADP algorithm before testing our solution at the network level.

In the second part of this section, we run the ADP algorithm and benchmark its allocation policy against the RLP bid-price policy on a real airline network with 136 flights, 309 itineraries, and 31 reading days. In the absence of a scalable method that simultaneously incorporates nesting, upsell, demand stochasticity, and network information, we select the RLP as it captures demand stochasticity and network information while providing scalability, reliable performances, and theoretical guarantees (see Talluri and Van Ryzin (1999) and Topaloglu (2009)) despite its incapability to capture upsell. We test our proposed allocation policy over various demand magnitudes to access the effect of the network fill rate, and different upsell probabilities to study the sensitivity of the policy to inaccurate probability forecasts.

Let us now present the architecture of the simulation module that we used to evaluate control policies. The flow of the simulation for the first reading day is summarized in Figure 1. Simulations for all other reading days have similar architecture. Mean demands and flight capacities are first queried from the database. While the capacities are loaded directly to the optimization module, the mean demands are randomized and scaled, and used to generate an arrival sample path for the simulation before loading to the optimization module to estimate the required policy. The order of the arrivals between two reading day are randomized before the control policy from the optimization is applied. Note that this violates the low-to-high arrival order assumption of our model, and hence, provides a more realistic simulation environment for a fair comparison to the bid-price policy which is independent to arrival orders. Upsells are then generated based on the rejected arrivals and accumulated with remaining empty seats. In the end, the revenue is computed based on the number of arrivals accepted for each class, and the flight capacities are updated before moving to the next reading day.

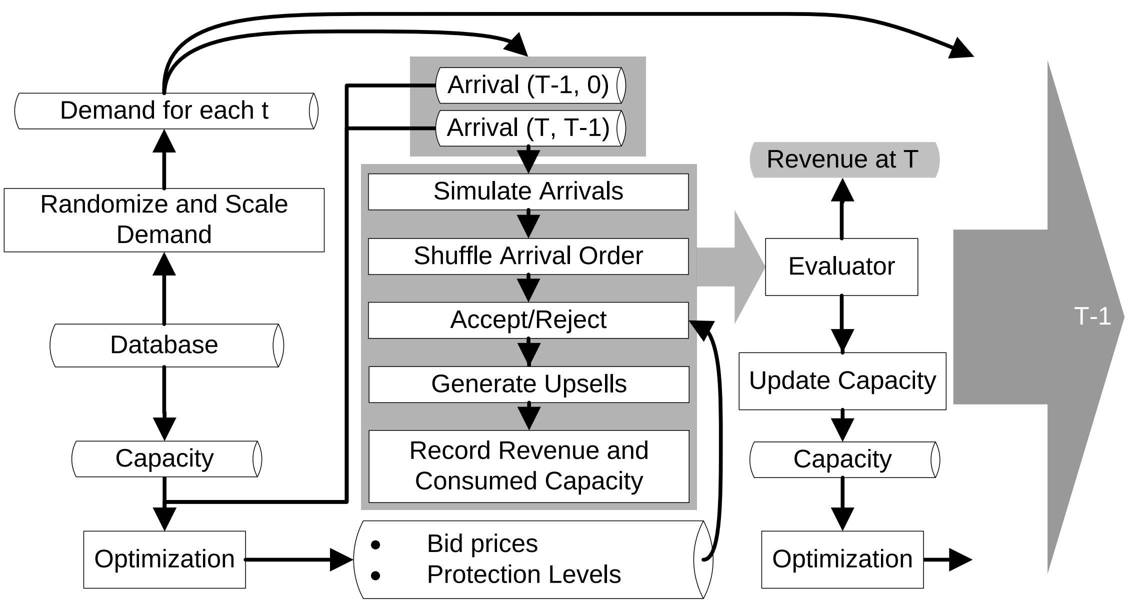


Figure 1: Flow chart of simulation at time T, the first reading day.

Let K^{sim} be the number of demand sample paths generated across the entire booking period for simulation, and K^{opt} be the number of demand samples generated for the optimization problem in question. For all single-leg simulation experiments, we follow the same simulation settings of Gallego et al. (2009) by setting both K^{sim} and K^{opt} to $100,000$ to eliminate the effect of the standard error. Furthermore, all examples in Gallego et al. (2009) as well as some constructed cases are tested. For each itinerary, all algorithms being tested allocate remaining seats, e.g. $\xi - \Pi_{|C_i|-1}$, to the lowest class except for the upsell heuristic which allocates the remaining seats by the upsell-adjusted seat allocation algorithm (Algorithm 5). For all network simulation experiments, due to the large amount of parameters we test, we restrict K^{sim} and K^{opt} to 100 and 50 respectively. Although the number of simulation samples for simulation is relatively small, our results lead to insightful conclusions, and are all valid under a 10% significant level with more than half of them also valid under a 5% level.

All simulation experiments have been run on a cluster of 50 servers. Each server has two 3.2 GHz Intel(R) Xeon(TM) CPUs and 6GB of memory, yet only up to 3GB were used due to our 32 bits limitation. After the allocation for each itinerary is determined (Line 3 of the ADP algorithm), the upsell heuristic is parallelized by itinerary over all available processors. Results are stored in an Oracle 11g database and visualized in IBM Cognos.

6.1 Single Leg Comparison

We access the performance of the upsell heuristic (Algorithm 2) by comparing it with both the algorithm in Wollmer (1992) for independent demand, the path independent choice-based algorithm in Gallego et al. (2009), and the optimal dynamic problem with upsell that evaluates all possible sets of protection levels via simulation. Let us abbreviate the path independent choice-based algorithm in Gallego et al. (2009) by `PaInd`, the algorithm to compute protection levels for independent demand in Wollmer (1992) by `Indep`, the optimal DP with upsell by `DP`, and the upsell heuristic by `Upsell`. All examples in Gallego et al. (2009) as well as some constructed examples are tested.

Let α be the fare or schedule quality multiplier, γ be the margin scaler, λ be the total demand, and 0 be the class to represent the alternative option. To construct the examples, we set the fare for each class to $r_c = \alpha exp(\gamma(|C| - c + 1)/|C|)$ and schedule quality to $s_c = \alpha exp(\gamma(|C| - c + 1)/(2|C|))$. The fare and schedule quality for the alternative option are the averaged fare and schedule quality. These functions are selected in a way that the margin increases with the fare class, and the resulting fare closely matches the real world data given. For all constructed examples, the elasticities of fare and schedule are set to be $\beta_r = -0.0035$ and $\beta_s = 0.005$ respectively. Table 1 shows simulation settings for two/three/four/five-class examples, where the first four cases are taken from Gallego et al. (2009), and the five-class example is constructed based on $\alpha = 100$ and $\gamma = 0.6$. The first row refers to the number of classes in each example. The second row show the elasticities of the schedule and price. The next row is the number of total booking requests.

Table 1: Simulation settings for two/three/four/five-class examples

$\|C\|$	2		3		4		5	
β_r/β_s	-0.005	0.005	-0.0035	0.005	-0.0035	0.005	-0.0035	0.005
λ	26.67		25		50		20	
Class	fare	schedule[b]	fare	schedule	fare	schedule	fare	schedule
0	900	1100	1100	500	1000	600	846.01	268.67
1	1000	1200	1000	200	1000	400	2008.55	448.17
2	800	1000	800	200	900	300	1102.32	332.01
3			500	200	600	300	604.96	245.96
4					500	300	332.01	182.21
5							182.21	134.99

Optimality gap comparisons are summarized in Figure 2. For 2 and 3 classes, both `Upsell` and `PaInd` performs close to `DP` with optimality gaps smaller than 1%, and the solution quality of `Indep` first sharply deteriorated before gradually approaching optimality as the capacity increases. For 4 classes, we can see that `Upsell` starts to outperforms `PaInd` when the number of seats available is higher than 16, and its optimality gap is still less than 1% while the optimality gap for `PaInd` is about 2% in several occasions. For 5 classes, the gaps for both `PaInd` and `Indep` are significantly widened. This illustrates the drawback of estimating optimal protection levels based only on simple probability statements that cannot fully capture the dynamic of upsells.

[b]The quality of the schedule assigned to an itinerary. It reflects how attractive the departure time of the itinerary is to the customer.

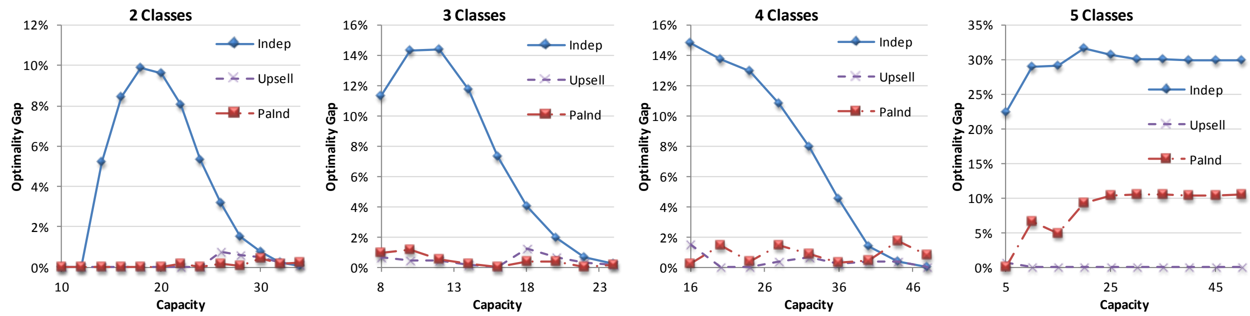


Figure 2: Optimality gap comparisons for two/three/four/five classes.

Figure 3 shows the running time as a portion of the running time of `DP`. It is clear that the relative running time of `Upsell` generally decreases as the number of classes increases. However, it sharply increases in the two-class example after the point where capacity and demand level meet. The reason is that `Upsell` starts with more seats for the highest yielding class (due to the upsell-adjusted seat allocation algorithm) and reallocates the seats one-by-one to lower classes that are more profitable stochastically. This incurs additional overhead and algorithmic operations by storing all information necessary (e.g. α_c^k, β_c^k, γ_c^k in the upsell revenue estimation algorithm) to recompute a revenue margin. It is also due to the fact that running `DP` for a problem with only two classes is relatively inexpensive. Such an increase of running time is gradually diminishing in the number of classes, as the running time of `DP` exponentially increases.

