Bounding Optimal Expected Revenues for Assortment Optimization under Mixtures of Multinomial Logits

Jacob Feldman
School of Operations Research and Information Engineering,
Cornell University, Ithaca, New York 14853, USA
jbf232@cornell.edu

Huseyin Topaloglu
School of Operations Research and Information Engineering,
Cornell University, Ithaca, New York 14853, USA
topaloglu@orie.cornell.edu
Tel: 1-607-255-0698, Fax: 1-607-255-9129

February 16, 2015

Abstract

We consider assortment problems under a mixture of multinomial logit models. There is a fixed revenue associated with each product. There are multiple customer types. Customers of different types choose according to different multinomial logit models whose parameters depend on the type of the customer. The goal is to find a set of products to offer so as to maximize the expected revenue obtained over all customer types. This assortment problem under the multinomial logit model with multiple customer types is NP-complete. Although there are heuristics to find good assortments, it is difficult to verify the optimality gap of the heuristics. In this paper, motivated by the difficulty of finding optimal solutions and verifying the optimality gap of heuristics, we develop an approach to construct an upper bound on the optimal expected revenue. Our approach can quickly provide upper bounds and these upper bounds can be quite tight. In our computational experiments, over a large set of randomly generated problem instances, the upper bounds provided by our approach deviate from the optimal expected revenues by 0.15% on average and by less than one percent in the worst case. By using our upper bounds, we are able to verify the optimality gaps of a greedy heuristic accurately, even when optimal solutions are not available.
Keywords: Multinomial logit model, assortment optimization, Lagrangian relaxation, retail operations, choice modeling.
Received: October 2013.
Accepted: January 2015 by William L. Cooper after one revision.

Customer choice models are becoming increasingly popular for modeling demand in modern revenue management systems. In particular, traditional models of demand assume that each customer arrives into the system with the intention of purchasing a fixed product. If this product is available for purchase, then the customer makes a purchase. Otherwise, the customer simply leaves the system. However, modern revenue management systems are able to offer a variety of products to customers, possibly by exploiting the availability of online sales channels. Often times, there are multiple offered products that satisfy the needs of a customer, in which case, the customer makes a choice among the offered products. Due to the choice process, the demand for a particular product depends on what other products are offered. Thus, customer choice models emerge as a useful tool for capturing the dependencies between the demands for the offered products.

In this paper, we study assortment problems that capture the customer choice process of the kind mentioned above. In our problem setting, a firm wants to find a set of products to offer to its customers. There is a fixed revenue associated with each product. An arriving customer may be one of multiple customer types. The firm does not known the type of an arriving customer, but it has access to the probability that an arriving customer is of a particular type. Customers choose among the offered products according to the multinomial logit model and customers of different types choose according to different multinomial logit models whose parameters depend on the type of the customer. This choice model is known as the mixture of multinomial logit models. The goal of the firm is to find a set or an assortment of products to offer to its customers so as to maximize the expected revenue obtained from each customer. Bront et al. (2009) show that this assortment problem is NP-complete, give a mixed integer programming formulation to obtain the optimal solution and provide computational experiments that demonstrate that a greedy heuristic performs quite well when compared with the optimal solutions obtained through the mixed integer programming formulation. One shortcoming of using a heuristic is that we use a heuristic simply due to the fact that we cannot obtain the optimal solution and there is no immediate way of being confident that the solution provided by a heuristic is actually a good one. In this paper, motivated by the difficulty of obtaining optimal solutions and evaluating the quality of the solutions provided by a heuristic, we develop a method to obtain upper bounds on the optimal expected revenue in our assortment problem. Thus, we can check the gap between the expected revenue from the solution provided by a heuristic and the upper bound on the optimal expected revenue to assess the optimality gap of the heuristic.

Our method for obtaining an upper bound on the optimal expected revenue has two crucial pieces. First, a natural approach for obtaining an upper bound on the optimal expected revenue is to assume that the firm knows the type of an arriving customer. In this case, we can focus on each customer type one by one and separately find an assortment that maximizes the expected revenue from each customer type. This approach essentially allows us to offer different assortments of products to customers of different types, whereas our assortment problem requires that we find a single assortment to offer to all customer types. Talluri and van Ryzin (2004) show that if we focus on one customer type at a time, then the assortment that maximizes the expected revenue from a single customer type can be obtained efficiently. This idea provides an efficient approach for obtaining an upper bound on the optimal expected revenue, but the upper bound provided by this idea can be quite loose since the assortments that maximize the expected revenues from different customer types can be drastically different from each other. To overcome this shortcoming, we still focus on each customer type one by one, but use penalty parameters to penalize a product that appears in the assortment offered to one customer type but does not appear in the assortment offered to another customer type. In this way, our goal is to synchronize the assortments offered to different customer types. We choose the penalty parameters from a certain set that ensures that we continue obtaining an upper bound on the optimal expected revenue even if we penalize the presence or absence of the products in the assortments offered to different customer types. We show that we can choose a good set of penalty parameters by solving a convex program.

Second, as we focus on each customer type one by one and use penalty parameters to penalize the presence or absence of the products, we obtain assortment problems with a single customer type, but the penalty parameters play the role of a fixed cost for offering a product. Kunnumkal et al. (2009) show that if customers choose according to the multinomial logit model, then the assortment problem with a fixed cost for offering a product is NP-complete, even when there is a single customer type. To deal with this difficulty, we develop a new approximation to the assortment problem with a single customer type and a fixed cost for offering a product. Our approximation is based on the assumption that the probability that a customer leaves without making a purchase can take on values over a prespecified grid. We design the grid so that we continue obtaining an upper bound on the optimal expected revenue. Denser grid points provide a tighter upper bound at the expense of larger computational effort. We give guidelines for choosing a good set of grid points to balance the tightness of the upper bound with the computational effort.

To our knowledge, our approach is a unique practical method to check the quality of solutions in assortment problems under a mixture of multinomial logit models. Computational experiments indicate that our approach can obtain quite tight upper bounds on the optimal expected revenues. We consider a large set of problem instances with large numbers of products so that we cannot obtain the optimal solutions in a reasonable amount of run time. In more than 98% of the problem instances, the upper bounds from our approach are within 0.5% of the optimal expected revenues. On average, the upper bounds from our approach deviate from the optimal expected revenues by 0.15%. In the process, we support the findings of Bront et al. (2009) on large problem instances for which we cannot compute the optimal solutions and demonstrate that the optimality gaps of the greedy heuristic are within a fraction of a percent. Without tight upper bounds on the optimal expected revenues, it would not be possible to obtain such an accurate characterization of the optimality gaps of the greedy heuristic.

There are three papers that particularly motivated us to construct upper bounds when customers choose according to a mixture of multinomial logit models. First, McFadden and Train (2000) show that a mixture of multinomial logit models can approximate any random utility choice model, where a customer associates random utilities with the products, choosing the product with the largest utility. This result holds irrespective of the joint distribution of the random utilities. So, a mixture of multinomial logit models is a powerful choice model and solving assortment problems under this choice model can have direct implications on solving assortment problems under arbitrary random utility choice models. Second, Talluri (2011) considers assortment problems under a mixture of multinomial logit models, but he focuses on a network revenue management setting. The author computes an upper bound on the optimal expected revenue by preallocating the available capacity to different customer types and his approach turns out to be equivalent to assuming that the firm knows the type of an arriving customer, so that the firm can offer different assortments to customers of different types. He does not use any penalty parameters to harmonize the assortments offered to different customer types. In our assortment problems, this approach can yield quite poor upper bounds and we see a need to improve this approach. The gap between the upper bounds provided by this approach and the optimal expected revenues can exceed 14%.

Finally, as mentioned above, Bront et al. (2009) show that the assortment problem under a mixture of multinomial logit models can be formulated as a mixed integer program. They demonstrate that a greedy heuristic performs quite well when compared with the optimal solutions obtained by the mixed integer program. It is difficult to evaluate the optimality gap of the greedy heuristic for large problem instances and a good upper bound on the optimal expected revenue becomes useful in this regard. Furthermore, a tempting approach to obtain an upper bound on the optimal expected revenue is to solve the linear programming relaxation of their mixed integer program, but we establish that the upper bound from this linear programming relaxation can be as poor as focusing on each customer type one by one without using any penalty parameters. In other words, the upper bound from the linear programming relaxation can correspond to the upper bound from the approach in Talluri (2011).

To sum up, we make the following contributions in this paper. 1) We develop a new approach to obtain an upper bound on the optimal expected revenue in assortment problems under a mixture of multinomial logit models. Our approach finds an assortment that maximizes the expected revenue from each customer type, but we use penalty parameters to synchronize the assortments offered to different customer types. This strategy requires solving assortment problems with a single customer type, but with a fixed cost for offering a product. We show how to approximate such assortment problems by assuming that the probability that a customer leaves without making a purchase lies on a prespecified grid. 2) We show how to choose a good set of penalty parameters by solving a convex program. 3) We show how to choose a good set of grid points. Denser grid points yield tighter upper bounds at the expense of larger computational effort, but we show that if we simply use exponential grid points of the form $\{(1+\rho)^{-k+1} : k = 1, 2, \ldots\}$ for some $\rho > 0$, then no other set of grid points, no matter how dense it is, can improve the upper bound by more than a factor of $1 + \rho$. 4) We show that the linear programming relaxation of the mixed integer program given by Bront et al. (2009) can be as loose as the upper bound obtained under the assumption that the firm knows the type of an arriving customer. 5) Our approach for obtaining an upper bound on the optimal expected revenue is flexible enough that we can extend it to the case where there is a constraint on the total space consumption of the offered products or where customers choose according to a mixture of nested logit models. We show how to make such extensions.

The paper is organized as follows. In Section 1, we review the related literature. In Section 2, we formulate the assortment problem under a mixture of multinomial logit models and we present our approach for obtaining an upper bound, which is based on offering different assortments to different customer types, but uses penalty parameters to synchronize the assortments offered to different customer types. In this way, we obtain assortment problems with a single customer type but with a fixed cost for offering a product. In Section 3, we show how to approximate such assortment problems by assuming that the probability of not making a purchase lies on a prespecified grid. In Section 4, we show how to choose a good set of penalty parameters. In Section 5, we show how to choose a good set of grid points. In Section 6, we relate our approach for obtaining upper bounds to a Lagrangian relaxation strategy on an appropriate formulation of our assortment problem. This development requires more notational overhead than the path we follow. So, we defer this development towards the end of the paper. Since our approach can be cast as a Lagrangian relaxation strategy for a nonconvex program, it is difficult to get theoretical tightness guarantees for our upper bounds and there are pathological problem instances that suffer from a large duality gap. In Section 7, we make extensions to the case where there is a constraint on the total space consumption of the offered products or where customers choose according to a mixture of nested logit models. In Section 8, we give computational experiments on both large problem instances and small problem instances with a special structure. In Section 9, we conclude.

1 Literature Review

Our work is related to assortment problems under the multinomial logit model. Gallego et al. (2004) and Talluri and van Ryzin (2004) consider assortment problems under the multinomial logit model with a single customer type and show that the optimal assortment can be obtained efficiently by focusing on assortments that include a certain number of products with the largest revenues. Bront et al. (2009) and Mendez-Diaz et al. (2010) consider the assortment problem under a mixture of multinomial logit models. They show that the problem is NP-complete, give a mixed integer programming formulation of the problem, present valid cuts to tighten this formulation and experiment with a greedy heuristic. Rusmevichientong et al. (2010) consider the assortment problem when there is a constraint on the number of products that can be offered and show that the optimal assortment can be found efficiently when there is a single customer type. Jagabathula et al. (2011) consider simple heuristics for assortment problems and show that these heuristics obtain the optimal assortment under the multinomial logit model with a single customer type. Gallego et al. (2011) and Wang (2013) study assortment problems under the multinomial logit model, where customers become more likely to leave without a purchase when the offered assortment lacks variety. Davis et al. (2013) give linear programming formulations for assortment problems with constraints on the offered assortment, when customers choose according to the multinomial logit model with a single customer type. Rusmevichientong et al. (2013) consider the assortment problem under a mixture of multinomial logit models, show that the problem is NP-complete even when there are two customer types and give performance guarantees for a certain class of assortments. Desir and Goyal (2013) give approximation schemes for various assortment problems. These approximation schemes can get cumbersome when the number of customer types is large.

There is assortment optimization work under other choice models. Davis et al. (2014), Li and Rusmevichientong (2012), Gallego and Topaloglu (2012) and Li et al. (2013) consider assortment problems when customers choose according a nested logit model with a single customer type and show that the problem is tractable. Farias et al. (2013) consider a choice model where each customer arrives with a particular ordering of products in mind and purchases the first product in the ordering that is offered. They focus on estimating the parameters of the choice model in a way consistent with observed sales data. Blanchet et al. (2013) consider a choice model, where if a customer finds that the product he is interested in is not available, then he makes a transition to another product according to a Markov chain and considers purchasing the other product, until he reaches a product that is available or reaches the option of leaving without purchasing anything. The authors show that the assortment problem is tractable under this choice model.

There is related literature on network revenue management models incorporating customer choice behavior. In this setting, an airline sells itinerary products over a flight network. Customers arriving into the system choose among the offered itineraries and the goal is to dynamically adjust the set of available itineraries over time so as to maximize the expected revenue obtained over the selling horizon. A common approach for such network revenue management problems is to formulate deterministic linear programming approximations. Examples of such approximations can be found in Gallego et al. (2004), Liu and van Ryzin (2008), Zhang and Adelman (2009), Kunnumkal and Topaloglu (2008), Meissner et al. (2012), Kunnumkal and Talluri (2012) and Vossen and Zhang (2013). Usually, the decision variables in these approximations correspond to the number of time periods during which a particular subset of itineraries is made available. Since there is one decision variable for each subset of itineraries, the number of decision variables can be large and it is common to solve the approximations by using column generation. The column generation subproblems in this setting precisely correspond to the assortment problem that we consider in this paper when customers choose according to a mixture of multinomial logit models.

Although McFadden and Train (2000) do not focus on solving assortment problems, their work demonstrates that a mixture of multinomial logit models is a powerful choice model as it can accurately approximate any choice model that is based on random utility maximization. Vulcano et al. (2012) consider the problem of estimating the parameters of the multinomial logit model with a single customer type from sales data. Kleywegt and Wang (2013) estimate the parameters of a mixture of multinomial logit models from sales data and they focus on the case where the sets of products offered to customers are not observable.

2 Problem Formulation and Decomposition Approach

We use N to denote the set of possible products that we can offer to customers. The revenue associated with product j is r_j. We use G to denote the set of customer types. The probability that a customer of type g arrives into the system is α^g, where we have $\sum_{g \in G} \alpha^g = 1$. We use the vector $x = \{x_j : i \in N\} \in \{0,1\}^{|N|}$ to capture the set of products that we offer to the customers, where we have $x_j = 1$ if product j is offered, otherwise we have $x_j = 0$. A customer of a certain type makes a choice among the offered products according to the multinomial logit model whose parameters depend on the type of the customer. In particular, a customer of type g associates the preference weight v_j^g with product j. For all customer types, we normalize the preference weight of the no purchase option to one. In this case, if the set of products that we offer to the customers is captured by the vector x, then a customer of type g purchases product j with probability $P_j^g(x) = v_j^g \, x_j / (1 + \sum_{i \in N} v_i^g \, x_i)$. Thus, if the set of products that we offer to the customers is captured by the vector x, then the expected revenue obtained from a customer is given by $\sum_{g \in G} \alpha^g \sum_{j \in N} r_j \, P_j^g(x)$. Noting the definition of $P_j^g(x)$, we can find the set of products that maximizes the expected revenue obtained from a customer by solving the problem

$$Z^* = \max_{x \in \{0,1\}^{|N|}} \left\{ \sum_{g \in G} \alpha^g \, \frac{\sum_{j \in N} r_j \, v_j^g \, x_j}{1 + \sum_{j \in N} v_j^g \, x_j} \right\}. \tag{1}$$

In the problem above, the fraction computes the expected revenue obtained from a customer of type g as a function of the set of products that we offer, whereas the outer sum computes the expected revenue over all customer types. It is likely that obtaining exact solutions to problem (1) is difficult. In particular, Bront et al. (2009) show that the problem above is NP-complete. Motivated by this complexity result, we focus on obtaining an upper bound on the optimal expected revenue Z^*, given by the optimal objective value of problem (1).

A natural approach for obtaining an upper bound on the optimal expected revenue Z^* is to proceed under the assumption that we can offer different sets of products to different customer types, but use penalty parameters to penalize the absence or presence of the products in the assortments offered to different customer types. To pursue this reasoning, we use $\lambda = \{\lambda_j^g : j \in N, \; g \in G\} \in \Re^{|N| \times |G|}$ to denote a vector of penalty parameters. As a function of the penalty parameters, we define $\Pi^g(\lambda)$ as the optimal objective value of the problem

$$\Pi^g(\lambda) \;= \max_{x \in \{0,1\}^{|N|}} \left\{ \frac{\sum_{j \in N} r_j \, v_j^g \, x_j}{1 + \sum_{j \in N} v_j^g \, x_j} - \sum_{j \in N} \lambda_j^g \, x_j \right\}. \tag{2}$$

The problem above finds a set of products to offer so as to maximize the expected profit obtained from a customer of type g, where we generate a revenue of r_j when we sell product j and incur a cost of λ_j^g when we offer product j. The next lemma shows that $\sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$ provides an upper bound on the optimal expected revenue Z^*, as long as the penalty parameters take values in the set $\Lambda = \{\lambda \in \Re^{|N| \times |G|} : \sum_{g \in G} \alpha^g \, \lambda_j^g = 0 \;\; \forall \, j \in N\}$. The proof is rather simple, but we include the proof to explicitly show the necessity of imposing the condition $\lambda \in \Lambda$.

Lemma 1 *For any $\lambda \in \Lambda$, we have $\sum_{g \in G} \alpha^g \, \Pi^g(\lambda) \geq Z^*$.*

Proof. Letting x^* be an optimal solution to problem (1), we observe that x^* is a feasible, but not necessarily an optimal solution to problem (2), in which case, we obtain

$$\sum_{g \in G} \alpha^g \, \Pi^g(\lambda) \geq \sum_{g \in G} \alpha^g \left\{ \frac{\sum_{j \in N} r_j \, v_j^g \, x_j^*}{1 + \sum_{j \in N} v_j^g \, x_j^*} - \sum_{j \in N} \lambda_j^g \, x_j^* \right\}$$

$$= \sum_{g \in G} \alpha^g \frac{\sum_{j \in N} r_j \, v_j^g \, x_j^*}{1 + \sum_{j \in N} v_j^g \, x_j^*} - \sum_{j \in N} \left\{ \sum_{g \in G} \alpha^g \, \lambda_j^g \right\} x_j^* = Z^*,$$

where the last equality follows from the definition of x^* and the fact that the penalty parameters satisfy $\lambda \in \Lambda$ so that we have $\sum_{g \in G} \alpha^g \, \lambda_j^g = 0$ for all $j \in N$. $\square$

The penalty parameters can be positive or negative, where a positive value for λ_j^g discourages offering product j to a customer of type g, whereas a negative value for λ_j^g encourages offering product j to a customer of type g. Since the zero vector $\bar{0} \in \Re^{|N| \times |G|}$ is in Λ, Lemma 1 implies that $\sum_{g \in G} \alpha^g \, \Pi^g(\bar{0})$ provides an upper bound on the optimal expected revenue Z^* and this upper bound corresponds to the one obtained by offering different sets of products to different customer types without using any penalty parameters. In general, using penalty parameters other than zero can potentially yield tighter upper bounds and our computational experiments indicate that the benefits from using penalty parameters other than zero can be substantial.

Noting that we can use $\sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$ for any $\lambda \in \Lambda$ as an upper bound on the optimal expected revenue Z^*, we can try to solve the problem $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$ to obtain the tightest possible upper bound, but solving the last optimization problem is intractable. In particular, computing $\Pi^g(\lambda)$ at any λ requires solving problem (2). Problem (2) maximizes the expected profit from a customer of type g, where we generate a revenue from each product we sell and incur a cost for each product we offer. Kunnumkal et al. (2009) show that such an assortment optimization problem that involves costs for offering the products is NP-complete. To overcome this difficulty, we develop an approximation $\Pi^g(\lambda)$, while maintaining the upper bound provided by Lemma 1.

3 Upper Bound on Optimal Expected Revenue

At the end of the previous section, we propose solving the problem $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$ to obtain the tightest possible upper bound on the optimal expected revenue Z^*, but solving this optimization problem turns out to be intractable. In this section, we develop an approximation $\tilde{\Pi}^g(\cdot)$ to $\Pi^g(\cdot)$. This approximation is tractable to compute and it satisfies $\tilde{\Pi}^g(\lambda) \geq \Pi^g(\lambda)$ for all $\lambda \in \Lambda$. In this case, by Lemma 1, we have $\sum_{g \in G} \alpha^g \, \tilde{\Pi}^g(\lambda) \geq \sum_{g \in G} \alpha^g \, \Pi^g(\lambda) \geq Z^*$ for any $\lambda \in \Lambda$, implying that we can use $\sum_{g \in G} \alpha^g \, \tilde{\Pi}^g(\lambda)$ for any $\lambda \in \Lambda$ as an upper bound on the optimal expected revenue Z^*. In this case, we can solve the problem $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \, \tilde{\Pi}^g(\lambda)$ to obtain the tightest possible upper bound on the optimal expected revenue provided by the approximations

$\{\tilde{\Pi}^g(\cdot) : g \in G\}$. Solving problem $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \, \tilde{\Pi}^g(\lambda)$ turns out to be tractable. To develop an approximation to $\Pi^g(\lambda)$, we note that $1/(1 + \sum_{j \in N} v_j^g \, x_j)$ in problem (2) is the probability that a customer of type g does not purchase anything when the set of offered products is captured by the vector x. We fix the value of this no purchase probability at p and solve the problem

$$\max_{x \in \{0,1\}^{|N|}} \left\{ \sum_{j \in N} p \; r_j \, v_j^g \, x_j - \sum_{j \in N} \lambda_j^g \, x_j \; : \; \frac{1}{1 + \sum_{j \in N} v_j^g \, x_j} = p \right\}$$

for a fixed value of p. In this case, it follows that if we solve the problem above for all values of p in the interval $[0, 1]$ and pick the largest optimal objective value over all values of p, then we obtain the optimal objective value $\Pi^g(\lambda)$ of problem (2). We make a few refinements in this approach. Since the smallest possible value of the no purchase probability for any customer type is $p_{\min} = \min_{g \in G}\{1/(1 + \sum_{j \in N} v_j^g)\}$, we can consider all possible values of p in the interval $[p_{\min}, 1]$, rather than $[0, 1]$. Furthermore, we can replace the equality constraint in the problem above with the corresponding greater than or equal to constraint $1/(1 + \sum_{j \in N} v_j^g \, x_j) \ge p$, since after replacing the equality constraint with the greater than or equal to constraint, if the constraint ends up being loose for any value of p, then we can increase the value of p until we make the constraint tight, which would only increase the objective value of the problem. So, since we want to find the value of p that makes the objective value of the problem above as large as possible, the values of p that render the constraint loose are not relevant to us. Thus, writing the objective function of the problem above as $\sum_{j \in N} (p \; r_j \, v_j^g - \lambda_j^g) \, x_j$ and noting that the constraint $1/(1 + \sum_{j \in N} v_j^g \, x_j) \ge p$ is equivalent to $\sum_{j \in N} v_j^g \, x_j \le 1/p - 1$, the discussion above implies that if we solve the problem

$$\max_{x \in \{0,1\}^{|N|}} \left\{ \sum_{j \in N} (p \; r_j \, v_j^g - \lambda_j^g) \, x_j \; : \; \sum_{j \in N} v_j^g \, x_j \le \frac{1}{p} - 1 \right\} \tag{3}$$

for all values of p in the interval $[p_{\min}, 1]$ and pick the largest optimal objective value over all values of p, then we obtain the optimal objective value $\Pi^g(\lambda)$ of problem (2). To develop an approximation to $\Pi^g(\lambda)$, we focus on the values of p over a set of grid points, while ensuring that our approximation is an upper bound on $\Pi^g(\lambda)$ even though we focus only on the grid points.

To develop an approximation to $\Pi^g(\lambda)$, consider problem (3) for some $p \in [p_{\min}, 1]$. If we replace the value of p in the objective function with a larger value and the value of p in the constraint with a smaller value, then the optimal objective value of problem (3) gets larger. For any $p_L, p_U \in [p_{\min}, 1]$ with $p_L \le p_U$, we define $\Pi^g(\lambda, p_L, p_U)$ as the optimal objective value of the problem

$$\Pi^g(\lambda, p_L, p_U) = \max_{x \in [0,1]^{|N|}} \left\{ \sum_{j \in N} (p_U \, r_j \, v_j^g - \lambda_j^g) \, x_j \; : \; \sum_{j \in N} v_j^g \, x_j \le \frac{1}{p_L} - 1 \right\}. \tag{4}$$

We observe that problem (4) is a continuous knapsack problem, where the capacity of the knapsack is $1/p_L - 1$, the utility of item j is $p_U \, r_j \, v_j^g - \lambda_j^g$ and the space consumption of item j is v_j^g. As mentioned above, for any $p \in [p_L, p_U]$, comparing problems (3) and (4), we observe that the objective function coefficients and the right side of the constraint in problem (4) are larger than those in problem (3). Furthermore, problem (4) does not impose integrality constraints on the decision variables. Thus, the optimal objective value of problem (4) is larger than that of problem (3). To develop an approximation on $\Pi^g(\lambda)$ while making sure that our approximation is an upper bound on $\Pi^g(\lambda)$, we consider an arbitrary set of grid points $\{p^k : k = 1, \ldots, K+1\}$ that satisfy $p_{\min} = p^1 \leq p^2 \leq \ldots \leq p^K \leq p^{K+1} = 1$. Focusing only on this set of grid points, we solve problem (4) for all values of p_L, p_U with $p_L = p^k$ and $p_U = p^{k+1}$ for all $k = 1, \ldots, K$. The next proposition shows that picking the largest optimal objective value of problem (4) over all values of p_L, p_U provides an upper bound on $\Pi^g(\lambda)$.

Proposition 2 *For any $\lambda \in \Lambda$, we have*

$$\max_{k \in \{1,\ldots,K\}} \left\{ \Pi^g(\lambda, p^k, p^{k+1}) \right\} \geq \Pi^g(\lambda).$$

Proof. We fix some $\lambda \in \Lambda$. We show that there exists $k \in \{1, \ldots, K\}$ such that $\Pi^g(\lambda, p^k, p^{k+1}) \geq \Pi^g(\lambda)$ and this inequality establishes the desired result. Letting x^* be an optimal solution to problem (2), we define p^* as $p^* = 1/(1 + \sum_{j \in N} v_j^g \, x_j^*)$ and choose k such that $p^* \in [p^k, p^{k+1}]$. Since $p^* \geq p^k$, we have $\sum_{j \in N} v_j^g \, x_j^* = 1/p^* - 1 \leq 1/p^k - 1$, which implies that the solution x^* is feasible to problem (4), when this problem is solved with $p_L = p^k$ and $p_U = p^{k+1}$. Thus, using the fact that $p^* \leq p^{k+1}$, we obtain $\Pi^g(\lambda) = \sum_{j \in N} p^* \, r_j \, v_j^g \, x_j^* - \sum_{j \in N} \lambda_j^g \, x_j^* \leq \sum_{j \in N} (p^{k+1} \, r_j \, v_j^g - \lambda_j^g) \, x_j^* \leq \Pi^g(\lambda, p^k, p^{k+1})$, where the first inequality is by $p^* \leq p^{k+1}$ and the second inequality is by the fact that the solution x^* is feasible to problem (4) when solved with $p_L = p^k$ and $p_U = p^{k+1}$. $\square$

Proposition 2 implies that if we let $\tilde{\Pi}^g(\lambda) = \max_{k \in \{1,\ldots,K\}} \{\Pi^g(\lambda, p^k, p^{k+1})\}$ and use $\tilde{\Pi}^g(\lambda)$ as an approximation to $\Pi^g(\lambda)$, then this approximation is an upper bound on $\Pi^g(\lambda)$. We note that Proposition 2 holds for any set of grid points $\{p^k : k = 1, \ldots, K+1\}$. In other words, we have $\max_{k \in \{1,\ldots,K\}} \{\Pi^g(\lambda, p^k, p^{k+1})\} \geq \Pi^g(\lambda)$ irrespective of the placement and number of grid points. Also, we observe that computing $\tilde{\Pi}^g(\lambda)$ for any $\lambda \in \Lambda$ requires solving K continuous knapsack problems. Each knapsack problem can be solved by ordering the items according to their utility to space consumption ratios and filling the knapsack starting from the item with the largest utility to space consumption ratio. Therefore, we can compute $\max_{k \in \{1,\ldots,K\}} \{\Pi^g(\lambda, p^k, p^{k+1})\}$ for any $\lambda \in \Lambda$ quickly as long as the number of grid points is not too large. In Section 5, we dwell on the question of how to choose a reasonable set of grid points.