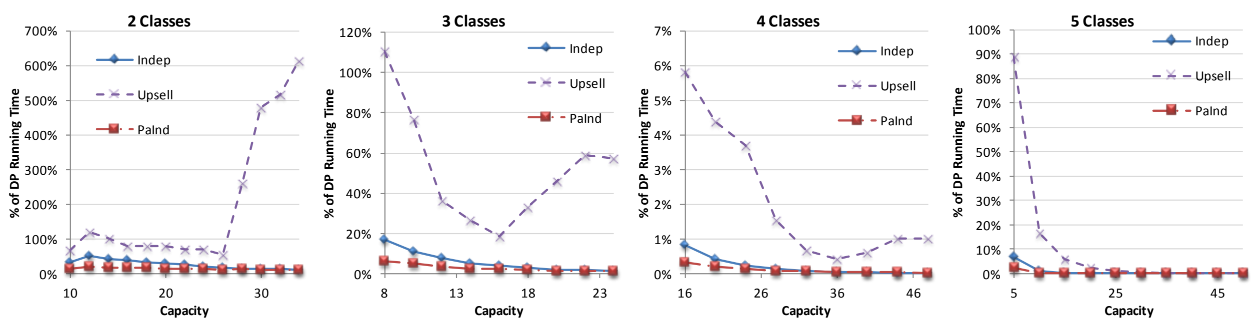


Figure 3: Percentage of the running time of `DP` for two/three/four/five classes from left to right, smaller the better.

In additional to the five cases discussed, extra simulation experiments were conducted for other cases that we constructed based on the method we described in the beginning of Section 6. We tested $\alpha \in \{100, 200, 300\}$, $\gamma \in \{2, 2.5\}$, $C \in \{2, 3, \ldots, 10\}$, $\lambda \in \{10, 20, 30, 40, 50\}$, and capacity in $[5, 10, \ldots, 50]$. Average relative optimality gaps based on `Upsell` are given in Table 2, and the percentages of running time increased are given in Table 3. A negative value means that `Upsell` is outperformed (it happens only once in Table 2 by a small percentage when the number of class is 3, and the demand is low).

In general, `Upsell` outperforms both `Indep` and `PaInd` up to 26% and 24% respectively when the number of classes increases or the number of total bookings decreases, yet the percentage of the running time also increases as the running times for `Indep` and `PaInd` are fairly constant regardless of the number of the fare classes and the magnitude of the demands. Interested reader may refer to Appendix 5 for the average running time of `Upsell` in second and the average demand factor (demand-to-capacity ratio) for different number of classes and total demands.

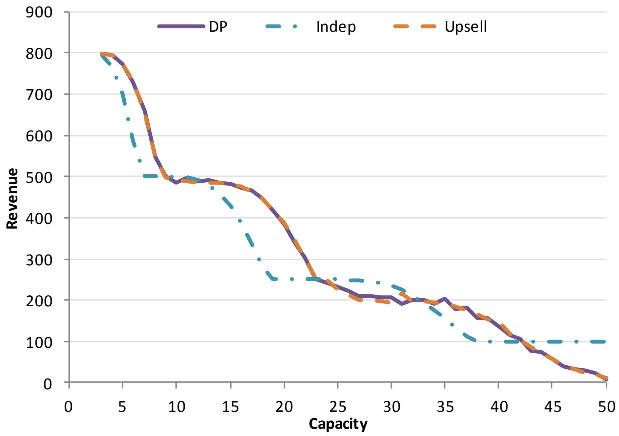
Table 2: Average relative optimality gap based on `Upsell`.

λ	10		20		30		40		50	
$\lvert C\rvert$	`Indep`	`PaInd`	`Indep`	`PaInd`	`Indep`	`PaInd`	`Indep`	`PaInd`	`Indep`	`PaInd`
2.0	1.33%	0.09%	3.90%	0.12%	6.98%	0.17%	9.04%	0.10%	11.79%	0.21%
3.0	22.49%	*-0.21%*	22.15%	0.35%	23.11%	0.80%	23.63%	1.16%	23.63%	1.44%
4.0	21.43%	3.62%	22.42%	3.98%	22.62%	4.36%	23.39%	4.53%	24.67%	4.77%
5.0	20.82%	10.62%	22.48%	10.10%	23.53%	9.32%	24.03%	7.97%	24.94%	6.81%
6.0	20.81%	14.51%	22.95%	13.86%	23.68%	12.30%	25.29%	10.87%	25.81%	9.04%
7.0	21.25%	17.45%	22.82%	16.41%	24.07%	14.50%	25.02%	12.25%	26.06%	9.95%
8.0	20.62%	23.52%	22.47%	21.67%	23.95%	19.49%	25.03%	16.79%	25.95%	13.76%
9.0	20.55%	19.43%	22.53%	18.21%	24.18%	16.24%	24.96%	13.74%	26.02%	11.28%
10.0	20.22%	24.19%	22.38%	22.59%	24.18%	20.18%	25.39%	17.44%	26.02%	14.50%

Table 3: Average percentage of running time based on `Upsell`

λ	10		20		30		40		50	
$\lvert C\rvert$	`Indep`	`PaInd`	`Indep`	`PaInd`	`Indep`	`PaInd`	`Indep`	`PaInd`	`Indep`	`PaInd`
2.0	3.64%	1.77%	5.50%	1.55%	7.78%	2.24%	9.75%	2.68%	12.02%	3.12%
3.0	8.82%	7.03%	6.88%	4.66%	5.51%	2.87%	6.69%	2.88%	7.09%	3.07%
4.0	5.26%	4.65%	4.09%	2.81%	4.26%	1.98%	4.82%	2.24%	6.36%	2.47%
5.0	3.81%	3.30%	3.31%	2.17%	4.03%	1.81%	4.45%	1.78%	4.74%	1.75%
6.0	3.01%	2.70%	2.93%	1.87%	3.29%	1.51%	4.22%	1.54%	4.66%	1.53%
7.0	2.57%	2.50%	2.63%	1.71%	2.97%	1.38%	3.46%	1.29%	4.42%	1.43%
8.0	2.25%	2.23%	2.47%	1.60%	2.90%	1.35%	3.33%	1.24%	4.31%	1.36%
9.0	2.14%	2.28%	2.33%	1.58%	2.61%	1.27%	2.95%	1.14%	3.49%	1.14%
10.0	2.01%	2.19%	2.27%	1.58%	2.57%	1.29%	2.98%	1.16%	3.35%	1.12%

In order to closely approximate (4) with piecewise linear functions and iteratively update their slopes with marginal revenues, we need to accurately estimate marginal revenue at each itinerary allocation level. Figure 4 illustrates how accurate the marginal revenues are estimated by `Upsell` in two examples. In each example, demand for the alternative buying option is two times of the total demand, and the attractiveness of a fare class is set to be the magnitude of its demand. The figure on the left displays marginal revenue curves generated based on `DP`, `Indep`, and `Upsell` for a four-class example with demand in $\{20, 15, 10, 5\}$ and revenue in $\{100, 250, 500, 800\}$. It shows that the marginal revenue curve generated based on `Upsell` collides with that of `DP`, while the margins from `Indep` are significantly different. The figure on the right similarly shows the marginal revenue curves obtained from an example with fifteen classes selected from a real world data set. Its demands are $\{2, 1, 24, 6, 10, 6, 15, 27, 2, 12, 8, 9, 3, 4, 23\}$ and the revenues are $\{19.89, 22.13, 29.49, 29.78, 32.11, 33.78, 44.49, 51.98, 56.34, 62.52, 74.27, 128.85, 135.05, 170.71, 272.26\}$. We did not include `DP` as it is intractable. Instead, we focus on how the projection operation in Line 12 of `Upsell` changes the marginal revenues. We represent the projected marginal revenue curve by `pjUpsell` in the figure. It is clear that the marginal revenue curve has several wedges, e.g. the objective function is not concave. After being projected, the marginal revenue curve becomes monotonic while many of the original margins are preserved. In summary, when the number of classes is small, `Upsell` is accurate in estimating marginal revenues, and the projection operation can be safely applied to expedite the convergence rate of the ADP algorithm without significantly altering the original revenue margins returned by `Upsell`.



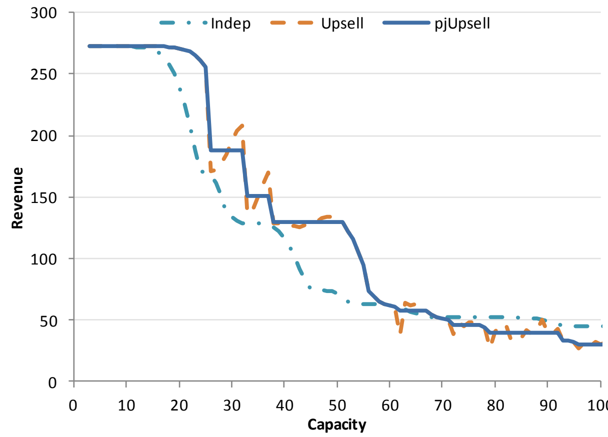


Figure 4: The marginal revenue curves for a four-class example (left) and a fifteen-class example (right).

6.2 Medium-Size Airline Network

In this section, we access the performance of the protection levels returned by the ADP Algorithm using a medium airline network based on a real world data set. Table 4 summarizes the airline network, which has 136 flights, 309 itineraries, 31 reading days, 10.5 classes in average for each itinerary, and 80% capacity filled in average. We first discuss details about the simulation and implementation settings. Then, we conclude this section with our simulation results.