There are two sources of error when we use $\tilde{\Pi}^g(\lambda) = \max_{k \in \{1,\ldots,K\}} \{\Pi^g(\lambda, p^k, p^{k+1})\}$ as an approximation to $\Pi^g(\lambda)$. First, the approximation $\tilde{\Pi}^g(\lambda)$ is obtained by solving problem (4) by using the set of grid points $\{p^k : k = 1, \ldots, K+1\}$, whereas $\Pi^g(\lambda)$ is obtained by solving problem (3) for all $p \in [p_{\min}, 1]$. Intuitively speaking, if the set of grid points is dense, then we expect the discrepancy due to focusing only on the grid points not to be large. This observation also indicates that by choosing a denser set of grid points, we can obtain better approximations to $\Pi^g(\lambda)$. Second, problem (3) imposes integrality constraints on the decision variables, whereas problem (4) does not. Our expectation is that the continuous relaxation of a knapsack problem provides good approximations to the original one and the discrepancy due to relaxing the integrality constraints is not large. It is indeed possible to formulate a continuous knapsack problem whose optimal objective value deviates from the original binary one at most by a factor of two, but the deviation tends to be much less in practice; see Williamson and Shmoys (2011).

4 Choosing Penalty Parameters

At the end of Section 2, we propose solving the problem $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$ to obtain an upper bound on the optimal expected revenue Z^*, but solving this optimization problem is intractable. To overcome this difficulty, we propose using $\max_{k \in \{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$ as an approximation to $\Pi^g(\lambda)$ and solving the problem

$$\min_{\lambda \in \Lambda} \left\{ \sum_{g \in G} \alpha^g \max_{k \in \{1,\ldots,K\}} \left\{ \Pi^g(\lambda, p^k, p^{k+1}) \right\} \right\}. \tag{5}$$

Noting that $\max_{k \in \{1,\ldots,K\}} \{\Pi^g(\lambda, p^k, p^{k+1})\} \geq \Pi^g(\lambda)$ for any $\lambda \in \Lambda$ by Proposition 2 and $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \, \Pi^g(\lambda) \geq Z^*$ by Lemma 1, it follows that the optimal objective value of problem (5) provides an upper bound on the optimal expected revenue Z^*. Also, it is worthwhile to note that our notation in problem (5) suggests that the sets of grid points $\{p^k : k = 1, \ldots, K+1\}$ that we use for different customer types are the same, but it does not have to be the case and we can use different sets of grid points for different customer types. In this section, we show that $\max_{k \in \{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$ is a convex function of λ, in which case, the objective function of the minimization problem in (5) is convex. Furthermore, we show how to obtain subgradients of $\max_{k \in \{1,\ldots,K\}} \Pi^g(\cdot, p^k, p^{k+1})$. Since the condition $\lambda \in \Lambda$ enforces a set of linear constraints on the penalty parameters, these results indicate that we can solve problem (5) by using subgradient search for minimizing a convex function subject to linear constraints; see Ruszczynski (2006).

It is not difficult to see that $\max_{k \in \{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$ is convex in λ. When we view $\Pi^g(\lambda, p^k, p^{k+1})$ as a function of λ, it corresponds to the optimal objective value of the linear program in (4) as a function of its objective function coefficients. Thus, it follows from linear programming theory that $\Pi^g(\lambda, p^k, p^{k+1})$ is convex in λ. Since the pointwise maximum of convex functions is also convex, it follows that $\max_{k \in \{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$ is convex in λ, as desired.

To show how to obtain subgradients of $\max_{k \in \{1,\ldots,K\}} \Pi^g(\cdot, p^k, p^{k+1})$, we let $\tilde{\Pi}^g(\lambda) = \max_{k \in \{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$. To compute a subgradient of $\tilde{\Pi}^g(\cdot)$ at some $\hat{\lambda} \in \Re^{|N| \times |G|}$, we solve problem (4) with $\lambda = \hat{\lambda}$ and $p_L = p^k$, $p_U = p^{k+1}$ for all $k = 1, \ldots, K$. We let $k^* \in \{1, \ldots, K\}$ be such that we obtain the largest optimal objective value for problem (4) when we solve this problem with $p_L = p^{k^*}$ and $p_U = p^{k^*+1}$. In other words, we have $\tilde{\Pi}^g(\hat{\lambda}) = \Pi^g(\hat{\lambda}, p^{k^*}, p^{k^*+1})$. Furthermore, we let x^* be an optimal solution to problem (4) when we solve this problem with $\lambda = \hat{\lambda}$, $p_L = p^{k^*}$ and $p_U = p^{k^*+1}$, in which case, we get $\sum_{j \in N} (p^{k^*+1} \, r_j \, v_j^g - \hat{\lambda}_j^g) \, x_j^* = \Pi^g(\hat{\lambda}, p^{k^*}, p^{k^*+1}) = \tilde{\Pi}^g(\hat{\lambda})$ as well. On the other hand, at any arbitrary λ, we have $\tilde{\Pi}^g(\lambda) \geq \Pi^g(\lambda, p^{k^*}, p^{k^*+1})$ by the definition of $\tilde{\Pi}^g(\cdot)$. Also, when we solve problem (4) with an arbitrary value of λ but with $p_L = p^{k^*}$ and

$p_U = p^{k^*+1}$, the solution x^* is feasible but not necessarily optimal to problem (4) and we obtain $\sum_{j \in N}(p^{k^*+1}\, r_j\, v_j^g - \lambda_j^g)\, x_j^* \le \Pi^g(\lambda, p^{k^*}, p^{k^*+1}) \le \tilde{\Pi}^g(\lambda)$. If we subtract this chain of inequalities from the equality $\sum_{j \in N}(p^{k^*+1}\, r_j\, v_j^g - \hat{\lambda}_j^g)\, x_j^* = \tilde{\Pi}^g(\hat{\lambda})$ obtained above, then we get

$$\tilde{\Pi}^g(\lambda) \ge \tilde{\Pi}^g(\hat{\lambda}) - \sum_{j \in N} x_j^*\, (\lambda_j^g - \hat{\lambda}_j^g).$$

The expression above indicates that $\tilde{\Pi}^g(\cdot)$ satisfies the subgradient inequality at the point $\hat{\lambda}$ with the subgradient $D(\hat{\lambda}) = \{D_j^c(\hat{\lambda}) : j \in N,\ c \in G\} \in \Re^{|N| \times |G|}$ given by $D_j^c(\hat{\lambda}) = -x_j^*$ if $c = g$ and $D_j^c(\hat{\lambda}) = 0$ if $c \in G \setminus \{g\}$. To sum up, if we want to compute a subgradient of $\tilde{\Pi}^g(\cdot)$ at the point $\hat{\lambda}$, then we solve problem (4) with $\lambda = \hat{\lambda}$ and $p_L = p^k$, $p_U = p^{k+1}$ for all $k = 1, \ldots, K$. We let $k^* \in \{1, \ldots, K\}$ be such that we obtain the largest optimal objective value for problem (4) when we solve this problem with $p_L = p^{k^*}$ and $p_U = p^{k^*+1}$. Finally, using x^* to denote an optimal solution to problem (4) when this problem is solved with $\lambda = \hat{\lambda}$, $p_L = p^{k^*}$ and $p_U = p^{k^*+1}$, $D(\hat{\lambda})$ as defined above provides a subgradient of $\tilde{\Pi}^g(\cdot)$ at $\hat{\lambda}$.

5 Effective Grid Points

The optimal objective value of problem (5) provides an upper bound on the optimal expected revenue Z^* for any choice of the grid points $\{p^k : k = 1, \ldots, K+1\}$. By the discussion that follows Proposition 2, we can obtain tighter upper bounds by using a denser set of grid points, but the computational effort to solve problem (5) increases with a denser set of grid points. To provide some guideline into the choice of the grid points, in this section, we explore the performance of exponential grid points. In particular, for fixed $\rho > 0$, we focus on the set of exponential grid points $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$, where K is large enough that $(1+\rho)^{-K} \le p_{\min} < (1+\rho)^{-K+1}$, in which case, these grid points cover the interval $[p_{\min}, 1]$.

In this section, we show that if we compute an upper bound on the optimal expected revenue Z^* by using the set of exponential grid points $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$ in problem (5), then no other set of grid points, irrespective of how dense the set of grid points is, can improve this upper bound by more than a factor of $1+\rho$. In other words, if we use $\bar{Z}_\rho^{\mathsf{exp}}$ to denote the optimal objective value of problem (5) when we use the set of exponential grid points $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$ in this problem and $\bar{Z}^{\mathsf{arb}}$ to denote the optimal objective value of problem (5) with any arbitrary set of grid points, then it always holds that $\bar{Z}_\rho^{\mathsf{exp}} \le (1+\rho)\, \bar{Z}^{\mathsf{arb}}$. Therefore, when we use the set of exponential grid points to obtain an upper bound, we can a priori be sure that it is not possible to improve this upper bound by more than a factor of $1+\rho$ by using a denser set of grid points. This result, in a sense, gives a performance guarantee for the set of exponential grid points. Furthermore, since the set of exponential grid points is denser to the left side of the interval $[p_{\min}, 1]$ and less dense to the right, this result builds the intuition that it is beneficial to use denser grid points when approximating smaller values of the no purchase probability. The next proposition becomes useful when showing the effectiveness of exponential grid points.

Proposition 3 *Let* $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$ *be a set of exponential grid points for some* $\rho > 0$ *and* $\{p^l : l = 1, \ldots, L+1\}$ *be an arbitrary set of grid points with* $p^1 \leq p^2 \leq \ldots \leq p^{L+1}$*, both covering the interval* $[p_{\min}, 1]$*. For any* $g \in G$ *and* $\lambda \in \Lambda$*, we have*

$$\max_{k \in \{1,\ldots,K\}} \Big\{ \Pi^g((1+\rho)\,\lambda, (1+\rho)^{-k}, (1+\rho)^{-k+1}) \Big\} \leq (1+\rho) \max_{l \in \{1,\ldots,L\}} \Big\{ \Pi^g(\lambda, p^l, p^{l+1}) \Big\}. \tag{6}$$

Proof. We let $k^* \in \{1, \ldots, K\}$ be the value of k that attains the maximum on the left side of the inequality in (6). Also, we let $l^* \in \{1, \ldots, L\}$ be such that $p^{l^*} \leq (1+\rho)^{-k^*} < p^{l^*+1}$. Finally, we let x^* be an optimal solution to problem (4) when we solve this problem after replacing λ with $(1+\rho)\,\lambda$ and with $p_L = (1+\rho)^{-k^*}$, $p_U = (1+\rho)^{-k^*+1}$. Since $p^{l^*} \leq (1+\rho)^{-k^*}$, we have $1/p^{l^*} - 1 \geq 1/(1+\rho)^{-k^*} - 1$, implying that x^* is a feasible solution to problem (4) when we solve this problem with $p_L = p^{l^*}$, $p_U = p^{l^*+1}$. Using the definition of x^*, we have

$$\begin{aligned}
\Pi^g((1+\rho)\,\lambda, (1+\rho)^{-k^*}, (1+\rho)^{-k^*+1}) &= \sum_{j \in N} ((1+\rho)^{-k^*+1}\, r_j\, v_j^g - (1+\rho)\,\lambda_j^g)\, x_j^* \\
\leq (1+\rho) \sum_{j \in N} (p^{l^*+1}\, r_j\, v_j^g - \lambda_j^g)\, x_j^* &\leq (1+\rho)\,\Pi^g(\lambda, p^{l^*}, p^{l^*+1}) \leq (1+\rho) \max_{l \in \{1,\ldots,L\}} \Big\{ \Pi^g(\lambda, p^l, p^{l+1}) \Big\},
\end{aligned}$$

where the first inequality follows by $(1+\rho)^{-k^*} < p^{l^*+1}$ and the second inequality holds since x^* is a feasible, but not necessarily an optimal solution to problem (4) when this problem is solved with $p_L = p^{l^*}$, $p_U = p^{l^*+1}$. By the definition of k^*, the first expression in the chain of inequalities above is equal to the expression on the left side of (6) and the desired result follows. $\square$

The inequality in (6) holds for any $g \in G$ and $\lambda \in \Lambda$, in which case, multiplying this inequality by α^g, adding over all $g \in G$ and taking the minimum of both sides over all $\lambda \in \Lambda$, we get

$$\min_{\lambda \in \Lambda} \left\{ \sum_{g \in G} \alpha^g \max_{k \in \{1,\ldots,K\}} \Big\{ \Pi^g((1+\rho)\,\lambda, (1+\rho)^{-k}, (1+\rho)^{-k+1}) \Big\} \right\} \leq \\ (1+\rho) \min_{\lambda \in \Lambda} \left\{ \sum_{g \in G} \alpha^g \max_{l \in \{1,\ldots,L\}} \Big\{ \Pi^g(\lambda, p^l, p^{l+1}) \Big\} \right\}.$$

By the definition of Λ, we have $\lambda \in \Lambda$ if and only if $(1+\rho)\,\lambda \in \Lambda$. So, the constraint in the minimization problem on the left side above can be written as $(1+\rho)\,\lambda \in \Lambda$. Thus, replacing all occurrences of $(1+\rho)\,\lambda$ with λ through change of variables, we write the inequality above as

$$\min_{\lambda \in \Lambda} \left\{ \sum_{g \in G} \alpha^g \max_{k \in \{1,\ldots,K\}} \Big\{ \Pi^g(\lambda, (1+\rho)^{-k}, (1+\rho)^{-k+1}) \Big\} \right\} \leq \\ (1+\rho) \min_{\lambda \in \Lambda} \left\{ \sum_{g \in G} \alpha^g \max_{l \in \{1,\ldots,L\}} \Big\{ \Pi^g(\lambda, p^l, p^{l+1}) \Big\} \right\}.$$

We observe that the expression on the left side of the inequality above is the optimal objective value of problem (5) when we use the set of exponential grid points $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$

in this problem, whereas the expression on the right side is the optimal objective value of problem (5) when we use an arbitrary set of grid points $\{p^l : l = 1, \ldots, L+1\}$. Therefore, the inequality above shows that the upper bound on the optimal expected revenue obtained by using an arbitrary set of grid points $\{p^l : l = 1, \ldots, L+1\}$ in problem (5) cannot improve the upper bound obtained by using the set of exponential grid points $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$ by more than a factor of $1+\rho$, which is the desired result.

Since $p_{\min} < (1+\rho)^{-K+1}$, we have $K = O(\log(p_{\min})/\log(1+\rho))$. For example, if the no purchase probability of a customer type is no smaller than 0.01 when we offer all of the products, then we can set $p_{\min} = 0.01$. If we want a performance guarantee of 0.1% by using a set of exponential grid points, then we can choose $\rho = 0.001$, in which case, K comes out to be about 4600. When we use this set of exponential grid points, no other set of grid points can improve the upper bound provided by the optimal objective value of problem (5) by more than 0.1%.

6 Connection to Lagrangian Relaxation

Noting that $\Pi^g(\lambda)$ is the optimal objective value of problem (2), Lemma 1 indicates that if we have $\lambda \in \Lambda$, then $\sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$ provides an upper bound on the optimal expected revenue Z^*. In this section, our goal is to show that this result can be motivated by using Lagrangian relaxation on an appropriate reformulation of problem (1). For this purpose, we define the decision variable x_j^g such that $x_j^g = 1$ if we offer product j to a customer of type g, otherwise we have $x_j^g = 0$. In this case, we choose an arbitrary customer type ϕ and write problem (1) equivalently as

$$\begin{aligned} Z^* = \max \quad & \sum_{g \in G} \alpha^g \frac{\sum_{j \in N} r_j \, v_j^g \, x_j^g}{1 + \sum_{j \in N} v_j^g \, x_j^g} \\ \text{subject to} \quad & x_j^g = x_j^\phi \qquad \forall\, j \in N, \;\; g \in G \setminus \{\phi\} \\ & x_j^g \in \{0,1\} \qquad \forall\, j \in N, \;\; g \in G. \end{aligned} \tag{7}$$

By the constraints above, we can replace the decision variables $\{x_j^g : g \in G\}$ with a single decision variable x_j^ϕ, in which case, the problem above becomes equivalent to problem (1). Relaxing the constraints in problem (7) by associating the Lagrange multipliers $\{\alpha^g \lambda_j^g : j \in N, \; g \in G \setminus \{\phi\}\}$ with them, the objective function of the problem above can be written as

$$\sum_{g \in G \setminus \{\phi\}} \alpha^g \left\{ \frac{\sum_{j \in N} r_j \, v_j^g \, x_j^g}{1 + \sum_{j \in N} v_j^g \, x_j^g} - \sum_{j \in N} \lambda_j^g \, x_j^g \right\} + \alpha^\phi \left\{ \frac{\sum_{j \in N} r_j \, v_j^\phi \, x_j^\phi}{1 + \sum_{j \in N} v_j^\phi \, x_j^\phi} + \sum_{j \in N} \Big[\sum_{g \in G \setminus \{\phi\}} \frac{\alpha^g \, \lambda_j^g}{\alpha^\phi} \Big] x_j^\phi \right\}. \tag{8}$$

We use $\{\alpha^g : g \in G \setminus \{\phi\}\}$ to scale the Lagrange multipliers $\{\alpha^g \lambda_j^g : j \in N, \; g \in G \setminus \{\phi\}\}$, as this scaling ultimately allows us to draw parallels with our earlier development more easily. This scaling is not a concern since if $\alpha^g = 0$ for some customer type g, then we can drop this customer type from consideration. If we define the additional Lagrange multipliers $\{\lambda_j^\phi : j \in N\}$ for the customer type ϕ as $\lambda_j^\phi = -\sum_{g \in G \setminus \{\phi\}} \alpha^g \, \lambda_j^g / \alpha^\phi$ for all $j \in N$, then the coefficient of the decision variable x_j^ϕ in the last sum in (8) is $-\lambda_j^\phi$. Also, noting that $\alpha^\phi \, \lambda_j^\phi = -\sum_{g \in G \setminus \{\phi\}} \alpha^g \, \lambda_j^g$, we have $\sum_{g \in G} \alpha^g \, \lambda_j^g = 0$ for all $j \in N$, which implies that $\lambda = \{\lambda_j^g : j \in N,\ g \in G\}$ satisfies $\lambda \in \Lambda$. In this case, noting that the coefficient of the decision variable x_j^ϕ in the last sum in (8) is $-\lambda_j^\phi$, we can write (8) as

$$\sum_{g \in G} \alpha^g \left\{ \frac{\sum_{j \in N} r_j\, v_j^g\, x_j^g}{1 + \sum_{j \in N} v_j^g\, x_j^g} - \sum_{j \in N} \lambda_j^g\, x_j^g \right\}.$$

The discussion so far shows that relaxing the constraints in problem (7) by associating the Lagrange multipliers $\{\alpha^g\, \lambda_j^g : j \in N,\ g \in G \setminus \{\phi\}\}$ with them is equivalent to solving the problem

$$\begin{aligned} \max \quad & \sum_{g \in G} \alpha^g \left\{ \frac{\sum_{j \in N} r_j\, v_j^g\, x_j^g}{1 + \sum_{j \in N} v_j^g\, x_j^g} - \sum_{j \in N} \lambda_j^g\, x_j^g \right\} \\ \text{subject to} \quad & x_j^g \in \{0,1\} \qquad \forall\, j \in N,\ g \in G, \end{aligned} \tag{9}$$

as long as $\lambda \in \Lambda$. Noting that problem (9) is obtained by relaxing the constraints in problem (7) by associating the Lagrange multipliers $\{\alpha^g\, \lambda_j^g : j \in N,\ g \in G \setminus \{\phi\}\}$ with them, it is straightforward to show that the optimal objective value of the problem above provides an upper bound on the optimal objective value of problem (7), which is Z^*. We observe that problem (9) decomposes by customer types and noting the definition of $\Pi^g(\lambda)$ in (2), the optimal objective value of problem (9) is $\sum_{g \in G} \alpha^g\, \Pi^g(\lambda)$. Therefore, it follows that $\sum_{g \in G} \alpha^g\, \Pi^g(\lambda)$ provides an upper bound on the optimal expected revenue Z^* as long as λ satisfies $\lambda \in \Lambda$. This result corresponds to the one given in Lemma 1, but as we show in this section, it is possible to reach this result by using Lagrangian relaxation on an appropriate reformulation of problem (1).

7 Extensions to Constrained Problems and Other Choice Models

In this section, we discuss two extensions of our approach. First, we consider the case where each product occupies a certain amount of space and the total space consumption of the offered products cannot exceed a certain space limit. Second, we consider the case where customers choose according to a mixture of nested logit models, rather than a mixture of multinomial logit models.

7.1 Space Constraint

To extend our approach to the case where there is a space constraint, we use w_j to denote the space consumption of product j. Letting c be the total amount of space available for the offered products, we want to solve the problem

$$Z^* = \max_{x \in \{0,1\}^{|N|}} \left\{ \sum_{g \in G} \alpha^g \frac{\sum_{j \in N} r_j\, v_j^g\, x_j}{1 + \sum_{j \in N} v_j^g\, x_j} \ :\ \sum_{j \in N} w_j\, x_j \le c \right\}, \tag{10}$$

which maximizes the expected revenue obtained from a customer while making sure that the total space consumption of the offered products does not exceed c. Thus, this problem is the analogue of problem (1) under a space constraint. If we have $w_j = 1$ for all $j \in N$, then problem (10) simply limits the total number of offered products to c. Following the same argument in Section 2, we use the penalty parameters $\lambda = \{\lambda_j^g : j \in N,\ g \in G\} \in \Re^{|N|\times|G|}$ to penalize the absence or presence of the products in the assortments offered to different customer types and define $\Pi^g(\lambda)$ as the optimal objective value of the problem

$$\Pi^g(\lambda) \ = \max_{x\in\{0,1\}^{|N|}} \left\{ \frac{\sum_{j\in N} r_j\, v_j^g\, x_j}{1+\sum_{j\in N} v_j^g\, x_j} - \sum_{j\in N} \lambda_j^g\, x_j \ : \ \sum_{j\in N} w_j\, x_j \le c \right\}. \tag{11}$$

The problem above is the analogue of problem (2). With the definitions of Z^* as in (10) and $\Pi^g(\lambda)$ as in (11), Lemma 1 continues to hold and we have $\sum_{g\in G} \alpha^g\, \Pi^g(\lambda) \ge Z^*$ for any $\lambda \in \Lambda$, where Λ is as defined in Section 2. As mentioned in Section 2, computing $\Pi^g(\lambda)$ at any λ requires solving an NP-complete problem. To approximate $\Pi^g(\lambda)$, we define $\Pi^g(\lambda, p_L, p_U)$ as

$$\Pi^g(\lambda, p_L, p_U) = \max_{x\in[0,1]^{|N|}} \left\{ \sum_{j\in N} (p_U\, r_j\, v_j^g - \lambda_j^g)\, x_j \ : \ \sum_{j\in N} v_j^g\, x_j \le \frac{1}{p_L} - 1,\ \sum_{j\in N} w_j\, x_j \le c \right\}, \tag{12}$$

which is the analogue of problem (4) under a space constraint. The problem above is a continuous knapsack problem with two dimensions and Sinha and Zoltners (1979) discuss efficient solution approaches for such knapsack problems. Letting $p_{\min} = \min_{g\in G}\{1/(1+\sum_{j\in N} v_j^g)\}$ and using $\{p^k : k = 1,\ldots,K+1\}$ to denote a set of grid points that satisfy $p_{\min} = p^1 \le p^2 \le \ldots \le p^K \le p^{K+1} = 1$, Proposition 2 continues to hold with the definitions of $\Pi^g(\lambda)$ as in (11) and $\Pi^g(\lambda, p_L, p_U)$ as in (12). In this case, we can solve problem (5) to obtain the tightest possible upper bound on the optimal expected revenue. Using the same approach in Section 4, we can show that the objective function of problem (5) is convex in λ with the definition of $\Pi^g(\lambda, p_L, p_U)$ as in (12) and we can efficiently obtain subgradients of the objective function of this problem, which indicates that problem (5) continues to be tractable under a space constraint. Therefore, the approach that we propose to compute upper bounds on the optimal expected revenue remains applicable when we have a constraint that limits the total space consumption of the offered products. The main difference is that we need to work with the continuous knapsack problem with two dimensions given in (12), instead of working with the continuous knapsack problem given in (4).

7.2 Mixture of Nested Logit Models

Under the nested logit model, the products are grouped into nests so that the products in the same nest are closer substitutes of each other when compared with the products in different nests. Given the nest structure, the choice process of a customer under the nested logit model proceeds in two stages. First, the customer either chooses one of the nests or decides to leave without a purchase. Second, if the customer chooses a nest, then the customer purchases one of the products offered in this nest. In this section, we extend our approach to the case where customers choose according to a mixture of nested logit models, as long as the number of products in each nest is reasonably small, but the number of nests can be large. To formulate the nested logit model, we use M to denote the set of nests and N to denote the set of products in each nest. Therefore, the total number of products is $|M| \times |N|$. The set of customer types is G and a customer of type g arrives with probability α^g. We use $S_i \subset N$ to denote the set of products that we offer in nest i. Therefore, the set of products offered in all nests are given by $\{S_i : i \in M\}$. A customer of type g associates the preference weight v_{ij}^g with product j in nest i. As a function of S_i, we use $V_i^g(S_i) = \sum_{j \in S_i} v_{ij}^g$ to denote the total preference weight of the products offered in nest i for a customer of type g. Under the nested logit model, if a customer of type g has already decided to make a purchase in nest i and the set S_i of products are offered in this nest, then the customer purchases product $j \in S_i$ with probability $v_{ij}^g / V_i^g(S_i)$. Thus, using r_{ij} to denote the revenue associated with product j in nest i, if a customer of type g has already decided to make a purchase in nest i and the set S_i of products are offered in this nest, then we obtain an expected revenue of

$$R_i^g(S_i) = \sum_{j \in S_i} r_{ij} \, \frac{v_{ij}^g}{V_i^g(S_i)}$$

from this customer. A customer of type g associates the dissimilarity parameter γ_i^g with nest i. In particular, the parameter γ_i^g measures how well the products in nest i substitute for each other for a customer of type g; see McFadden (1974) and Train (2003). Under the nested logit model, if the set of products offered in all nests are given by $\{S_i : i \in M\}$, then a customer of type g decides to make a purchase in nest i with probability

$$\frac{(V_i^g(S_i))^{\gamma_i^g}}{1 + \sum_{l \in M} (V_l^g(S_l))^{\gamma_l^g}},$$

where we normalize the preference weight of the no purchase option to one. Thus, since $R_i^g(S_i)$ is the expected revenue from a customer of type g that has already decided to make a purchase in nest i, if the set of products offered over all nests are given by $\{S_i : i \in M\}$, then the expected revenue obtained from a customer of type g is $\sum_{i \in M} R_i^g(S_i)\,(V_i^g(S_i))^{\gamma_i^g} / (1 + \sum_{i \in M} (V_i^g(S_i))^{\gamma_i^g})$. In this case, we can solve the problem

$$Z^* \quad = \max_{\substack{\{S_i \,:\, i \in M\} \,: \\ S_i \subset N \;\forall\, i \in M}} \left\{ \sum_{g \in G} \alpha^g \, \frac{\sum_{i \in M} R_i^g(S_i)\,(V_i^g(S_i))^{\gamma_i^g}}{1 + \sum_{i \in M} (V_i^g(S_i))^{\gamma_i^g}} \right\} \tag{13}$$

to find the set of products to offer over all nests so as to maximize the expected revenue obtained from a customer. In the problem above, the fraction computes the expected revenue from a customer of type g, whereas the outer sum computes the expected revenue over all customer types. In our formulation, we assume that there are $|N|$ products in each nest, but it is straightforward to extend our formulation to the case where different nests have different numbers of products.

If we have $\gamma_i^g = 1$ for all $i \in M$, $g \in G$, then the nested logit model becomes equivalent to the multinomial logit model; see Train (2003). Thus, solving problem (13) is at least as difficult as finding a set of products to offer that maximizes the expected revenue under a mixture of multinomial logit models. We focus on obtaining an upper bound on the optimal expected revenue Z^* in problem (13). Our approach for obtaining such an upper bound exploits the assumption that the number of products in each nest is reasonably small. The starting point for our approach is an appropriate reformulation of problem (13). To give this reformulation, we define the decision variable $x_i(S_i)$ such that $x_i(S_i) = 1$ is we offer the set S_i of products in nest i, otherwise we have $x_i(S_i) = 0$. In this case, using $x = \{x_i(S_i) : i \in M,\ S_i \subset N\} \in \{0,1\}^{|M| \times 2^{|N|}}$ to capture the sets of products offered in different nests and letting $\nu_i^g(S_i) = (V_i(S_i))^{\gamma_i^g}$ for notational brevity, we observe that problem (13) is equivalent to the problem

$$Z^* = \max_{x \in \{0,1\}^{|M| \times 2^{|N|}}} \left\{ \sum_{g \in G} \alpha^g \frac{\sum_{i \in M} \sum_{S_i \subset N} R_i^g(S_i)\, \nu_i^g(S_i)\, x_i(S_i)}{1 + \sum_{i \in M} \sum_{S_i \subset N} \nu_i^g(S_i)\, x_i(S_i)} \ : \sum_{S_i \subset N} x_i(S_i) = 1 \ \forall\, i \in M \right\}, \quad (14)$$

where the decision variables $\{x_i(S_i) : S_i \subset N\}$ describe which set of products we offer in nest i and the constraints ensure that we offer exactly one set of products in each nest, but this set can be the empty set. In the problem above, we have one decision variable $x_i(S_i)$ for each nest i and for each set $S_i \subset N$. Thus, the number of decision variables is manageable when the number of products in each nest $|N|$ is reasonably small. The number of decision variables is manageable even when the number of nests $|M|$ is large. Redefining the set of products appropriately, it is possible to see that the objective function of problem (14) is similar to the expected revenue function when customers choose according to a mixture of multinomial logit models. In particular, we index the products by $\{(i, S_i) : i \in M,\ S_i \subset N\}$. If a customer of type g purchases product (i, S_i), then we generate a revenue of $R_i^g(S_i)$. Furthermore, a customer of type g associates a preference weight of $\nu_i^g(S_i)$ with product (i, S_i). For all $i \in M$, $S_i \subset N$ and $g \in G$, we can compute and store $R_i^g(S_i)$ and $\nu_i^g(S_i)$ so that $\{R_i^g(S_i) : i \in M,\ S_i \subset N,\ g \in G\}$ and $\{\nu_i^g(S_i) : i \in M,\ S_i \subset N,\ g \in G\}$ become constant parameters in problem (14). Thus, comparing the objective function of problem (14) with that of problem (1), the objective function of problem (14) is similar to the expected revenue function when customers choose according to a mixture of multinomial logit models.