Table 4: Summary of the medium airline network

No. of flights	136	Min. demand factor[a]	3%
No. of itineraries	309	Avg. demand factor	80%
No. of reading days	31	Max. demand factor	240%
Avg. No. of Classes	10.5		

6.2.1 Implementation Details on the ADP Algorithm

To run the ADP Algorithm, we need to initialize the marginal revenues properly and determine appropriate stopping criteria to prevent the algorithm from stalling without significantly trading off optimality. In our implementation, marginal revenues are initialized based on the fare-adjusted seat allocation algorithm (Algorithm 6). The use of the fare-adjusted seat allocation algorithm is solely for efficiency as marginal revenue for each possible itinerary allocation level has to be computed. If the upsell heuristic is used instead, too much running time will be consumed without improving solution quality significantly (due to the fact that initial marginal revenues, if not too far off, do not affect the final solution, and running several more iterations of the ADP algorithm is considerably inexpensive). The ADP Algorithm is stopped if the current revenue is in $[\mu \pm 0.001\sigma]$, where μ and σ are the average revenue and standard deviation computed based on the last 30 revenue points. This stopping criterion guarantees that revenue shifting is unlikely. To expedite the algorithm, we cease learning (updating marginal revenues) for an itinerary if the difference between its revenue from the last iteration and the revenue from the current iteration are less than 1% of its average revenue over the last 10 iterations.

6.2.2 Simulation Settings

To access the performance of the nested allocation policy under multiple demand scenarios, we randomize and scale the mean of the demands. To be more specific, the mean demands are randomized by a normal distribution and scaled by a multiplier. The adjusted mean demand serves as the mean to generate demand samples for both simulation and optimization. The standard derivation is selected to be a multiple of the non-scaled mean. This is designed to merely shift the demand without inflating its variability. Denoting the number of the original demands by $\mathcal{D}_{ict}$, the randomized demand is $D_{ict} \sim Round(\mathcal{N}ormal((1+m_1)\mathcal{D}_{ict}, (\mathcal{D}_{ict}/m_2)^2))$, where $m_1 \in \{-0.4, -0.2, 0, 0.2, 0.4\}$ is the demand multiplier (see Appendix 6 for the corresponding demand factors), and $m_2 = 3$ specifies the scaling factor of the standard deviation in order to match the original mean value, e.g. about 99.7% of the random demands fall into the interval of $[\mathcal{D}_{ict} \pm \mathcal{D}_{ict}]$. We regenerate if the realized demand is negative.

In reality, upsell probabilities are difficult to estimate due to data censorship. It is important to examine how forecasting error on upsell affect the performance of the nested allocation policy. Toward this end, we use two different sets of upsell probabilities, one for simulation, and one for optimization (when upsell information is generated in Algorithm 2). In both sets, the attractiveness of a class is set to be the value of its demand with a scaling parameter to determine the attractiveness of the alternative option. Formally, the upsell probability is computed based on $p^j_{cc'} = m^j_3 \mathbb{E}D_{c'} / \sum_{l=c'}^{c-1} \mathbb{E}D_l$ for $c, c' \in C$, where $j \in \{simulation, optimization\}$ and m^j_3 is the scaling parameter. If $m^j_3 = 0$, no upsell occurs. If $m^j_3 = 1$, no alternative exists. The values of m^j_3 are selected in $[0, 0.1, \ldots, 0.9]$.

We benchmark our allocation policy against the bid-price policy from RLP. The number of arrivals for simulation is 100 across the entire booking period, the number of demand samples generated per ADP iteration is 50, and 50 demand samples are generated for RLP.

6.2.3 Discussions

We first discuss results when demand varies and upsell probabilities are forecasted accurately, i.e. the upsell probabilities in optimization are "accurate". The results are summarized in Figure 5. Each line represents results from a demand multiplier ranging from -0.4 to 0.4. The figure shows in general that when demand increases (the demand multipliers represent the different series), percentage of revenue improvement increases over all scales of upsell probabilities, and are especially prominent in the central region where the upsell probability multiplier is in $[0.3, 0.7]$. The difference can be as much as 5% between the lowest (-0.4) and highest (0.4) demand multipliers when the upsell probability multiplier is 0.6. When the upsell probability multiplier increases, the percentage of revenue improvement increases in a convex manner. Also note that the improvement is close for different demand multipliers, which suggests that the improvement is fairly robust to the magnitude of the demand.

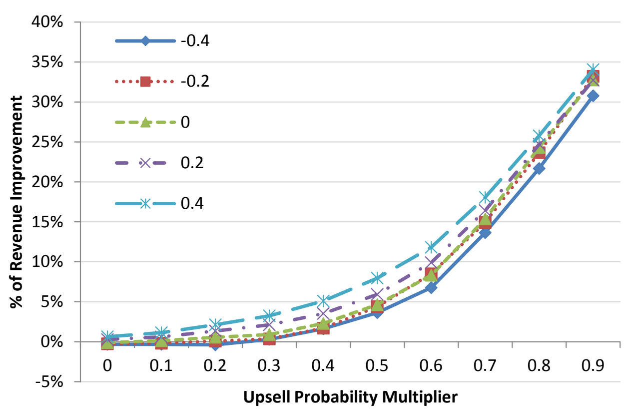


Figure 5: Percentage of revenue improvement against upsell probability multiplier when upsell probabilities are forecasted accurately.

Figure 6 shows the percentage of revenue improvement for all tested combinations of $m_3^{simulation}$ and $m_3^{optimization}$ when demands are not scaled, e.g. $m_1 = 0$. The results are similar for other demand multipliers. See Table 7 in Appendix C for numerical values. The left figure shows results from simulations when $m_3^{simulation}$ is in $\{0, 0.1, 0.2, 0.3, 0.4\}$ with each series corresponding to one value. Overall, we see that each improvement curve slowly inclines until it is at its peak when $m^{simulation} = m^{optimization}$ and gradually declines afterward. Largest improvement is about 2.29% when $m_3^{simulation} = m_3^{optimization} = 0.4$. The declining rate is faster when $m_3^{simulation}$ is small. It also shows that when $m_3^{optimization} \leq m_3^{simulation}$, improvement is almost guaranteed, except for the case when upsell does not exist, e.g. $m_3^{simulation} = 0$. It signifies that it is better to underestimate the upsell probabilities when solving the RM problem. The right figure show simulation results for $m_3^{simulation} \in [0.5, 0.6, 0.7, 0.8, 0.9]$. The situation is just the opposite: overestimating upsell probabilities provides better results, and the revenue improvement can be as high as 33% when 90% chance a rejected customer will upsell and we forecast it accurately. The reason of such an opposite behavior can be contributed to both the cascading effect of the upsells and the heuristic nature of the upsell heuristic. When $m_3^{simulation} \geq 0.5$, every rejected booking is more likely to upsell than opt for alternative options. It results in pushing more low-class demands upward when $m_3^{simulation}$ increases, and the phenomenon is particularly obvious when there exists many classes. Since the effect is accumulated starting from the lowest class, having additional seats for higher classes resulting from overestimating the actual upsell ($m_3^{optimiation} \geq m_3^{simulation}$) is beneficial. On the heuristic side, although the upsell-adjusted seat allocation algorithm intentionally allocates additional seats to higher classes, it still underestimates the number of seats required when there is such an upsell-cascading effect. Also, it is worth to note that the upsell heuristic can only reduces seats from higher classes, and hence, would not be able to push allocations upward and extract benefits from the cascading effect given an initial set of allocations from the upsell-adjusted seat allocation algorithm.

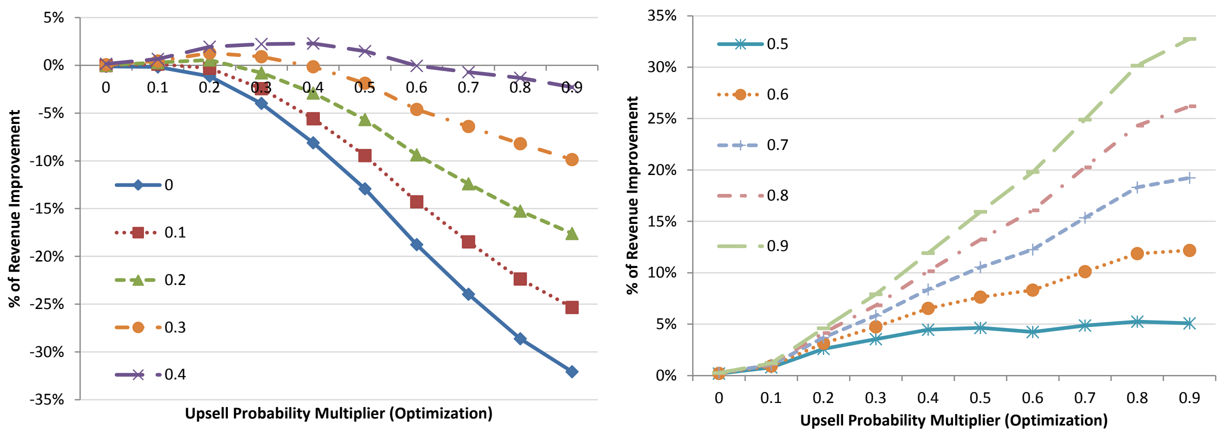


Figure 6: Percentage of revenue improvement against $m_3^{optimization}$ when $m_1 = 0$.

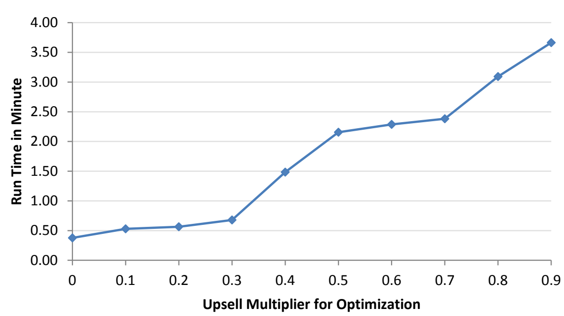


Figure 7: Average running time of the ADP algorithm in minute.

Figure 7 shows the average running time of the ADP algorithm which takes longer to estimate a set of protection levels when the upsell probability for optimization increases. The running time stretches from less than 30 seconds to about 4 minutes. The reason for such an increase in running time is that when more upsells are available, the upsell heuristic needs more enumerations to adjust the seat allocations and to compute the required margins to estimate the benefit from upselling. On the other hand, the running time of the RLP are negligible, and hence, are not reported.