Building on this similarity, we can use the approach in Section 2 to obtain an upper bound on the optimal expected revenue Z^* in problem (14). In particular, we use the penalty parameters $\lambda = \{\lambda_i^g(S_i) : i \in M,\ S_i \subset N,\ g \in G\} \in \Re^{|M| \times 2^{|N|} \times |G|}$, where $\lambda_i^g(S_i)$ penalizes offering or not offering product (i, S_i) to a customer of type g. As a function of the penalty parameters, we define $\Pi^g(\lambda)$ as the optimal objective value of the problem

$$\begin{aligned} \Pi^g(\lambda) = \max_{x \in \{0,1\}^{|M| \times 2^{|N|}}} \Bigg\{ & \frac{\sum_{i \in M} \sum_{S_i \subset N} R_i^g(S_i)\, \nu_i^g(S_i)\, x_i(S_i)}{1 + \sum_{i \in M} \sum_{S_i \subset N} \nu_i^g(S_i)\, x_i(S_i)} - \sum_{i \in M} \sum_{S_i \subset N} \lambda_i^g(S_i)\, x_i(S_i) \\ & : \sum_{S_i \subset N} x_i(S_i) = 1 \ \forall\, i \in M \Bigg\}, \end{aligned} \quad (15)$$

which is the analogue of problem (2) under a mixture of nested logit models. We define the set of penalty parameters $\Lambda = \{\lambda \in \Re^{|M| \times 2^{|N|} \times |G|} : \sum_{g \in G} \alpha^g\, \lambda_i^g(S_i) = 0 \ \forall\, i \in M,\ S_i \subset N\}$. With this definition of Λ and the definitions of Z^* as in (14) and $\Pi^g(\lambda)$ as in (15), it is possible to check that Lemma 1 continues to hold and we have $\sum_{g \in G} \alpha^g\, \Pi^g(\lambda) \geq Z^*$ for any $\lambda \in \Lambda$. Due to the binary decision variables and the nonlinear objective function in problem (15), computing $\Pi^g(\lambda)$

for a particular value of λ can be difficult. We get around this difficulty by using an approximation to $\Pi^g(\lambda)$. For our approximation of $\Pi^g(\lambda)$, we define $\Pi^g(\lambda, p_L, p_U)$ as

$$\Pi^g(\lambda, p_L, p_U) \;=\; \max_{x\in[0,1]^{|M|\times 2^{|N|}}} \left\{ \sum_{i\in M} \sum_{S_i\subset N} (p_U R_i^g(S_i)\, \nu_i^g(S_i) - \lambda_i^g(S_i))\, x_i(S_i) \right.$$
$$\left. : \sum_{i\in M} \sum_{S_i\subset N} \nu_i^g(S_i)\, x_i(S_i) \le \frac{1}{p_L} - 1, \;\; \sum_{S_i\subset N} x_i(S_i) = 1 \;\forall\, i\in M \right\}, \quad (16)$$

which is the analogue of problem (4) under a mixture of nested logit models. The problem above is a continuous multiple choice knapsack problem; see Sinha and Zoltners (1979). In the objective function of problem (15), the smallest value of $1/(1+\sum_{i\in M}\sum_{S_i\subset N}\nu_i^g(S_i)\,x_i(S_i))$ in any feasible solution to this problem is $1/(1+\sum_{i\in M}\nu_i^g(N))$. Thus, letting $p_{\min} = \min_{g\in G}\{1/(1+\sum_{i\in M}\nu_i^g(N))\}$ and using $\{p^k : k = 1,\ldots,K+1\}$ to denote a set of grid points that satisfy $p_{\min} = p^1 \le p^2 \le \ldots \le p^K \le p^{K+1} = 1$, Proposition 2 continues to hold with the definitions of $\Pi^g(\lambda)$ as in (15) and $\Pi^g(\lambda, p_L, p_U)$ as in (16). In this case, we can solve problem (5) to obtain the tightest possible upper bound on the optimal expected revenue. We can use the same approach in Section 4 to show that the objective function of problem (5) is convex in λ with the definition of $\Pi^g(\lambda, p_L, p_u)$ as in (16). Also, we can use the same approach in Section 4 to obtain subgradients of the objective function of problem (5), which implies that problem (5) continues to be tractable under a mixture of nested logit models. These observations indicate that the approach that we propose to obtain upper bounds on the optimal expected revenue remains applicable under a mixture of nested logit models. The main difference is that we need to work with problem (16), instead of problem (4).

8 Computational Experiments

In this section, we provide computational experiments that test the quality of the upper bounds on the optimal expected revenue that we obtain by solving problem (5).

8.1 Benchmark Strategies

We compare the upper bounds provided by the following three benchmark strategies.

Penalty Multipliers (PM). This benchmark strategy corresponds to the upper bound provided by the optimal objective value of problem (5). The set of grid points that we use is of the form $\{(1+\rho)^{-k+1} : k = 1,\ldots,K+1\}$ with $K = O(\log(p_{\min})/\log(1+\rho))$. We use $\rho = 0.001$. To ensure that $\lambda \in \Lambda$, we choose an arbitrary customer type ϕ and assume that only the penalty multipliers $\{\lambda_j^g : j \in N,\; g \in G \setminus \{\phi\}\}$ are decision variables in problem (5). We solve the penalty parameters corresponding to customer type ϕ in terms of the other penalty parameters to obtain $\lambda_j^\phi = -\sum_{g\in G\setminus\{\phi\}} \alpha^g \lambda_j^g/\alpha^\phi$ for all $j \in N$. In this way, we ensure that $\lambda \in \Lambda$ without explicitly imposing this constraint. When implementing PM, we solve problem (5) by using subgradient search with the initial solution $\lambda = \bar{0}$. We use subgradient search for 100 iterations, where the step size at iteration t is of the form $1/t$. Our hope is that these 100 iterations get us into the vicinity of a reasonable solution. After these 100 iterations, we switch to another form for the step size, where we increase the step size by a factor of two after each iteration that yields an improvement in the objective value of problem (5), whereas we decrease the step size by a factor of two after each iteration that does not yield an improvement. This form for the step size may not ensure convergence to an optimal solution to problem (5), but it provides consistently good performance in our experience. Since the optimal objective value of problem (5) provides an upper bound on the optimal expected revenue Z^* and this problem is a minimization problem, any feasible solution to problem (5) also provides an upper bound on the optimal expected revenue.

Customer Type Decomposition (CD). This benchmark strategy corresponds to the upper bound obtained under the assumption that we know the type of an arriving customer so that we can offer different sets of products to customers of different types. In particular, we can solve the problem $\max_{x \in \{0,1\}^{|N|}} \sum_{j \in N} P_j^g(x)\, r_j = \max_{x \in \{0,1\}^{|N|}} (\sum_{j \in N} r_j\, v_j^g\, x_j)/(1 + \sum_{j \in N} v_j^g\, x_j)$ to find a set of products that maximizes the expected revenue obtained from a customer of type g. Talluri and van Ryzin (2004) show that this problem, which involves a single customer type, can be solved efficiently. Thus, letting $\hat{Z}^g$ be the optimal objective value of this problem, the largest expected revenue that we can obtain from a customer of type g is given by $\hat{Z}^g$. The upper bound provided by CD is $\sum_{g \in G} \alpha^g\, \hat{Z}^g$, which corresponds to the optimal expected revenue that can be obtained under the assumption that we can offer different sets of products to customers of different types. Since problem (1) requires that we offer a single set of products to customers of all types, $\sum_{g \in G} \alpha^g\, \hat{Z}^g$ is an upper bound on the optimal objective value of problem (1). CD builds on Talluri (2011), where the author shows that allowing a retailer to offer different sets to different customer types can provide good approximations in network revenue management problems.

Branch and Bound (BB). Bront et al. (2009) give a mixed integer programming formulation for problem (1), but solving this mixed integer programming formulation to optimality can be time consuming for large problem instances. We apply branch and bound on the mixed integer programming formulation for a fixed amount of run time and check the best upper bound that branch and bound achieves on the optimal objective value of the mixed integer program. Therefore, the upper bound provided by BB corresponds to the best upper bound that we obtain by using branch and bound for a fixed amount of run time. We choose the run time for branch and bound as twice the run time for PM, so that we can compare the upper bounds obtained by PM and BB within comparable amounts of run time.

8.2 Experimental Setup

In our computational experiments, we generate a large number of problem instances and compare the upper bounds provided by PM, CD and BB for each problem instance. We use the following approach for generating our problem instances. Throughout our computational experiments, the number of products $|N|$ is 100. We vary the number of customer types $|G|$. To come up with the revenues, we simply sample r_j from the uniform distribution over $[0, 2000]$ for all $j \in N$. To come up with the probabilities $\{\alpha^g : g \in G\}$ of observing customers of different types, we sample β^g from the uniform distribution over $[0, 1]$ for all $g \in G$ and set $\alpha^g = \beta^g / \sum_{c \in G} \beta^c$.

To come up with the preference weights, we choose a set $S \subset N$ of products and designate them as specialty products. We refer to the remaining set of products as staple products. Throughout our computational experiments, the number of staple products $|S|$ is 40. Customers of different types can associate significantly different preference weights with a specialty product, indicating that the evaluations of a specialty product by customers of different types can be quite different. Customers of different types evaluate a staple product more or less in the same fashion. To generate preference weights with these characteristics, for all $j \in N$, $g \in G$, we sample X_j^g as follows. If product j is a specialty product, then we sample X_j^g from the uniform distribution over $[0.1, 0.3] \cup [0.7, 0.9]$, whereas if product j is a staple product, then we sample X_j^g from the uniform distribution over $[0.3, 0.7]$. Thus, the variance of X_j^g is larger when product j is a specialty product. For all $j \in N$, we also sample κ_j from the uniform distribution over $[1, \bar{K}]$, where $\bar{K}$ is a parameter that we vary in our computational experiments. In this case, we set the preference weight v_j^g that a customer of type g associates with product j as a quantity that is proportional to $\kappa_j X_j^g$. In this setup, the value of κ_j determines an overall magnitude for the preference weights $\{v_j^g : g \in G\}$ associated with product j. Furthermore, if product j is a specialty product, then the variance of X_j^g is relatively large, in which case, the variance of $\kappa_j X_j^g$ is relatively large as well. So, if product j is a specialty product, then the preference weights $\{v_j^g : g \in G\}$ that customers of different types associate with product j display relatively large variability among themselves, which agrees with our expectation from a specialty product. Similarly, if product j is a staple product, then the variance of X_j^g is relatively small so that the preference weights $\{v_j^g : g \in G\}$ that customers of different types associate with product j display relatively small variability among themselves.

As mentioned above, we set the preference weight v_j^g that a customer of type g associates with product j as a quantity that is proportional to $\kappa_j X_j^g$. To come up with the values of the preference weights, we sample P_0^g from the uniform distribution over $[0, \bar{P}_0]$ for all $g \in G$, where $\bar{P}_0$ is a parameter that we vary in our computational experiments. In this case, we set the value of the preference weight v_j^g as $v_j^g = \kappa_j X_j^g (1 - P_0^g) / (P_0^g \sum_{i \in N} \kappa_i X_i^g)$. Noting that $\sum_{j \in N} v_j^g = \sum_{j \in N} \kappa_j X_j^g (1 - P_0^g) / (P_0^g \sum_{i \in N} \kappa_i X_i^g) = (1 - P_0^g) / P_0^g$ in this setup, even if we offer all of the products to the customers, a customer of type g leaves without making a purchase with probability $1/(1 + \sum_{j \in N} v_j^g) = 1/(1 + (1 - P_0^g)/P_0^g) = P_0^g$. Therefore, if we use a larger value for $\bar{P}_0$, then customers are more likely to leave without making a purchase. Also, if we use a larger value for $\bar{P}_0$, then the variance of P_0^g gets larger and customers of different types tend to become more heterogeneous in terms of their tendency to leave without making a purchase.

In our computational experiments, we vary $|G|$, $\bar{K}$ and $\bar{P}_0$ over $|G| \in \{25, 50, 75\}$, $\bar{K} \in \{5, 10, 20\}$ and $\bar{P}_0 \in \{0.6, 0.8, 1.0\}$. This setup provides 27 parameter combinations. In each parameter combination, we generate 1000 individual problem instances by using the approach described above. For each problem instance, we compute the upper bounds on the optimal expected revenue provided by PM, CD and BB. To put these upper bounds into perspective, we also use a greedy heuristic to find a solution to problem (1). In the greedy heuristic, we start with a solution to problem (1) that does not include any products. Given the current solution, we try adding or removing each one of the products into or from this current solution. Among all of these options, we update the current solution by using the option that provides the largest improvement in the expected revenue from the current solution. If none of the options provides an improvement, then we stop. The expected revenue from the solution obtained by the greedy heuristic provides a lower bound on the optimal expected revenue. By checking the gap between the upper bound on the optimal expected revenue provided by PM, CD or BB and the expected revenue from the greedy heuristic, we can assess how PM, CD and BB compare with each other in terms of the tightness of their upper bounds and we can get a conservative estimate of how much the upper bounds provided by PM, CD and BB deviate from the optimal expected revenue.