7 Conclusion

Despite the prevalence of the bid-price policy, we aim to capture capacity nesting and customer upsell by extending an itinerary-based nesting model originally by Curry (1990). We show that the itinerary-based nesting model can be rewritten into a stochastic program, which allows us to adapt an ADP framework to efficiently approximate the originally complicated objective function. We also derive an efficient upsell heuristic based on the recursive structure of the problem, and integrate it with into the ADP algorithm to solve the network RM problem over a medium airline network using a real world data set. Furthermore, we show the asymptotic optimality of the nested allocation policy when demands are independent. When demands are correlated, we derive a sufficient condition under which the objective function is concave in the remaining capacity. The result allows us to recursively check for the optimality of the protection levels when demand history is given.

From our single-leg simulation experiments, we observe that the upsell heuristic significantly outperforms the independent algorithm of Wollmer (1992) by 26% and the path independent algorithm of Gallego et al. (2009) by 24% when the number of classes increases. When the number of classes is large, the projection operation does not significantly alter the seat margin, and thus, can be safely applied to expedite the ADP algorithm without losing much accuracy. From our network experiments, we find that the percentage of revenue improvement by using our nested allocation policy, compared against RLP bid-price policy, increases when demand is large and upsell is likely. When upsell probabilities can be forecasted accurately, we do not do worse than the RLP bid-prices when there are no upsells, and can improve the revenue up to 35% ($\sim$ $\$420,000$) when the upsell probability is high. To be more encouraging, the results are robust to demand magnitude, and hence, similar revenue improvement can be expected from a network that is more or less capacitated. When upsell probabilities cannot be forecasted accurately, it is better to underestimate upsell probability when upsell is less likely than opting for the alternative option. Otherwise, overestimating upsell probability is more beneficial. In the end, we also want to stress the practicality of our algorithm by recalling that it only takes 4 minute to finish over a medium airline network by using a dual-core machine.

Several interesting questions remain open: 1). Is there a way to estimate the bid prices based on our ADP algorithm? Currently, the bid prices are itinerary-based and inferior to the RLP bid-prices. 2). Under what conditions should we switch to the bid-price policy. Note that our itinerary-based allocation policy may not work well when the network is heavily intertwined. 3). Can the ADP algorithm be easily extended to capture other customer behaviors including cancellation and no-show?

8 Appendices

A Proofs

A.1 Lemma 1

Proof. Let $\bar{D}_{jt} = \mathbb{E}D_{jt}$ be the expected demand, and $\bar{D}_j = \sum_{t\in T} \bar{D}_{jt}$ the total expected demand. In the case with a single itinerary, consider the deterministic version of (2) and rewrite the function into an optimization problem without protection levels by substituting $\xi - \Pi_{c-1}$ with x_c. We have

$$R_c(\mathbf{\Pi}^{c-1}, \xi) = \max_{0\le x_c\le \xi, \mathbf{x}\in\mathbb{N}} r_c \min\{x_c, \bar{D}_c\} + R_{c-1}(\mathbf{\Pi}^{c-2}, \xi - \min\{y_c, \bar{D}_c\}).$$

It is easy to see that solving the above DP is equivalent to solving

$$\max \left\{ \sum_{c\in C} r_c \min\{x_c, \bar{D}_c\} \middle| \sum_{c\in C} x_c \le \xi \text{ and } x_c \in \mathbb{N} \text{ for } c \in C \right\}.$$

By coupling the itinerary allocation decision with network capacity constraints, we have

$$\max_{\mathbf{y}\in\mathcal{I}(\mathbf{K})} \left\{ \sum_{j\in J} r_j \min\{x_j, \bar{D}_j\} \middle| \sum_{c\in C_i} x_{ic} \le y_i \text{ for } i \in I \text{ and } x_j \in \mathbb{N} \text{ for } j \in J \right\} = \overline{SP^*}(K).$$

$\square$

A.2 Lemma 2

Proof. Let HL be the high-to-low arrival order, LH be the low-to-high arrival order, R be a random arrival order, and $\overline{\mathcal{R}}^{\pi}(O)$ be the revenue obtained by applying the optimal allocation policy of problem π under arrival order O. Then we have $\overline{\mathcal{R}}^{P(\mathbf{K})}(HL) \ge \overline{\mathcal{R}}^{P(\mathbf{K})}(R) \ge \overline{\mathcal{R}}^{P(\mathbf{K})}(LH)$ and $\overline{\mathcal{R}}^{SP(\mathbf{K})}(HL) = \overline{\mathcal{R}}^{SP(\mathbf{K})}(R) = \overline{\mathcal{R}}^{SP(\mathbf{K})}(LH)$. The first observation is based on the fact that seats occupied by low-yield passengers in the LH arrival order can be given to high-yield passengers in both the random and HL arrival orders, and the second observation is based on the fact that since allocations are partitioned, arrival order does not matter. Suppose $\mathbf{x}^\pi$ is the optimal solution of problem π and z^π is the extracted empty seats based on the optimal solution of problem π. We have

$$\begin{aligned}
\overline{\mathcal{R}}^{P(\mathbf{K})}(HL) \geq \overline{\mathcal{R}}^{P(\mathbf{K})}(R) \geq \mathcal{R}^{P(\mathbf{K})}(LH) &= \sum_{i\in I}\sum_{c\in C_i} r_{ic}\min\{x_{ic}^P + z_{ic+1}^P, D_{ic}\} \\
&\geq \sum_{i\in I}\sum_{c\in C_i} r_{ic}\min\{x_{ic}^{SP} + z_{ic+1}^{SP}, D_{ic}\} \\
&\geq \sum_{i\in I}\sum_{c\in C_i} r_{ic}\min\{x_{ic}^{SP}, D_{ic}\} \\
&= \mathcal{R}^{SP(\mathbf{K})}(LH) = \mathcal{R}^{SP(\mathbf{K})}(R) = \mathcal{R}^{SP(\mathbf{K})}(HL).
\end{aligned}$$

Hence, we have $\mathcal{R}^{P(K)}(R) \geq \mathcal{R}^{SP(K)}(R)$, which implies $\mathbb{E}\mathcal{R}^{P(K)} \geq \mathbb{E}\mathcal{R}^{SP(K)}$. The last inequality is due to the fact that $SP(\mathbf{K})$ is optimal under stochastic demand. □

A.3 Proposition 2

Proof. Following (2), it is easy to see

$$R_c(\mathbf{\Pi}^{c-1}, \xi) = \mathbb{E}[r_c\min\{\xi - \Pi_{c-1}, D_c\} + R_{c-1}(\mathbf{\Pi}^{c-2}, \xi - \min\{\xi - \Pi_{c-1}, D_c\})].$$

Note that $\xi - \min\{\xi - \Pi_{c-1}, D_c\} = \Pi_{c-1} + (\xi - \Pi_{c-1} - D_c)^+$. By mapping $\mathcal{P}(\mathbf{\Pi}, \xi)$ and the definition of z_c in (7), we have

$$\begin{aligned}
R_{|C_i|}(\mathbf{\Pi}^{|C_i|-1}, y_i) =& \mathbb{E}\left[r_{|C_i|}\min\{y_i - \Pi_{|C_i|-1}, D_{|C_i|}\} + R_{|C_i|-1}(\mathbf{\Pi}^{|C_i|-2}, \Pi_{|C_i|-1} + z_{|C_i|})\right] \\
=& \mathbb{E}\left[r_{|C_i|}\min\{x_{|C_i|}, D_{|C_i|}\} + r_{|C_i|-1}\min\{\Pi_{|C_i|-1} - \Pi_{|C_i|-2} + z_{|C_i|}, D_{|C_i|-1}\}\right. \\
&\left. + R_{|C_i|-2}(\mathbf{\Pi}^{|C_i|-3}, \Pi_{|C_i|-2} + z_{|C_i|-1})\right] \\
&\vdots \\
=& \mathbb{E}\left[r_{|C_i|}\min\{x_{|C_i|}, D_{|C_i|}\} + \sum_{c'\leq |C_i|-1} r_{c'}\min\{\Pi_{c'} - \Pi_{c'-1} + z_{c'+1}, D_{c'}\}\right] \\
=& \mathbb{E}\left[\sum_{c'\leq |C_i|} r_{c'}\min\{x_{c'} + z_{c'+1}, D_{c'}\}\right] \\
=& \mathbb{E}[\mathcal{S}_i(\mathbf{x}_i, \mathbf{D}_i)]
\end{aligned}$$

For mapping $\mathcal{N}(\mathbf{x})$, we can follow the argument backward. □

A.4 Propositions 5 and 6

Let X and Y be one-dimensional continuous random variables. By definition, $f_{XY}(x,y) = f_{X|Y}(x|y)f_Y(y) = f_{Y|X}(y|x)f_X(x)$ where $f_Y(y) = \int_0^\infty f_{XY}(x,y)dx$ is the marginal density of Y at y. If the cumulative probability function is differentiable, then the probability conditioning on zero probability event can be appropriately defined as

$$\mathbb{P}[X \in E | Y = \phi] = \lim_{\epsilon\to 0} P[X \in E | Y \in [\phi, \phi+\epsilon)] = \frac{\int_{x\in E} f_{XY}(x,\phi)dxdy}{f_Y(\phi)}.$$

This definition can be found in Grimmett and Stirzaker (2001).

Lemma 3. *We have* $\mathbb{P}[X \in E | Y \geq \phi] \geq \mathbb{P}[X \in E | Y = \phi]$ *for any event* E.