8.3 Computational Results

We give our main computational results in Table 1. The first column in this table shows the parameter combinations for our test problems by using $(|G|, \bar{K}, \bar{P}_0)$. We recall that we generate 1000 problem instances in each parameter combination. For each problem instance k, we compute the expected revenue from the solution obtained by the greedy heuristic. We let GRR^k be this expected revenue. We use PM, CD and BB to compute upper bounds on the optimal expected revenue. We let PMU^k, CDU^k and BBU^k respectively be the upper bounds provided by PM, CD and BB for problem instance k. The second column in Table 1 shows the percent gap between the upper bounds from PM and the expected revenues from the greedy heuristic, averaged over all problem instances in a parameter combination. In other words, this column shows the average of the data points $\{100 \times (\mathsf{PMU}^k - \mathsf{GRR}^k)/\mathsf{PMU}^k : k = 1, \ldots, 1000\}$, which can be used to assess the average optimality gap of the greedy heuristic when we use the upper bounds from PM to check the quality of a solution. The third and the fourth columns respectively show the 95th percentile and maximum of the same data points $\{100 \times (\mathsf{PMU}^k - \mathsf{GRR}^k)/\mathsf{PMU}^k : k = 1, \ldots, 1000\}$. The interpretations of the fifth, sixth and seventh columns are similar to those of the previous three columns, but the fifth, sixth and seventh columns respectively show the average, 95th percentile and maximum of the percent gaps between $\{\mathsf{CDU}^k : k = 1, \ldots, 1000\}$ and $\{\mathsf{GRR}^k : k = 1, \ldots, 1000\}$, giving a feel for the optimality gaps of the greedy heuristic when we only use the upper bounds provided by CD to check the quality of a solution. Finally, the eighth, ninth and tenth columns respectively show the average, 95th percentile and maximum of the percent gaps between $\{\mathsf{BBU}^k : k = 1, \ldots, 1000\}$ and $\{\mathsf{GRR}^k : k = 1, \ldots, 1000\}$, which indicate the optimality gaps of the greedy heuristic when we only use BB to obtain upper bounds on the optimal expected revenues.

The results in Table 1 indicate that the upper bounds provided by PM for our problem instances are quite tight. Over all of our problem instances, the average gap between the upper bounds

Param. Comb. $(\lvert G\rvert,\ \bar{K},\ \bar{P}_0)$	% Gap. of PMU^k with GRR^k			% Gap. of CDU^k with GRR^k			% Gap. of BBU^k with GRR^k		
	Avg.	95th	Max.	Avg.	95th	Max.	Avg.	95th	Max.
(25, 5, 0.6)	0.14	0.29	0.96	5.13	8.38	11.20	4.66	7.79	10.22
(25, 5, 0.8)	0.15	0.33	0.93	5.87	9.74	13.09	4.63	8.10	12.27
(25, 5, 1.0)	0.16	0.35	0.93	6.12	10.35	15.37	3.87	7.35	12.91
(25, 10, 0.6)	0.14	0.29	0.90	5.42	9.13	12.89	5.09	8.58	12.75
(25, 10, 0.8)	0.15	0.42	0.99	5.98	10.48	15.57	5.30	9.59	14.05
(25, 10, 1.0)	0.16	0.38	0.96	6.37	11.19	15.37	5.06	9.30	13.77
(25, 20, 0.6)	0.15	0.35	0.85	5.56	9.26	13.14	5.19	8.87	12.91
(25, 20, 0.8)	0.15	0.34	0.97	6.14	10.52	16.96	5.58	9.89	16.00
(25, 20, 1.0)	0.16	0.39	0.99	6.36	11.23	17.52	5.31	9.64	14.70
(50, 5, 0.6)	0.15	0.32	0.67	5.46	8.10	10.75	5.03	7.64	10.44
(50, 5, 0.8)	0.16	0.34	0.68	6.14	9.28	12.95	5.08	7.87	11.35
(50, 5, 1.0)	0.16	0.32	0.95	6.38	9.80	12.17	4.91	7.84	9.79
(50, 10, 0.6)	0.15	0.29	0.80	5.69	8.62	12.20	5.51	8.35	12.05
(50, 10, 0.8)	0.15	0.32	0.86	6.34	9.60	12.72	5.73	8.82	12.02
(50, 10, 1.0)	0.16	0.37	0.96	6.70	10.14	14.76	5.65	8.62	12.63
(50, 20, 0.6)	0.15	0.29	0.82	5.81	8.68	12.47	5.63	8.49	12.34
(50, 20, 0.8)	0.16	0.33	0.97	6.57	9.83	14.26	6.12	9.27	13.97
(50, 20, 1.0)	0.16	0.36	0.87	6.85	10.53	14.73	5.87	9.41	13.32
(75, 5, 0.6)	0.15	0.27	0.84	5.52	7.91	9.98	5.43	7.76	9.78
(75, 5, 0.8)	0.15	0.28	0.85	6.28	8.89	10.64	5.85	8.43	10.08
(75, 5, 1.0)	0.16	0.32	0.75	6.57	9.70	11.36	5.63	8.57	10.14
(75, 10, 0.6)	0.15	0.26	0.81	5.77	8.39	11.01	5.70	8.29	10.65
(75, 10, 0.8)	0.15	0.29	0.93	6.47	9.60	13.77	6.05	9.19	13.27
(75, 10, 1.0)	0.15	0.30	0.77	6.81	9.88	14.08	6.00	9.02	13.12
(75, 20, 0.6)	0.14	0.27	0.97	5.93	8.56	12.69	5.86	8.46	12.64
(75, 20, 0.8)	0.15	0.29	0.83	6.63	9.61	13.50	6.24	9.18	13.15
(75, 20, 1.0)	0.16	0.31	0.98	7.04	10.11	13.21	6.25	9.26	12.30
Average	0.15			6.15			5.45		

Table 1: Comparison of the upper bounds provided by PM, CD and BB.

from PM and the expected revenues from the greedy heuristic is 0.15%, whereas the maximum gap between the upper bounds from PM and the expected revenues from the greedy heuristic is 0.99%. The small gaps between the upper bounds from PM and the expected revenues from the greedy heuristic demonstrate that the upper bounds provided by PM are within a fraction of a percent of the optimal expected revenues. Furthermore, if we use PM to obtain upper bounds on the optimal expected revenues, then we can establish that the greedy heuristic provides optimality gaps no larger than 0.99% for our problem instances. In contrast, the upper bounds provided by CD or BB can be substantially looser. The gap between the upper bound from CD and the expected revenue from the greedy heuristic can be as large as 17.52%. In other words, if we use the upper bounds from CD to evaluate the quality of a solution, then there are problem instances where we are left with the impression that the greedy heuristic may have optimality gaps as large as 17.52%, although we can use the upper bounds from PM to establish that the optimality gaps of the greedy heuristic are actually no larger than 0.99%. The upper bounds provided by BB improve those provided by CD slightly. The average and maximum gaps between the upper bounds from BB and the expected revenues from the greedy heuristic are respectively 5.45% and 16%. The same gaps are respectively 6.15% and 17.52% when we consider CD. Overall, our computational results for PM demonstrate that the upper bounds from PM are within 1% of the optimal expected revenues and the optimality gaps of the greedy heuristic are no larger than 1%. The last observation, together with the fact that the gap between the upper bound provided by BB and the expected revenue from the greedy heuristic can exceed 15%, indicates that the upper bound provided by BB can deviate from the optimal expected revenue by almost 14%. Naturally, the mixed integer programming formulation used by BB would eventually obtain the optimal expected revenue, but it turns out that this formulation is not effective when we want to obtain good upper bounds on the optimal expected revenues within a limited amount of run time. We shortly investigate the reasons why BB does not yield tight upper bounds.

It is useful to point out an interesting trend in Table 1. As $\bar{P}_0$ increases, there are larger gaps between the upper bound provided by CD and the expected revenue from the greedy heuristic. As mentioned when describing our experimental setup in Section 8.2, as $\bar{P}_0$ increases, customers of different types tend to become more heterogeneous in terms of their tendency to leave without making a purchase. As customers of different types become more heterogeneous, CD, which is based on the assumption that we can offer different sets of products to different customer types, ends up offering significantly different sets to different customer types. In this case, the upper bound from CD can deviate significantly from the optimal objective value of problem (1), which does not allow offering different sets of products to different customer types. In contrast, the gaps between the upper bound provided by PM and the expected revenue from the greedy heuristic remain quite stable as $\bar{P}_0$ increases.

The results in Table 1 show that the upper bounds provided by BB are not as tight, indicating that the mixed integer program used by BB is ineffective in obtaining good upper bounds within a limited amount of run time. One reason that BB is not able to obtain good upper bounds is that the linear programming relaxation of the mixed integer program used by BB turns out to be loose. In all of our test problems, the linear programming relaxation of the mixed integer program only slightly improves the upper bound from CD. To shed more light into this observation, Proposition 4 in Online Appendix A shows that when we focus on each customer type individually, if customers of each type make a purchase with a probability that exceeds 1/2, then the optimal objective value of the linear programming relaxation of the mixed integer program used by BB precisely corresponds to the upper bound provided by CD. Thus, although it is tempting to try to obtain upper bounds on the optimal expected revenue by solving the linear programming relaxation of the mixed integer program used by BB, this upper bound does not improve the one provided by CD when customers make a purchase with a reasonably large probability.

In Table 2, we give the details on the gaps between the upper bounds obtained by our benchmark strategies and the expected revenues from the greedy heuristic. The first column in this table shows the parameter combinations for our test problems. The second column shows the number of problem instances where the gap between the upper bound obtained by PM and the expected revenue from the greedy heuristic is less than 0.125%. The interpretations of the third, fourth, fifth, sixth and seventh columns are similar to that of the second column, but these columns show the numbers of problem instances where the gap between the upper bound obtained by PM and the expected revenue from the greedy heuristic is respectively less than 0.25%, 0.5%, 2.5%, 5% and 7.5%. The eighth to thirteenth columns have the same interpretations as the second to seventh columns, but they focus on the gap between the upper bound obtained by BB and the expected revenue from the greedy heuristic. The upper bounds provided by BB are slightly better than those from CD. For economy of space, we do not provide the details on CD. The results in Table 2 indicate that in more than 24000 out of 27000 problem instances, we can use the upper bounds from PM to conclude that the optimality gap of the greedy heuristic is smaller than 0.25%, which also implies that the upper bounds provided by PM for these problem instances deviate from the optimal expected revenues by at most 0.25%. In contrast, the upper bounds provided by BB deviate from the expected revenues from the greedy heuristic by less than 5% in only about 11500 out of 27000 problem instances. For PM, the gaps between the upper bounds and the expected revenues from the greedy heuristic are almost exclusively less than 0.5%, whereas the gaps between the upper bounds from BB and the expected revenues from the greedy heuristic almost never falls below 0.5%.

The run times for PM are quite reasonable. Over all of our problem instances, the average run time for PM is 3.95 seconds. Considering the problem instances with 25, 50 and 75 customer types separately, the average run time for PM is respectively 1.91, 3.94 and 6.01 seconds. Overall, our results indicate that PM can obtain quite tight upper bounds on the optimal expected revenues. The small gaps between the upper bounds from PM and the expected revenues from the greedy heuristic do not only demonstrate that the upper bounds provided by PM are close to the optimal expected revenues, but also point out that the greedy heuristic is effective in obtaining near optimal solutions. In this way, the upper bounds provided by PM can be used to check the quality of the solutions provided by not only the greedy heuristic, but also any other heuristic or approximation method that is used to obtain solutions to assortment problems, when customers choose according to a mixture of multinomial logit models.

8.4 Specially Structured Problem Instances

The computational results that we present so far indicate that the upper bounds provided by PM can be quite tight. In this section, we work with small and specially structured problem instances to demonstrate that it is possible to come up with problem instances where the upper bounds provided by PM are not quite as tight. In particular, we focus on a class of problem instances where the number of customer types $|G|$ is equal to the number of products $|N|$. For a scalar $\theta > 1$, Table 3 lists the parameters of a problem instance for the case with $|G| = |N| = 3$. For example, the revenues associated with the products in this problem instance are $(r_1, r_2, r_3) = (1, \theta, \theta^2)$. A customer of type two associates the preference weights $(v_1^2, v_2^2, v_3^2) = (\theta^6, \theta^4, 0)$ with the products. The probability of observing a customer of type two is $\alpha^2 = \theta/(1 + \theta + \theta^2)$. Following

Param. Comb. $(\lvert G\rvert,\ \bar{K},\ \bar{P}_0)$	Number of Problems with a Certain % Gap between PMU^k and GRR^k						Number of Problems with a Certain % Gap between BBU^k and GRR^k					
	0.125%	0.25%	0.5%	2.5%	5%	7.5%	0.125%	0.25%	0.5%	2.5%	5%	7.5%
(25, 5, 0.6)	622	932	990	1000	1000	1000	0	0	0	105	598	939
(25, 5, 0.8)	606	908	976	1000	1000	1000	0	0	0	159	592	908
(25, 5, 1.0)	577	891	973	1000	1000	1000	0	0	1	257	757	956
(25, 10, 0.6)	642	926	984	1000	1000	1000	0	0	0	78	524	876
(25, 10, 0.8)	672	903	965	1000	1000	1000	0	0	0	83	510	828
(25, 10, 1.0)	608	889	973	1000	1000	1000	0	0	0	110	536	856
(25, 20, 0.6)	673	919	979	1000	1000	1000	0	0	0	65	518	857
(25, 20, 0.8)	665	916	984	1000	1000	1000	0	0	0	65	437	816
(25, 20, 1.0)	652	896	973	1000	1000	1000	0	0	1	85	492	824
(50, 5, 0.6)	519	907	989	1000	1000	1000	0	0	0	15	529	937
(50, 5, 0.8)	508	899	986	1000	1000	1000	0	0	0	26	518	924
(50, 5, 1.0)	498	900	980	1000	1000	1000	0	0	0	46	559	938
(50, 10, 0.6)	571	926	993	1000	1000	1000	0	0	0	15	420	878
(50, 10, 0.8)	555	924	986	1000	1000	1000	0	0	0	13	373	837
(50, 10, 1.0)	534	900	977	1000	1000	1000	0	0	0	20	376	854
(50, 20, 0.6)	573	918	987	1000	1000	1000	0	0	0	9	367	873
(50, 20, 0.8)	548	897	984	1000	1000	1000	0	0	0	11	294	755
(50, 20, 1.0)	513	893	984	1000	1000	1000	0	0	0	15	354	820
(75, 5, 0.6)	498	942	993	1000	1000	1000	0	0	0	4	399	934
(75, 5, 0.8)	516	932	990	1000	1000	1000	0	0	0	2	313	862
(75, 5, 1.0)	446	905	986	1000	1000	1000	0	0	0	8	386	874
(75, 10, 0.6)	522	944	995	1000	1000	1000	0	0	0	0	350	874
(75, 10, 0.8)	515	927	984	1000	1000	1000	0	0	0	1	257	832
(75, 10, 1.0)	471	925	992	1000	1000	1000	0	0	0	7	302	824
(75, 20, 0.6)	561	941	996	1000	1000	1000	0	0	0	0	288	873
(75, 20, 0.8)	523	921	989	1000	1000	1000	0	0	0	3	240	784
(75, 20, 1.0)	471	913	988	1000	1000	1000	0	0	0	0	238	777
Total	15059	24694	26576	27000	27000	27000	0	0	2	1202	11527	23310

Table 2: Distribution of the upper bounds provided by PM and BB.

the pattern in Table 3, it is straightforward to generalize this problem instance to a larger value for $|G|$ and $|N|$. For example, for the case with $|G| = |N| = 5$, the revenues associated with the five products are $(r_1, r_2, r_3, r_4, r_5) = (1, \theta, \theta^2, \theta^3, \theta^4)$. A customer of type one associates the preference weights $(v_1^1, v_2^1, v_3^1, v_4^1, v_5^1) = (\theta^{10}, \theta^8, \theta^6, \theta^4, \theta^2)$ with the products, whereas a customer of type two associates the preference weights $(v_1^2, v_2^2, v_3^2, v_4^2, v_5^2) = (\theta^{10}, \theta^8, \theta^6, \theta^4, 0)$. The probability of observing a customer of type two would be $\alpha^2 = \theta/(1 + \theta + \theta^2 + \theta^3 + \theta^4)$.