Proof. Under Assumption A1, for any $\epsilon > 0$ and continuous random variable Y, if we have

$$\begin{aligned}\mathbb{P}[X \in E|Y \geq \phi + \epsilon] - \mathbb{P}[X \in E|Y \geq \phi] &\geq 0\\ \frac{\mathbb{P}[X \in E, Y \geq \phi] - \mathbb{P}[X \in E, Y \in [\phi, \phi+\epsilon)]}{\mathbb{P}[Y \geq \phi] - \mathbb{P}[Y \in [\phi, \phi+\epsilon)]} - \frac{\mathbb{P}[X \in E, , Y \geq \phi]}{\mathbb{P}[Y \geq \phi]} &\geq 0\\ \mathbb{P}[X \in E, Y \geq \phi]\mathbb{P}[Y \in [\phi, \phi+\epsilon)] - \mathbb{P}[X \in E, Y \in [\phi, \phi+\epsilon)]\mathbb{P}[Y \geq \phi] &\geq 0\\ \frac{\mathbb{P}[X \in E, Y \geq \phi]}{\mathbb{P}[Y \geq \phi]} - \frac{\mathbb{P}[X \in E, Y \in [\phi, \phi+\epsilon)]}{\mathbb{P}[Y \in [\phi, \phi+\epsilon)]} &\geq 0\\ \mathbb{P}[X \in E|Y \geq \phi] - \lim_{\epsilon \to 0} \mathbb{P}[X \in E|Y \in [\phi, \phi+\epsilon)] &\geq 0\\ \mathbb{P}[X \in E|Y \geq \phi] - \mathbb{P}[X \in E|Y = \phi] &\geq 0.\end{aligned}$$

Similarly, if Y is a discrete random variable, the proof is equivalent by substituting ϵ with 1. ☐

Let $\mathbf{D}^{\overleftarrow{c}} = \{D_1, \ldots, D_c\}$ be the set of demands for classes $1, \ldots, c$, $\mathbf{D}^c = \{D_c, \ldots, D_{|C|}\}$ be the set of demands for classes $c+1, \ldots, |C|$, $\mathbf{d}^{\overleftarrow{c}}$ and $\mathbf{d}^c$ be their realizations respectively, $\mathbf{d}^c(D_c \in E) = \mathbf{d}^{c+1} \cup \{D_c \in E\}$ be the updated set of observed events given observed demands $\mathbf{d}^{c+1}$ and event $\{D_c \in E\}$, $f_c(d_c)$ be the marginal density of demand d_c, $f_{c|c+1}(d_c|\mathbf{d}^{c+1})$ be the conditional density of d_c given a set of realized demands $\mathbf{d}^{c+1}$. Given a sample path $\mathbf{d}$, the revenue function is

$$g_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{\overleftarrow{c}}, \mathbf{d}^{c+1}) = \begin{cases} g_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^{\overleftarrow{c-1}}, \mathbf{d}^c) & \text{if } 0 \leq \xi < \Pi_{c-1}\\ r_c(\xi - \Pi_{c-1}) + g_{c-1}(\mathbf{\Pi}^{c-2}, \Pi_{c-1}, \mathbf{d}^{\overleftarrow{c-1}}, \mathbf{d}^c) & \text{if } \Pi_{c-1} \leq \xi < \Pi_{c-1} + d_c\\ r_c d_c + g_{c-1}(\mathbf{\Pi}^{c-2}, \xi - d_c, \mathbf{d}^{\overleftarrow{c-1}}, \mathbf{d}^c) & \text{if } \Pi_{c-1} + d_c \leq \xi \end{cases}$$

and

$$g_1(\emptyset, \xi, \mathbf{d}^{\overleftarrow{1}}, \mathbf{d}^2) = \begin{cases} r_1\xi & \text{if } 0 \leq \xi < d_1\\ r_1 d_1 & \text{if } d_1 \leq \xi. \end{cases}$$

Note that $\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) = \mathbb{E}g_c(\cdot, \xi, \mathbf{D}^{\overleftarrow{c}}, \mathbf{d}^{c+1})$ and $g_c(\cdot, \xi, \mathbf{D}^{\overleftarrow{c}}, \mathbf{d}^{c+1})$ is continuous and piecewise linear in ξ with three break points. For the remaining paragraphs, we assume integral demands. It is then easy to see that $\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1})$ is also continuous and piecewise linear in ξ with countably many breakpoints. To show that $\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1})$ is concave in ξ, it suffices to show that the derivative on the left is no less than the derivative on the right for all possible value of ξ. The corresponding right and left derivatives are showed below:

$$\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) = \begin{cases} r_c\mathbb{E}[\min\{\xi - \Pi_{c-1}, D_c\}|\mathbf{d}^{c+1}] & \text{if } \xi \geq \Pi_{c-1}\\ \quad +\mathbb{E}[\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi - \min\{\xi - \Pi_{c-1}, D_c\}, \mathbf{d}^c)|\mathbf{d}^{c+1}] & \\ \mathbb{E}[\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^c)|\mathbf{d}^{c+1}] & \text{otherwise.} \end{cases}$$

$$\delta_\xi^+\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) = \begin{cases} r_c\mathbb{P}[D_c > \xi - \Pi_{c-1}|\mathbf{d}^{c+1}] & \text{if } \xi \geq \Pi_{c-1}\\ \quad +\sum_{d \leq \xi - \Pi_{c-1}} \delta^+_{x=\xi-d}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c(D_c = d))\mathbb{P}[D_c = d|\mathbf{d}^{c+1}] & \\ \mathbb{E}[\delta^+_{x=\xi}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c)|\mathbf{d}^{c+1}] & \text{otherwise.} \end{cases}$$

$$\delta_\xi^-\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) = \begin{cases} r_c\mathbb{P}[D_c \geq \xi - \Pi_{c-1}|\mathbf{d}^{c+1}] & \text{if } \xi > \Pi_{c-1}\\ \quad +\sum_{d < \xi - \Pi_{c-1}} \delta^-_{x=\xi-d}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c(D_c = d))\mathbb{P}[D_c = d|\mathbf{d}^{c+1}] & \\ \mathbb{E}[\delta^-_{x=\xi}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c)|\mathbf{d}^{c+1}] & \text{otherwise.} \end{cases}$$

First, we show the following properties for the left and right derivatives.

Lemma 4. *If Lemma 3 is satisfied, then*

$$\delta_\xi^+\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^c(D_c \geq \phi)) \geq \delta_\xi^+\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^c(D_c = \phi)).$$

Proof. The proof is by the fact the left derivatives are mainly composed with conditional probabilities. ☐

Lemma 5. *We have*

$$\delta_\xi^+ \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}(D_{c+1} \in E)) = \mathbb{E}[\delta_\xi^+ \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) | D_{c+1} \in E, \mathbf{d}^{c+2}].$$

Proof. The proof is by induction on c. For $c = 1$. we have both sides equal to $r_1 \mathbb{P}[D_1 > \xi | D_2 \in E]$. Suppose the statement is true for $c - 1$, then we have

$$\begin{aligned}
&\mathbb{E}[\delta_\xi^+ \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) | D_{c+1} \in E, \mathbf{d}^{c+2}] \\
&= \frac{\sum_{d \in E} \delta_\xi^+ \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}(D_{c+1} = d)) \mathbb{P}[D_{c+1} = d | \mathbf{d}^{c+2}]}{\mathbb{P}[D_{c+1} \in E | \mathbf{d}^{c+2}]} \\
&= r_c \mathbb{P}[D_c > \xi - \Pi_{c-1} | D_{c+1} \in E, \mathbf{d}^{c+2}] \\
&\quad + \frac{\sum_{d \in E} \sum_{d' \leq \xi - \Pi_{c-1}} \delta^+_{x = \xi - d'} \mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^{c+2} \cup \{D_c = d', D_{c+1} = d\}) \mathbb{P}[D_c = d', D_{c+1} = d | \mathbf{d}^{c+2}]}{\mathbb{P}[D_{c+1} \in E | \mathbf{d}^{c+2}]} \\
&= r_c \mathbb{P}[D_c > \xi - \Pi_{c-1} | D_{c+1} \in E, \mathbf{d}^{c+2}] \\
&\quad + \frac{\sum_{d \in E} \sum_{d' \leq \xi - \Pi_{c-1}} \mathbb{E}\left[\delta^+_{x = \xi - d'} \mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c) | D_c = d', D_{c+1} = d, \mathbf{d}^{c+2})\right] \mathbb{P}[D_c = d', D_{c+1} = d | \mathbf{d}^{c+2}]}{\mathbb{P}[D_{c+1} \in E | \mathbf{d}^{c+2}]} \\
&= r_c \mathbb{P}[D_c > \xi - \Pi_{c-1} | D_{c+1} \in E, \mathbf{d}^{c+2}] \\
&\quad + \sum_{d \leq \xi - \Pi_{c-1}} \delta^+_{x = \xi - d} \mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^{c+2} \cup \{D_c = d, D_{c+1} \in E\}) \mathbb{P}[D_c = d | D_{c+1} \in E, \mathbf{d}^{c+2}] \\
&= \delta_\xi^+ \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}(D_{c+1} \in E)).
\end{aligned}$$

The proof is completed. □

The following is a direct result of Lemma 5.

Corollary 1. *We have*

$$\sum_{d \in E} \delta_\xi^+ \mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^c(D_c = d)) \mathbb{P}[D_c = d | \mathbf{d}^{c+1}] = \delta_\xi^+ \mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^c(D_c \in E)) \mathbb{P}[D_c \in E | \mathbf{d}^{c+1}].$$

Note that Lemmas 4 and 5, and Corollary 1 are also applicable to the right derivatives.

Proposition 7. *Under Assumption A1, and if for all $l \leq c - 1$, the protection level Π_l satisfies*

$$r_l \in \left[\delta^+_{x = \Pi_{l-1}} \mathcal{G}_{l-1}(\mathbf{\Pi}^{l-2}, x, \mathbf{d}^l(D_c \geq \xi - \Pi_{l-1})), \delta^-_{x = \Pi_{l-1}} \mathcal{G}_{l-1}(\mathbf{\Pi}^{l-2}, x, \mathbf{d}^l(D_c > \xi - \Pi_{l-1}))\right], \tag{10}$$

then $\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1})$ is concave in ξ.