The motivation behind the problem instance in Table 3 is that if the value of θ is large, then CD, which offers different sets of products to different customer types, provides an upper bound that deviates from the optimal expected revenue by a factor that is close to $|G| = |N|$. Thus, for this problem instance, the upper bound obtained under the assumption that we can offer different sets of products to different customer types can be quite poor. To see this result, we observe that if we offer the set $\{3\}$ of products to a customer of type one, then we obtain an expected revenue of $\theta^4/(1 + \theta^2)$ from this customer type. If we offer the set $\{2\}$ of products to a customer of type two, then we obtain an expected revenue of $\theta^5/(1 + \theta^4)$ from this customer type. Lastly, if we offer the

Revenues

Product		
1	2	3
1	θ	θ^2

Preference Weights

Cus. Typ.	Product		
	1	2	3
1	θ^6	θ^4	θ^2
2	θ^6	θ^4	0
3	θ^6	0	0

Arrival Probs.

Cus. Typ.		
1	2	3
$\dfrac{1}{1+\theta+\theta^2}$	$\dfrac{\theta}{1+\theta+\theta^2}$	$\dfrac{\theta^2}{1+\theta+\theta^2}$

Table 3: A specially structured problem instance with $|G| = |N| = 3$.

set $\{1\}$ of products to a customer of type three, then we obtain an expected revenue of $\theta^6/(1+\theta^6)$ from this customer type. Thus, noting the probability of observing each customer type in Table 3, the upper bound on the optimal expected revenue that we obtain by offering different sets of products to customers of different types is at least

$$\frac{1}{1+\theta+\theta^2}\left\{\frac{\theta^4}{1+\theta^2}+\theta\,\frac{\theta^5}{1+\theta^4}+\theta^2\,\frac{\theta^6}{1+\theta^6}\right\}. \tag{17}$$

In other words, using CDU to denote the upper bound obtained by CD, CDU is no smaller than the quantity in (17). On the other hand, considering the optimal expected revenue in problem (1) for this problem instance, we can check the expected revenue from each set of products, in which case, it is straightforward to establish that the optimal expected revenue Z^* in problem (1) for this problem instance is at most $(3+2\,\theta+\theta^2)/(1+\theta+\theta^2)$. We give the details of this computation in Online Appendix B. In this case, noting the expression in (17), it follows that

$$\frac{\mathsf{CDU}}{Z^*} \geq \frac{\dfrac{1}{1+\theta+\theta^2}\left\{\dfrac{\theta^4}{1+\theta^2}+\theta\,\dfrac{\theta^5}{1+\theta^4}+\theta^2\,\dfrac{\theta^6}{1+\theta^6}\right\}}{\dfrac{3+2\,\theta+\theta^2}{1+\theta+\theta^2}}.$$

As θ approaches to infinity, the expression on the right side above approaches to three. Thus, for large values of θ, the upper bound obtained by offering different sets of products to customers of different types exceeds the expected revenue by a factor that is close to three, indicating that the upper bounds provided by CD can be quite loose for this problem instance.

A natural question is how much we can improve the upper bound provided by CD for this class of problem instances through the use of penalty multipliers, which corresponds to the upper bound provided by PM. It is difficult to compute the upper bound provided by PM in closed form and we carry out numerical experiments. In Table 4, we give the optimal expected revenue Z^*, along with the upper bounds on the optimal expected revenue obtained by PM and CD for different problem instances parameterized by θ and $|G|$. All of the problem instances in Table 4 are generated by following the pattern in Table 3 with values of $\theta \in \{2,4,8\}$ and $|G| \in \{3,4,5\}$. The first column in this table shows the parameter combinations by using $(\theta, |G|)$. The second column shows the optimal expected revenue Z^*, corresponding to the optimal objective value of problem (1). Since the number of products is reasonably small, we compute the optimal expected revenue by checking

Param. Comb. $(\theta, \lvert G \rvert)$	Z^*	PMU	CDU	$\frac{\text{PMU}}{Z^*}$	$\frac{\text{CDU}}{Z^*}$
$(2,3)$	1.09	1.09	1.56	1.00	1.43
$(2,4)$	1.12	1.27	1.99	1.13	1.77
$(2,5)$	1.13	1.49	2.44	1.32	2.15
$(4,3)$	1.04	1.24	2.24	1.19	2.14
$(4,4)$	1.05	1.73	2.96	1.65	2.83
$(4,5)$	1.05	2.00	3.71	1.91	3.54
$(8,3)$	1.01	1.37	2.62	1.35	2.58
$(8,4)$	1.01	1.98	3.49	1.96	3.44
$(8,5)$	1.01	2.26	4.36	2.23	4.30

Table 4: Upper bounds provided by PM and CD for the specially structured problem instances.

the expected revenue provided by every possible assortment. The third column shows the upper bound obtained by PM, which corresponds to the approach that propose in this paper. The fourth column shows the upper bound obtained by CD, which corresponds to the upper bound obtained by offering different sets to different customer types. Letting PMU and CDU respectively be the upper bounds obtained by PM and CD, the fifth and sixth columns give the ratios PMU$/Z^*$ and CDU$/Z^*$, characterizing the tightness of the upper bounds provided by PM and CD.

The results in Table 4 indicate that the upper bounds provided by CD for the specially structured problem instances can be quite loose. For example, for the problem instance with $\theta = 8$ and $|G| = 5$, the upper bound provided by CD deviates from the optimal expected revenue by a factor of about 4.3. This observation is consistent with the earlier discussion that establishes that the upper bound provided by CD deviates from the optimal expected revenue by a factor that is close to $|G|$ when θ is large. The upper bounds provided by PM can significantly improve those provided by CD. For the problem instance with $\theta = 8$ and $|G| = 5$, the upper bound provided by PM deviates from the optimal expected revenue by a factor of about 2.23, while the upper bound provided by CD deviates by a factor of about 4.3. Nevertheless, the upper bounds provided by PM can still be quite loose. Furthermore, we observe that the upper bounds provided by both CD and PM tend to get looser as $|G|$ increases and there are more customer types. Thus, the results in Table 4 indicate that PM can significantly improve the upper bounds from CD even for these specially structured problem instances, but it is possible to construct problem instances where the upper bounds provided by PM can still be quite loose. It is also worthwhile to note that these problem instances, for which the upper bounds provided by PM can be quite loose, involve products whose revenues and preference weights differ by orders of magnitude. For example, even with the smallest value of two that we use for θ, if $|G| = 5$, then we have $v_1^1 = 1024$, but $v_5^1 = 4$.

There are three sources of error in the approach that PM uses to obtain upper bounds on the optimal expected revenue. As discussed in Section 6, PM is equivalent to using Lagrangian relaxation on an appropriate reformulation of problem (1). Since problem (1) does not have a concave objective function and it involves binary decision variables, we do not necessarily obtain the optimal objective value of this problem through Lagrangian relaxation. Thus, the first source of error is due to the fact that we use Lagrangian relaxation on a nonconvex optimization problem, potentially resulting in a duality gap. The other two sources of error, as discussed at the end of Section 3, is related to the approximation $\max_{k\in\{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$ that we use for $\Pi^g(\lambda)$. In particular, the second source of error is due to the fact that we use a finite number of grid points in $\{p^k : k = 1, \ldots, K+1\}$. Proposition 3 and the following discussion imply that if we use grid points of the form $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$, then this source of error is no more than a multiplicative factor of $1+\rho$. Thus, the second source of error can be alleviated by using a small value for ρ. The third source of error is due to the fact that we do not impose binary constraints on the decision variables in problem (4). The intuitive motivation is that the linear programming relaxations of knapsack problems can give good approximations to the version with binary constraints. For the large problem instances used in the previous section, none of the three sources of error appears to be problematic, as the results in Table 1 indicate that the upper bounds obtained by PM are quite tight. In the remainder of this section, we use the specially structured problem instances to investigate the three sources of error. For these problem instances, we compute the optimal expected revenue Z^* in problem (1). In addition to computing the optimal expected revenue, we use PM with $\rho \in \{0.5, 0.25, 0.1, 0.01, 0.001\}$, corresponding to five different numbers of grid points. Furthermore, we also use PM with $\rho = 0.001$, but impose binary constraints on the decision variables in problem (4).

Our results are summarized in Table 5. The first column in this table shows the parameter combinations by using $(\theta, |G|)$. The second column shows the optimal expected revenue Z^* in problem (1). The third to seventh columns show the upper bounds obtained by PM when we respectively use the values of 0.5, 0.25, 0.1, 0.01 and 0.001 for ρ. Finally, the eighth column shows the upper bound obtained by PM when we use the value of 0.001 for ρ and impose binary constraints on the decision variables in problem (4). As expected, when we use a smaller value for ρ and the set of grid points are denser, the upper bounds provided by PM become tighter. When we decrease ρ from 0.5 to 0.01, the upper bound quickly tightens, but decreasing ρ further from 0.01 to 0.001 yields a marginal improvement in the upper bound. These observations are consistent with the fact that error caused by a finite number of grid points is at most a multiplicative factor of $1+\rho$. If we impose binary constraints on the decision variables in problem (4), then the upper bound provided by PM only marginally improves. Over all of our problem instances, the improvement was no larger than 8×10^{-4}, corresponding to an improvement of about 0.07%. This improvement occurs for problem instance $(\theta, |G|) = (2, 3)$. Nevertheless, the upper bounds provided by PM can still be loose when compared with the optimal expected revenue. For these specially structured problem instances, our results indicate that by using a smaller value for ρ, we can quickly overcome the source of error due to the fact that we use a finite number of grid points. In addition, the error caused by the fact that we do not impose binary constraints in problem (4) does not appear to be problematic, as imposing binary constraints only marginally improves the upper bound. Thus, the most significant error appears to be due to the fact that we use Lagrangian relaxation on

Param. Comb. $(\theta, \|G\|)$	Z^*	PMU with a Certain Value for ρ 0.5	0.25	0.1	0.01	0.001	PMU Bin. Const.
$(2, 3)$	1.09	1.44	1.30	1.18	1.10	1.09	1.09
$(2, 4)$	1.12	1.79	1.47	1.34	1.28	1.27	1.27
$(2, 5)$	1.13	1.89	1.67	1.60	1.50	1.49	1.49
$(4, 3)$	1.04	1.68	1.43	1.31	1.25	1.24	1.24
$(4, 4)$	1.05	2.09	1.91	1.78	1.73	1.73	1.73
$(4, 5)$	1.05	2.47	2.24	2.07	2.01	2.00	2.00
$(8, 3)$	1.01	1.72	1.59	1.46	1.38	1.37	1.37
$(8, 4)$	1.01	2.22	2.23	2.10	2.01	1.98	1.98
$(8, 5)$	1.01	2.63	2.52	2.38	2.27	2.26	2.26

Table 5: Upper bounds provided by PM with different values for ρ and with binary constraints in problem (4).

a nonconvex optimization problem and we have a duality gap. Thus, by specially structuring pathological problem instances, it is possible to come up with cases where the duality gap is large, but the duality gaps do not appear to be a problem in any of the large problem instances that we work with. Also, these pathological instances are relatively unlikely to appear in practice, since as mentioned above, they involve products whose revenues and preference weights differ from each other by orders of magnitude.