Proof. Let $\delta_\xi^2 \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) = \delta_\xi^+ \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) - \delta_\xi^- \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1})$ be the difference between the right and left derivatives. We want to show $\delta_\xi^2 \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) \leq 0$ for all ξ. The proof is by induction on c. For the base case with $\xi > \Pi_1$, we have

$$\begin{aligned}
\delta_\xi^2 \mathcal{G}_2(\mathbf{\Pi}^1, \xi, \mathbf{d}^3) =& \delta_\xi^+ \mathcal{G}_2(\mathbf{\Pi}^1, \xi, \mathbf{d}^3) - \delta_\xi^- \mathcal{G}_2(\mathbf{\Pi}^1, \xi, \mathbf{d}^3) \\
=& \left[\delta^+_{x = \Pi_1} \mathcal{G}_1(\emptyset, x, \mathbf{d}^2(D_2 = \xi - \Pi_1)) - r_2\right] \mathbb{P}[D_2 = \xi - \Pi_1 | \mathbf{d}^3] \\
&+ \sum_{d < \xi - \Pi_1} \delta^2_{x = \xi - d} \mathcal{G}_1(\emptyset, x, \mathbf{d}^2(D_2 = d)) \mathbb{P}[D_2 = d | \mathbf{d}^3] \\
=& \left[r_1 \mathbb{P}[D_1 > \Pi_1 | \mathbf{d}^2(D_2 = \xi - \Pi_1)] - r_2\right] \mathbb{P}[D_2 = \xi - \Pi_1 | \mathbf{d}^3] \\
&- \sum_{d < \xi - \Pi_1} r_1 \mathbb{P}[D_1 = \xi - d | \mathbf{d}^2(D_2 = d)] \mathbb{P}[D_2 = d | \mathbf{d}^3] \\
\leq& \left[r_1 \mathbb{P}[D_1 > \Pi_1 | \mathbf{d}^2(D_2 \geq \xi - \Pi_1)] - r_2\right] \mathbb{P}[D_2 = \xi - \Pi_1 | \mathbf{d}^3]
\end{aligned}$$

$$
\begin{aligned}
&- \sum_{d<\xi-\Pi_1} r_1 \mathbb{P}[D_1 = \xi - d|\mathbf{d}^2(D_2 = d)]\mathbb{P}[D_2 = d|\mathbf{d}^3] \\
&\leq \left[\delta^+_{x=\Pi_1}\mathcal{G}_1(\emptyset, x, \mathbf{d}^2(D_2 \geq \xi - \Pi_1)) - r_2\right]\mathbb{P}[D_2 = \xi - \Pi_1|\mathbf{d}^3] \\
&- \sum_{d<\xi-\Pi_1} r_1 \mathbb{P}[D_1 = \xi - d|\mathbf{d}^2(D_2 = d)]\mathbb{P}[D_2 = d|\mathbf{d}^3] \\
&\leq 0.
\end{aligned}
$$

The inequalities are obtained by applying (9) and Lemma 4. To verify at the break point $\xi = \Pi_1$, we have

$$
\begin{aligned}
\delta^2_\xi \mathcal{G}_2(\mathbf{\Pi}^1, \xi, \mathbf{d}^3) =& r_2\mathbb{P}[D_2 > 0|\mathbf{d}^3] + \delta^+_{x=\Pi_1}\mathcal{G}_1(\emptyset, x, \mathbf{d}^2(D_2 = 0))\mathbb{P}[D_2 = 0|\mathbf{d}^3] \\
&- \sum_{d=0}^{\infty} \delta^-_{x=\Pi_1}\mathcal{G}_1(\emptyset, x, \mathbf{d}^2(D_2 = d))\mathbb{P}[D_2 = d|\mathbf{d}^3] \\
=& r_2\mathbb{P}[D_2 > 0|\mathbf{d}^3] - \sum_{d>0} r_1\mathbb{P}[D_1 \geq \Pi_1, D_2 = d|\mathbf{d}^3] \\
=& (r_2 - r_1\mathbb{P}[D_1 \geq \Pi_1|D_2 > 0, \mathbf{d}^3])\mathbb{P}[D_2 > 0|\mathbf{d}^3] \\
=& (r_2 - \delta^-_{x=\Pi_1}\mathcal{G}_1(\emptyset, x, \mathbf{d}^2(D_2 > 0)))\mathbb{P}[D_2 > 0|\mathbf{d}^3] \\
\leq& 0.
\end{aligned}
$$

Support the induction assumption holds for $c-1$, e.g. $\delta^2_\xi \mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^c) \leq 0$. For $\xi > \Pi_{c-1}$, we have

$$
\begin{aligned}
\delta^2_\xi \mathcal{G}_c(\Pi^{c-1}, \xi, \mathbf{d}^{c+1}) =& \left[\delta^+_{x=\Pi_{c-1}}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c(D_c = \xi - \Pi_{c-1})) - r_c\right]\mathbb{P}[D_c = \xi - \Pi_{c-1}|\mathbf{d}^{c+1}] \\
&+ \sum_{d<\xi-\Pi_{c-1}} \delta^2_{x=\xi-d}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c(D_c = d))\mathbb{P}[D_c = d|\mathbf{d}^{c+1}] \\
\leq& \left[\delta^+_{x=\Pi_{c-1}}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c(D_c \geq \xi - \Pi_{c-1})) - r_c\right]\mathbb{P}[D_c = \xi - \Pi_{c-1}|\mathbf{d}^{c+1}] \\
&+ \sum_{d<\xi-\Pi_{c-1}} \delta^2_{x=\xi-d}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c(D_c = d))\mathbb{P}[D_c = d|\mathbf{d}^{c+1}].
\end{aligned}
$$

The first term of the last inequality is non-positive by Lemma 4, and the second term is non-positive due to our induction assumption. Hence, $\delta^2_\xi \mathcal{G}_c(\Pi^{c-1}, \xi, \mathbf{d}^{c+1}) \leq 0$. At the break point where $\xi = \Pi_{c-1}$, we have

$$
\begin{aligned}
\delta^2_\xi \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) =& r_c\mathbb{P}[D_c > 0|\mathbf{d}^{c+1}] \\
&+ \delta^+_{x=\Pi_{c-1}}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c(D_c = 0))\mathbb{P}[D_c = 0|\mathbf{d}^{c+1}] \\
&- \sum_{d=0}^{\infty} \delta^-_{x=\Pi_{c-1}}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c(D_c = d))\mathbb{P}[D_c = d|\mathbf{d}^{c+1}] \\
=& r_c\mathbb{P}[D_c > 0|\mathbf{d}^{c+1}] - \sum_{d>0} \delta^-_{x=\Pi_{c-1}}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c(D_c = d))\mathbb{P}[D_c = d|\mathbf{d}^{c+1}] \\
\leq& \left[r_c - \delta^-_{x=\Pi_{c-1}}\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, x, \mathbf{d}^c(D_c > 0))\right]\mathbb{P}[D_c > 0|\mathbf{d}^{c+1}] \leq 0.
\end{aligned}
$$

Similarly, the inequalities are obtained by applying (9) and Corollary 1. Hence, the right derivative is no larger than the left derivative, and the continuous piecewise linear function $\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1})$ is concave in ξ when $\mathbf{\Pi}^{c-1}$ satisfies (9). ☐

Lemma 5 states that $\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1})$ is concave in ξ if the allocation policy $\mathbf{\Pi}$ satisfies (9),, which in fact are also the optimality conditions for $\mathbf{\Pi}$.

Proposition 8. *Condition (9) is the optimality condition for the protection level Π_{c-1} given any $\mathbf{\Pi}_{c-2}$ that satisfy Proposition 5.*

Proof. To proof this lemma, we inspect the left and right derivatives of $\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1})$ with respect to Π_{c-1}:

$$\delta_{\Pi_{c-1}^{+}} \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) = \begin{cases} -r_c \mathbb{P}[D_c \geq \xi - \Pi_{c-1}] & \text{if } \xi > \Pi_{c-1} \\ \quad +\delta^{+}_{\xi=\Pi_{c-1}} \mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^{c}(D_c \geq \xi - \Pi_{c-1})) \mathbb{P}[D_c \geq \xi - \Pi_{c-1}] & \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{\Pi_{c-1}^{-}} \mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1}) = \begin{cases} -r_c \mathbb{P}[D_c > \xi - \Pi_{c-1}] & \text{if } \xi \geq \Pi_{c-1} \\ \quad +\delta^{-}_{\xi=\Pi_{c-1}} \mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^{c}(D_c > \xi - \Pi_{c-1})) \mathbb{P}[D_c > \xi - \Pi_{c-1}] & \\ 0 & \text{otherwise.} \end{cases}$$

The left and right derivatives above together with (9) imply that $\delta^{+}_{\xi=\Pi_{c-1}} \mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^{c}(D_c \geq \xi - \Pi_{c-1})) \leq 0 \leq \delta^{-}_{\xi=\Pi_{c-1}} \mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^{c}(D_c > \xi - \Pi_{c-1}))$, and by Lemma 5 that $\mathcal{G}_{c-1}(\mathbf{\Pi}^{c-2}, \xi, \mathbf{d}^{c})$ is concave in ξ, Π_{c-1} that satisfies (9) maximizes $\mathcal{G}_c(\mathbf{\Pi}^{c-1}, \xi, \mathbf{d}^{c+1})$ as required. $\square$

B Algorithms

B.1 Upsell Revenue Estimation Algorithm

The upsell revenue estimation algorithm estimates the upsell revenue given demand samples and class-level partition allocations. It heavily relies on the recursive structure of equations (5) and (6). It takes $\{x_c\}$ the set of class-level partitioned allocations, $\{\zeta_c^k\}$ the set of demand samples, $\{p_{cc}\}$ the set of upsell probabilities, and returns r the average revenue over all demand scenarios along with reject and upsell information encoded by three indicating variables.

Let α_c^k be a binary variable that indicates if a class-c booking is rejected, β_c^k be a binary variable that indicates if an upsell to class c is rejected owning to insufficient allocations, and $\gamma_{cc'}^k$ be a binary variable that indicates if a rejected class-c booking upsells to class c'. These three indicating variables are used to store information for the upsell margin estimation algorithm (Algorithm 4) to compute revenue margins if an additional seat is given to any one of the classes. The algorithm is summarized in Algorithm 3.

Algorithm 3 Upsell revenue estimation algorithm

Require: x_c for $c \in C$, $p_{cc'}$ for $c, c' \in C$, and ζ_c^k for $c \in C$ and $k = 1, \dots K$.

1: Set $r = 0$, $\alpha_c^k = 0$, $\beta_c^k = 0$, and $\gamma_{cc'}^k = 0$ for $c, c' \in C$ and $k = 1, \dots, K$.
2: **for** each demand sample **do**
3:   Initialize $z = 0$, $r' = 0$, and $u_{cc'} = 0$ for $c, c' \in C$.
4:   **for** $c = |C|, \dots, 1$ **do**
5:     $\eta = \sum_{c'=c+1}^{|C|} u_{c'c}$.
6:     Update r' by $r' + r_c \min\{x_c + z, \zeta_c^k + \eta\}$.
7:     **if** $\zeta_c^k > \{x_c + z - \eta\}^+$ **then**
8:       Set $\alpha_c^k = 1$.
9:       **if** c is not the highest class **then**
10:         Generate $\{u_{cc'}\}_{c'=1,\dots,|C|}$ based on $\mathcal{B}(\zeta_c^k - (x_c + z - \eta)^+, \mathbf{p}(c))$.
11:         **for** $c' = 1, \dots, |C|$ **do**
12:           **if** $u_{cc'} > 0$ **then**
13:             Set $\gamma_{cc'} = 1$
14:           **end if**
15:         **end for**.
16:       **end if**
17:     **end if**
18:     **if** $\eta > x_c + z$ **then**
19:       Set $\beta_c^k = 1$.
20:     **end if**
21:     Update z by $\left((x_c + z - \eta)^+ - \zeta_c^k\right)^+$.
22:   **end for**
23:   Update r by $r + r'/K$.
24: **end for**
25: **return** r, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$.

For each demand scenario, the algorithm starts from the lowest class and compute the total upsell to the current class c. Revenue is collected according to objective (4) at Line 6. If there exists a rejected booking in class c, α_c^k is set to one to indicate that there exists a rejected booking. Furthermore, if c is not the highest class, $\gamma_{c\tilde{c}}$ is set to one to record the class that the rejected booking is upselling to. The algorithm then checks if upsells to class c occupy all seats allocated to class c. If it is the case, β_c^k is set to one to record the fact that there exists at least one upsell to class c. The three variables $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ record information required to compute the seat margin in margin estimation algorithm. By recording these rejection and upsell information, we can, instead of computing the finite differences based on the estimated revenue for each class when one more seat is added, reduce the number of algorithmic operations by first computing the base revenue (before a seat is added) and marginally estimating the seat margins for all classes. This reduces the running time significantly when the number of classes is high. In the end, we update the number of empty seats available for higher classes at Line 21 using equation (5).

B.2 Upsell Margin Estimation Algorithm

The upsell margin estimation algorithm computes the revenue margin if one more seat is given to a particular class in question. All necessary information to compute the margin are encoded in arrays $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ (see Algorithm 3 for definitions). It requires the same inputs as those in the upsell revenue estimation algorithm (Algorithm 3) without demand samples. In addition, $\hat{c}$ the class in question is also needed to indicate which class the seat should be added, and the margin should be computed. The algorithm is summarized in Algorithm 4.

Algorithm 4 Upsell margin estimation algorithm

Require: $\hat{c}$, x_c for $c \in C$, $\{\alpha_c^k\}$, $\{\beta_c^k\}$, and $\{\gamma_{cc'}^k\}$.

1: Initialize $m = 0$.
2: **for** each demand scenario **do**
3:   Initialize $m' = 0$.
4:   **if** $\beta_{\hat{c}}^k = 1$ **then**
5:     $m' = r_{\hat{c}}$.
6:   **else**
7:     **for** any higher classes c' (class c inclusive) **do**
8:       **if** $\alpha_{c'}^k = 1$ **then**
9:         Find from $\gamma_{c',\tilde{c}}^k$ over classes in $c' + 1, \ldots, 1$ an upsell resulting from a rejected class-c booking.
10:         **if** an upsell is found at class $\tilde{c}$ **then**
11:           $m' = r_{c'} - r_{\tilde{c}}$.
12:         **else**
13:           $m' = r_{c'}$.
14:         **end if**
15:         **go to** Line 19.
16:       **end if**
17:     **end for**
18:   **end if**
19:   Set $m \leftarrow m + m'/K$.
20: **end for**
21: **return** m

The upsell margin estimation algorithm starts with checking if there exists a rejected upsell from any lower classes to class $\hat{c}$. If an upsell exists, it returns the unit price of class $\hat{c}$ at Line 5. This is due to the fact that if one more seat was given, the rejected upsold booking should have been captured rather than being rejected. If such a rejected upsell does not exist, then for any equal and higher classes $c' = 1, \ldots, \hat{c}$, the algorithm checks both if there exists a rejected booking and if such a rejected booking results in an upsell. If both conditions are satisfied, we need to adjust our margin according to Line 11. The reason is that if one more seat was allocated to class $\hat{c}$, the upsell should have been impossible due to the nesting nature of the allocation policy. Thus, the resulting margin should be non-positive. In the end, if an upsell cannot be found, the margin is set to be the unit price of class c' at Line 13. Although the upsell heuristic (Algorithm 2) can essentially start with allocating all seats to the highest class, the set of allocations returned by this algorithm can significantly reduce the running time of the upsell heuristic.

B.3 Upsell-adjusted Seat Allocation Algorithm

The upsell-adjuste seat allocation algorithm effectively handles cases when spilling demand $\{\sum_{c \in C_i} D_{ic} > X\}$ is likely. It takes the number of total allocated seats, a set of demand samples, and a set of upsell probabilities, and returns a set of class-level partition allocations for the upsell heuristic (Algorithm 2) to further adjust the allocations. Given a set of demand samples and a set of partitioned allocations, the algorithm aggregates demand for higher classes together with the upsells from lower classes. It essentially uses arrival distributions to approximate the upsell demands that are multinomial logit. Once the demands are aggregated, the algorithm treats the aggregated demands as if they are independent and updates the partitioned allocations by the fare-adjusted seat allocation algorithm (Algorithm 6). The process repeats until no more rejection occurs in any demand sample.

Algorithm 5 Upsell-adjusted seat allocation algorithm

Require: y, ζ_c^k for $c \in C$ and $k = 1, \dots, K$, $p_{cc'}$ for $c, c' \in C$.

1: Initialize $\alpha_c^k = 0$ for $c \in C$ and $k = 1, \dots, K$.
2: **loop**
3:   Compute $\mathbf{x}$ using Algorithm 6.
4:   Set $\beta_c^k = 0$ and $\varphi_c^k = 0$ for $c \in C$ and $k = 1, \dots, K$.
5:   **for** $c = |C|, \dots, 1$ **do**
6:     **for** $k = 1, \dots, K$ **do**
7:       $d = \max\{\zeta_c^k - \alpha_c^k, 0\}$.
8:       $z = \max\{x_c - \alpha_c^k, 0\}$.
9:       $\beta_c^k = \max\{d - z - \varphi_{c+1}^k, 0\}$.
10:       $\varphi_c^k = \max\{z + \varphi_{c+1}^k - d, 0\}$.
11:     **end for**
12:   **end for**
13:   Using β, find from the last class, and record the first class $\hat{c}$ with a rejected booking in at least one scenario.
14:   **if** $\hat{c}$ exists and is not the highest class **then**
15:     **for** $k = 1, \dots, K$ **do**
16:       Generate u^k by a multinomial random number generator with p and $\beta_{\hat{c}}^k$.
17:       **for** $c' = \hat{c} - 1, \dots, 1$ **do**
18:         Update $\zeta_{c'}^k$ by $\zeta_{c'}^k + u_{c'}^k$.
19:         Update $\alpha_{c'}^k$ by $\alpha_{c'}^k + u_{c'}^k$.
20:       **end for**
21:       Update $\zeta_{\hat{c}}^k$ by $\zeta_{\hat{c}}^k - \beta_{\hat{c}}^k$.
22:     **end for**
23:   **else**
24:     **return** x.
25:   **end if**
26: **end loop**

The algorithm first initializes $\boldsymbol{\alpha}$ the array that stores accumulated upsells. Once the algorithm enters the infinite loop, it computes the partitioned allocations by the fare-adjusted seat allocation algorithm at Line 3, the associated rejected bookings at Line 9, and the empty seats at Line 10. Next, the algorithm starts from the lowest class and finds $\hat{c}$ the first class with a rejected booking in at least one of the demand samples. Demands to the corresponding upselling classes are then adjusted according to Line 18. The upsells are accumulated and recorded at Line 19. In the end, the number of rejected class-c bookings is subtracted from class $\hat{c}$ demand.

B.4 Fare-adjusted Seat Allocation Algorithm

The fare-adjusted seat allocation algorithm is a heuristic that returns a set of partitioned allocations while accounting some upsell information. It takes y the number of total allocated seats, $\boldsymbol{\zeta}$ a set of demand samples, $\mathbf{p}$ a set of upsell probabilities, $\theta_c = \sum_{c'=1}^{c-1} p_{cc'}$ the probability that a rejected class-c booking ever upsells, and $q_c = \sum_{c'=1}^{c} r_c \mathbb{E} D_c / \sum_{c'=1}^{c} \mathbb{E} D_c$ the average fare over all classes above or equal to class c. It returns a set of partitioned allocations and a set of approximated revenue margins.

The algorithm relies on the derived optimal condition in Curry (1990) and a simple fare-adjusted criterion in Gallego et al. (2009) to efficiently approximate a set of protection levels that accounts for upsell. Additional cares are given to update marginal revenue and to turn the protection levels into partition allocations according to mapping function $\mathcal{P}(\boldsymbol{\Pi}, y)$. As observed in Gallego et al. (2009), the fare-adjusted criterion numerically yields a set of protection levels that tends to reserve more seats to higher classes. This is desirable as the upsell heuristic (Algorithm 2) iteratively and profitably reallocates seats from higher classes to lower classes.

Algorithm 6 Fare-adjusted seat allocation algorithm

```
Require: y, p_{cc'} for c, c' ∈ C_i, ζ_c^k for c ∈ C and k = 1, ..., K, and θ_c and q_c for c ∈ C.
 1: Estimate f_c(s) p.d.f. and F_c(s) c.d.f. of the demands based on {ζ_c^k} for c ∈ C.
 2: Initialize π = 0, x_c = 0 for c ∈ C, S_s = 0 and S'_s = 0 for s = 0, ..., y.
 3: for c = |C|, ..., 1 do
 4:     for s = π, ..., y do
 5:         S'_s = r_c(1 − F_c(s − π)) + Σ_{s'=0}^{x−Π} f_c(s') S_{x−s'}.
 6:     end for
 7:     if c is not the highest class then
 8:         π' = arg min_{s=π,...,y} {(r_{c−1} − θ_{c−1} q_c) ≥ S'_s(1 − θ_{c−1})}.
 9:         S_s = S'_s for s = π, ..., y.
10:         x_c = max{π' − Π, 0}.
11:         π = π'.
12:     else
13:         S_s = S'_s for s = 0, ..., y.
14:     end if
15: end for
16: return x, S.
```

The algorithm first empirically estimates the p.d.f. and c.d.f. of the demands, which are then fed to Line 5 to compute the seat margins based on the optimality conditions valid for independent demands (see Brumelle and McGill (1993)). Once the seat margins are computed, the optimal protection level is determined for the current class based on the adapted fare-adjusted criterion in Gallego et al. (2009) at Line 9. Next, the slope vector is updated, and the corresponding partitioned allocation is computed.

C Tables

This section includes all results used to create the figures in the main document. Table 5 shows the running time of the upsell heuristic (Algorithm 2) for various numbers of classes and demands (λ). In general, the running time increases when the number of classes or demand increases.

Table 5: Average running time for the upsell heuristic and the associated demand factor

	Average Running Time (s)					Average Demand Factor				
$\lvert C\rvert \backslash \lambda$	10	20	30	40	50	10	20	30	40	50
2	21.61	31.93	36.36	37.52	36.95	0.48	0.97	1.45	1.94	2.42
3	8.95	18.28	30.56	42.07	49.94	0.50	1.01	1.51	2.01	2.52
4	18.97	35.26	52.70	68.25	82.66	0.52	1.04	1.55	2.07	2.59
5	32.33	57.19	82.19	106.25	124.61	0.53	1.05	1.58	2.11	2.64
6	51.95	87.38	119.88	150.91	173.12	0.54	1.07	1.61	2.14	2.67
7	69.53	117.82	163.87	203.67	236.21	0.54	1.08	1.62	2.17	2.71
8	99.63	159.36	211.73	259.14	295.22	0.55	1.09	1.64	2.18	2.72
9	112.35	193.95	269.92	332.24	382.36	0.55	1.10	1.65	2.20	2.75
10	142.37	235.57	317.42	390.32	450.86	0.55	1.10	1.66	2.21	2.76

Table 6 shows the minimum, average, and maximum demand factors over all flights for multiple demand multipliers. When the demand multiplier is 0, then the average demand factor is 80% representing that the network is 80% full. When the demand factor is above 100%, the number of bookings are more than the number of seats available.

Table 6: Minimum, average, and maximum demand factors over all flights for different demand multipliers

Demand Multiplier	min	avg	max
-0.4	0.02	0.48	1.18
-0.2	0.03	0.65	2.13
0	0.03	0.8	2.4
0.2	0.04	0.96	2.81
0.4	0.06	1.1	2.97

Table 7 shows the average percentage of improvement in revenue against RLP for different demand multipliers and upsell probability multipliers. The average is taken over all generated demand sample paths, one for each simulation experiment. The percentages showed are the revenue improvement of our proposed allocation policy over the RLP bid-prices.

Table 7: Average percentage of revenue improvement against RLP over all simulations for different demand multipliers and upsell probability multipliers when upsells are forecasted accurately.

Upsell Probability Multiplier\ Demand Multiplier	-0.4	-0.2	0	0.2	0.4
0	-0.33%	-0.26%	-0.11%	0.28%	0.63%
0.1	-0.33%	-0.19%	0.12%	0.60%	1.13%
0.2	-0.39%	0.08%	0.56%	1.34%	2.13%
0.3	0.30%	0.35%	0.90%	2.12%	3.25%
0.4	1.62%	1.69%	2.29%	3.53%	5.09%
0.5	3.64%	4.38%	4.64%	5.92%	7.96%
0.6	6.76%	8.50%	8.31%	9.90%	11.81%
0.7	13.64%	14.92%	15.36%	16.41%	18.06%
0.8	21.67%	23.65%	24.32%	24.70%	25.77%
0.9	30.77%	33.21%	32.75%	32.72%	34.06%

REFERENCES

Belobaba, P. P. and Weatherford, L. R. (1996). Comparing decision rules that incorporate customer diversion in perishable asset revenue management situations. *Decision Sciences Journal*, 27(2):343–363. [cited at p. 3]

Bitran, G. and Caldentey, R. (2003). An overview of pricing models for revenue management. *Manufacturing & Service Operations Management*, 5(3):203 – 229. [cited at p. 1]

Brumelle, S. L. and McGill, J. I. (1993). Airline seat allocation with multiple nested fare classes. *Operations Research*, 41(1):127–137. [cited at p. 2, 3, 9, 11, 28]

Brumelle, S. L., McGill, J. I., Oum, T., Sawaki, K., and Tretheway, M. W. (1990). Allocation of airline seats between stochastically dependent demands. *Transportation Science*, 24(3):183–192. [cited at p. 3, 9]

Chen, L. and Homem-de-Mello, T. (2010). Re-solving stochastic programming models for airline revenue management. *Annals of Operations Research*, 177(1):91–114. [cited at p. 3]

Cooper, W. L. (2002). Asymptotic behavior of an allocation policy for revenue management. *Operations Research*, 50(4):720–727. [cited at p. 2, 9, 10]

Cooper, W. L. and Gupta, D. (2006). Stochastic comparisons in airline revenue management. *Manufacturing & Service Operations Management*, 8(3):221 – 234. [cited at p. 2, 9, 11]

Curry, R. E. (1990). Optimal airline seat allocation with fare classes nested by origins and destinations. *Transportation Science*, 24(3):193–204. [cited at p. 2, 4, 18, 27] de Boer, S. V., Freling, R., and Piersma, N. (2002). Mathematical programming for network revenue management revisited. *European Journal of Operational Research*, 137(1):72 – 92. [cited at p. 1]

Fiig, T., Isler, K., Hopperstad, C., and Belobaba, P. P. (2010). Optimization of mixed fare structures: Theory and applications. *Journal of Revenue and Pricing Management*, 9(1/2). [cited at p. 3]

Gallego, G., Li, L., and Ratliff, R. (2009). Choice-based emsr methods for single-leg revenue management with demand dependencies. *Journal of Revenue and Pricing Management*, 8(2/3):207–240. [cited at p. 2, 3, 5, 11, 12, 13, 19, 27, 28]

Grimmett, G. and Stirzaker, D. (2001). *Probability and Random Processes*. Oxford University Press, 3rd edition. [cited at p. 20]

Higle, J. L. (2007). Bid-price control with origin-destination demand: A stochastic programming approach. *Journal of Revenue and Pricing Management*, 5(4):291–304. [cited at p. 3]

Li, M. Z. F. and Oum, T. H. (2002). A note on the single leg, multifare seat allocation problem. *Transportation Science*, 36(3):349. [cited at p. 9]

Littlewood, K., editor (1972). *Forecasting and control of passenger bookings*, volume 12. The 12th AGIFORS Symposium, Nathanya, Israel. [cited at p. 2, 3]

Liu, Q. and van Ryzin, G. (2008). On the choice-based linear programming model for network revenue management. *Manufacturing Service Operations Management*, 10(2):288–310. [cited at p. 3]

McGill, J. I. and Van Ryzin, G. J. (1999). Revenue management: Research overview and prospects. *Transportation Science*, 33(2):233. [cited at p. 1]

Powell, W., Ruszczynski, A., and Topaloglu, H. (2004). Learning algorithms for separable approximations of discrete stochastic optimization problems. *Mathematics of Operations Research*, 29(4):814–836. [cited at p. 6, 7]

Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley Series in Probability and Statistics. Wiley-Interscience, 1st edition. [cited at p. 4, 7]

Robinson, L. W. (1995). Optimal and approximate control policies for airline booking with sequential nonmonotonic fare classes. *Operations Research*, 43(2):252–263. [cited at p. 4]

Talluri, K. and van Ryzin, G. (1998). An analysis of bid-price controls for network revenue management. *Management Science*, 44(11):1577–1593. [cited at p. 1, 4]

Talluri, K. and Van Ryzin, G. (1999). A randomized linear programming method for computing network bid prices. *Transportation Science*, 33(2):207–216. [cited at p. 4, 12]

Talluri, K. and van Ryzin, G. (2004). *The Theory and Practice of Revenue Management*. International Series in Operations Research & Management Science. Springer, 1st edition. [cited at p. 4]

Topaloglu, H. (2009). On the asymptotic optimality of the randomized linear program for network revenue management. *European Journal of Operational Research*, 197(3):884–896. [cited at p. 12]

van Ryzin, G. J. and McGill, J. I. (2000). Revenue management without forecasting of optimization: An adaptive algorithm for determining airline seat protection levels. *Management Science*, 46(6):760–775. [cited at p. 3]

Williamson, E. L. (1992). *Airline Network Seat Inventory Control: Methodologies and Revenue Impacts*. PhD thesis, Massachusetts Institution of Technology, Cambridge, Massachusetts. [cited at p. 1]

Wollmer, R. D. (1992). An airline seat management model for a single leg route when lower fare classes book first. *Operations Research*, 40(1):26–37. [cited at p. 2, 4, 9, 11, 13, 19]

Zhang, D. and Adelman, D. (2009). An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science*, 43(3):381–394. [cited at p. 3]