8.5 Problem Instances with a Space Constraint

In Section 7.1, we describe how to obtain upper bounds on the optimal expected revenue when there is a space constraint on the set of offered products. In this section, we provide computational experiments under a space constraint. We generate our test problems by using the approach described in Section 8.2. The only difference is that we need to generate the space consumption of each product and the total amount of space available. To come up with the space consumption of product j, we simply set $w_j = 1$. This setup corresponds to the case where we limit the total number of products in the offered set. We also carried out computational experiments where we sample the space consumption of each product from the uniform distribution over $[0, 1]$ and the performance of the upper bounds obtained by PM qualitatively remained the same. For economy of space, we report the results for the case where $w_j = 1$ for all $j \in N$. One advantage of working with the case where $w_j = 1$ for all $j \in N$ is that if there is a single customer type and we have a limit on the number of products that can be offered, then Rusmevichientong et al. (2010) show that the optimal set of products to offer can efficiently be computed. Thus, it is tractable to compute the upper bound provided by CD when we have $w_j = 1$ for all $j \in N$, but this is not the case when different products have different space consumptions. To come up with the total amount of space available, after generating all of the other problem parameters as described in Section 8.2, we use the greedy heuristic at the end of Section 8.2 to compute a reasonably good solution without a space constraint. Using $\hat{x} = \{\hat{x}_j : j \in N\} \in \{0, 1\}^{|N|}$ to capture the set of products offered in this

Param. Comb. ($\|G\|$, $\bar{K}$, $\bar{P}_0$, γ)	% Gap. of PMU^k with GRR^k			% Gap. of CDU^k with GRR^k			% Gap. of BBU^k with GRR^k		
	Avg.	95th	Max.	Avg.	95th	Max.	Avg.	95th	Max.
(20, 5, 0.8, 0.6)	0.21	0.31	0.69	10.94	14.09	15.81	4.78	6.92	8.90
(20, 5, 0.8, 0.8)	0.20	0.27	0.39	7.15	11.35	12.28	4.11	7.61	10.53
(20, 5, 1.0, 0.6)	0.21	0.27	0.45	11.20	14.36	19.73	4.27	6.97	10.01
(20, 5, 1.0, 0.8)	0.21	0.33	0.61	6.76	11.49	13.30	3.33	6.59	9.32
(20, 10, 0.8, 0.6)	0.21	0.32	0.70	11.42	15.84	17.61	5.40	8.86	9.67
(20, 10, 0.8, 0.8)	0.21	0.31	0.40	7.00	11.85	14.72	4.75	8.83	11.55
(20, 10, 1.0, 0.6)	0.20	0.22	0.34	11.02	14.14	15.18	4.96	7.49	8.53
(20, 10, 1.0, 0.8)	0.20	0.29	0.34	7.74	12.49	16.49	4.61	8.26	10.97
(40, 5, 0.8, 0.6)	0.22	0.33	0.66	11.35	14.74	15.92	5.69	8.03	11.52
(40, 5, 0.8, 0.8)	0.21	0.34	0.38	7.56	11.28	12.34	5.03	7.92	9.70
(40, 5, 1.0, 0.6)	0.21	0.25	0.55	11.47	14.44	17.31	5.05	7.42	10.44
(40, 5, 1.0, 0.8)	0.21	0.27	0.87	7.32	10.26	11.76	4.36	6.59	8.30
(40, 10, 0.8, 0.6)	0.20	0.23	0.47	11.29	13.94	15.96	5.89	7.83	9.49
(40, 10, 0.8, 0.8)	0.21	0.34	0.77	7.75	11.48	12.71	5.27	8.28	10.00
(40, 10, 1.0, 0.6)	0.21	0.32	0.49	11.92	15.17	17.01	5.77	8.70	10.32
(40, 10, 1.0, 0.8)	0.21	0.29	0.70	8.16	11.49	12.05	5.02	7.85	9.13
Average	0.31			10.16			5.43		

Table 6: Comparison of the upper bounds provided by PM, CD and BB under a space constraint.

solution, we set the total amount of space available as $c = \gamma \sum_{j \in N} w_j \hat{x}_j$, where γ is a parameter that we vary. Thus, the total amount of space available is a γ fraction of the total amount of space consumed by an unconstrained reasonably good assortment. Bront et al. (2009) extend their mixed integer programming formulation to the case where there is a space constraint. Building on this formulation, we can continue using BB when there is a space constraint.

In our computational experiments, we vary $|G|$, $\bar{K}$, $\bar{P}_0$ and γ over $|G| \in \{20, 40\}$, $\bar{K} \in \{5, 10\}$, $\bar{P}_0 \in \{0.8, 1.0\}$ and $\gamma \in \{0.6, 0.8\}$, where $\bar{P}_0$ and $\bar{K}$ are as described in Section 8.2. This setup provides 16 parameter combinations. In each parameter combination, we generate 100 individual problem instances. Table 6 gives our computational results. The format of this table is similar to that of Table 1. The results in Table 6 indicate that PM continues to provide quite tight upper bounds on the optimal expected revenues when we have a space constraint and these upper bounds are significantly tighter than the ones from CD and BB. Over all of our problem instances, the average gap between the upper bounds obtained by PM and the expected revenues from the greedy heuristic is 0.31%. This observation implies that the average gap between the upper bounds obtained by PM and the optimal expected revenues is no larger than 0.31% as well. In the worst case, the gap between the upper bound obtained by PM and the optimal expected revenue is 0.87%. The upper bounds provided by CD and BB are significantly looser. The average gap between the upper bounds provided by CD and the expected revenues from the greedy heuristic is 10.16%. This gap can be as large as 19.73% in the worst case. On the other hand, the average gap between the upper bounds from BB and the expected revenues from the greedy heuristic is 5.43%. This gap can reach 11.52% in the worst case.

8.6 Problem Instances under a Mixture of Nested Logit Models

In Section 7.2, we describe how to obtain upper bounds on the optimal expected revenue when customers choose according to a mixture of nested logit models. In this section, we provide computational experiments under a mixture of nested logit models. Throughout our computational experiments, the number of nests $|M|$ is 10 and the number of products in each nest $|N|$ is 5, yielding a total of 50 products. The number of customer types $|G|$ is 50. To generate our test problems, for all $i \in M$ and $j \in N$, we sample the revenue r_{ij} associated with product j in nest i from the uniform distribution over $[200, 600]$. For all $g \in G$ and $i \in M$, we sample the dissimilarity parameter γ_i^g from the uniform distribution over $[0, 1]$. To come up with the preference weights, for all $g \in G$, $i \in M$ and $j \in N$, we sample η_{ij}^g from the uniform distribution over $[0, 200]$. For all $g \in G$, we also sample η_0^g from the uniform distribution over $[0, \bar{P}_0]$, where $\bar{P}_0$ is a parameter that we vary. In this case, we set the preference weight v_{ij}^g that a customer of type g associates with product j in nest i as $v_{ij}^g = \eta_{ij}^g / (\eta_0^g)^{1/\gamma_i^g}$. In this setup, if we offer all of the products in all of the nests, then a customer of type g leaves without making a purchase with probability $1/(1 + \sum_{i \in M} (V_i^g(N))^{\gamma_i^g}) = \eta_0^g / (\eta_0^g + \sum_{i \in M} (\sum_{j \in N} \eta_{ij})^{\gamma_i^g})$. So, customers of type g are more likely to leave without making a purchase when η_0^g is larger. Thus, as $\bar{P}_0$ gets larger, customers are more likely to leave without making a purchase. To come up with the customer arrival probabilities, we sample β^g from the uniform distribution over $[0, 1]$ for all $g \in G$ and set $\alpha^g = \beta^g / \sum_{c \in G} \beta^c$.

In our computational experiments, we vary $\bar{P}_0$ over $\bar{P}_0 \in \{25, 50, 100\}$. This setup provides three parameter combinations. In each parameter combination, we generate 100 individual problem instances. Table 7 gives our computational results. The first column in this table shows the parameter combination by using $\bar{P}_0$. The interpretations of the second, third and fourth columns in Table 7 are similar to those of the second, third and fourth columns in Table 1. These columns respectively show the average, 95th percentile and maximum of the percent gaps between the upper bound obtained by our approach and the expected revenue from the greedy heuristic, when we focus on 100 problem instances in a particular parameter combination. The interpretations of the fifth, sixth and seventh columns in Table 7 are similar to those of the second, third and fourth columns in Table 2. These columns show the number of problem instances for which the percent gap between the upper bound obtained by our approach and the expected revenue from the greedy heuristic are respectively less than 0.5%, 1% and 2%. Over all of our problem instances, the upper bounds obtained by our approach deviate from the expected revenues from the greedy heuristic by 0.58% on average, which implies that the average gap between our upper bounds and the optimal expected revenues is no larger than 0.58%. The maximum gaps in Table 7 are larger than those in Table 1. This difference is likely due to the fact that our extension to a mixture of nested logit models works with problem (16), which is an $|M|+1$ dimensional knapsack problem and the linear programming relaxations of such knapsack problems tend to be looser than those of one dimensional knapsack problems. Nevertheless, we observe that in more than 85% of our problem instances, the upper bounds obtained by our approach are within 1% of the optimal expected revenues.

Param. Comb. $\bar{P}_0$	% Gap of PMU^k with GRR^k Avg.	95th	Max	No. of Problems with a Certain % Gap between PMU^k and GRR^k 0.5%	1%	2%
25	0.48	1.47	3.24	72	90	98
50	0.64	2.20	3.46	63	83	91
100	0.60	1.90	2.67	60	84	96
Average	0.58					

Table 7: Upper bounds provided by our approach under a mixture of nested logit models.

9 Conclusions

We developed a method to obtain an upper bound on the optimal expected revenue in assortment problems under a mixture of multinomial logit models. Our approach focuses on each customer type one by one and finds a separate assortment that maximizes the expected revenue from each customer type, but we use penalty parameters to synchronize the assortments offered to different customer types. This strategy requires solving assortment problems with a single customer type but with a fixed cost for offering a product. We develop tractable approximations to such assortment problems by assuming that the probability of not making a purchase can take values over a prespecified grid. We show how to obtain a set of good penalty parameters and a good set of grid points. We extend our approach to the case where there is a constraint on the total space consumption of the offered products or where the customers choose according to a mixture of nested logit models. In our computational experiments, our upper bounds for randomly generated problem instances are quite tight. However, there are pathological problem instances, where the revenues and preference weights of the products differ from each other by orders of magnitude, for which our upper bounds are not as tight. Ultimately, our approach will hopefully increase the practical use of mixture of multinomial logit models. Although heuristics tend to provide good assortments, it is generally difficult to check the quality of the solutions obtained by heuristics and our upper bounds allow checking the quality of the solutions from any heuristic or approximation method.

There are several future research directions. Our approach for obtaining upper bounds has three sources of error. First, as mentioned in Section 6, our approach is based on using Lagrangian relaxation on a nonconvex optimization problem and there can be a duality gap. Second, we use a finite number of grid points. Third, we do not impose binary constraints in problem (4). In Section 5, we show that by using exponential grid points, we can bound the error due to the second source of error by a multiplicative factor of $1 + \rho$ for any $\rho > 0$. It is possible to show that the optimal objective value of a certain linear programming relaxation of a knapsack problem deviates from the optimal objective value of the binary version by at most a factor of two; see Williamson and Shmoys (2011). By using these two results, we can bound the error due the second and third sources of error by a multiplicative factor of $2\,(1 + \rho)$. However, it is difficult to bound the error due to the first source of error and it might be possible to find special cases where we can bound this error. Also, our extension to a mixture of nested logit models is under the assumption that the number of products in each nest is reasonably small, but the number of nests can be large. We can make extensions to the case where the number of nests is reasonably small but the number of products in each nest can be large. Briefly, the main idea for this extension is that if we have a reasonably small number of nests, then we can assume that the total preference weight of the products offered in a particular nest lies on a prespecified grid and we can carry out a search over an $|M|$ dimensional grid. Naturally, carrying out a search over an $|M|$ dimensional grid is tractable when the number of nests does not exceed three or four. A useful research direction is to make extensions to a mixture of nested logit models with a large number of nests and a large number of products in each nest. In addition to the nested logit model, it is useful to investigate upper bounds under more general choice models, for which it is difficult to compute the optimal assortment.

Acknowledgement. We thank the area editor, the senior editor and two anonymous referees whose comments improved the paper in many ways. This work was supported in part by National Science Foundation grant CMMI-0969113.

References

Blanchet, J., Gallego, G. and Goyal, V. (2013), A Markov chain approximation to choice modeling, Technical report, Columbia University, New York, NY.

Bront, J. J. M., Diaz, I. M. and Vulcano, G. (2009), 'A column generation algorithm for choice-based network revenue management', *Operations Research* **57**(3), 769–784.

Davis, J., Gallego, G. and Topaloglu, H. (2013), Assortment planning under the multinomial logit model with totally unimodular constraint structures, Technical report, Cornell University, School of Operations Research and Information Engineering. Available at `http://legacy.orie.cornell.edu/~huseyin/publications/publications.html`.

Davis, J., Gallego, G. and Topaloglu, H. (2014), 'Assortment optimization under variants of the nested logit model', *Operations Research* **62**(2), 250–273.

Desir, A. and Goyal, V. (2013), An FPTAS for capacity constrained assortment optimization, Technical report, Columbia University, School of Industrial Engineering and Operations Research.

Farias, V. F., Jagabathula, S. and Shah, D. (2013), 'A non-parametric approach to modeling choice with limited data', *Management Science* (to appear).

Gallego, G., Iyengar, G., Phillips, R. and Dubey, A. (2004), Managing flexible products on a network, Computational Optimization Research Center Technical Report TR-2004-01, Columbia University.

Gallego, G., Ratliff, R. and Shebalov, S. (2011), A general attraction model and an efficient formulation for the network revenue management problem, Technical report, Columbia University, New York, NY.

Gallego, G. and Topaloglu, H. (2012), Constrained assortment optimization for the nested logit model, Technical report, Cornell University, School of Operations Research and Information Engineering. Available at `http://legacy.orie.cornell.edu/~huseyin/publications/publications.html`.

Jagabathula, S., Farias, V. and Shah, D. (2011), Assortment optimization under general choice, *in* 'INFORMS Conference, Charlotte, NC'.

Kleywegt, A. and Wang, X. (2013), A stochastic trust region algorithm for mixed logit type problems, *in* 'International Conference on Stochastic Programming, Bergamo, Italy'.

Kunnumkal, S., Rusmevichientong, P. and Topaloglu, H. (2009), Assortment optimization under the multinomial logit model with product costs, Technical report, Cornell University, School of Operations Research and Information Engineering.

Kunnumkal, S. and Talluri, K. (2012), A new compact linear programming formulation for choice network revenue management, Technical report, Universitat Pompeu Fabra, Barcelona, Spain.

Kunnumkal, S. and Topaloglu, H. (2008), 'A refined deterministic linear program for the network revenue management problem with customer choice behavior', *Naval Research Logistics Quarterly* **55**(6), 563–580.

Li, G. and Rusmevichientong, P. (2012), Technical note: A simple greedy algorithm for assortment optimization in the two-level nested logit model, Technical report, University of Southern California, Marshall School of Business.

Li, G., Rusmevichientong, P. and Topaloglu, H. (2013), The *d*-level nested logit model: Assortment and price optimization problems, Technical report, Cornell University, School of Operations Research and Information Engineering. Available at
`http://legacy.orie.cornell.edu/~huseyin/publications/publications.html`.

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.

McFadden, D. (1974), Conditional logit analysis of qualitative choice behavior, *in* P. Zarembka, ed., '*Frontiers in Economics*', Academic Press, pp. 105–142.

McFadden, D. and Train, K. (2000), 'Mixed MNL models for discrete response', *Journal of Applied Economics* **15**, 447–470.

Meissner, J., Strauss, A. and Talluri, K. (2012), 'An enhanced concave program relaxation for choice network revenue management', *Production and Operations Management* **22**(1), 71–87.

Mendez-Diaz, I., Bront, J. J. M., Vulcano, G. and Zabala, P. (2010), 'A branch-and-cut algorithm for the latent-class logit assortment problem', *Discrete Applied Mathematics* **36**, 383–390.

Rusmevichientong, P., Shen, Z.-J. M. and Shmoys, D. B. (2010), 'Dynamic assortment optimization with a multinomial logit choice model and capacity constraint', *Operations Research* **58**(6), 1666–1680.

Rusmevichientong, P., Shmoys, D. B., Tong, C. and Topaloglu, H. (2013), 'Assortment optimization under the multinomial logit model with random choice parameters', *Production and Operations Management* (to appear).

Ruszczynski, A. (2006), *Nonlinear Optimization*, Princeton University Press, Princeton, New Jersey.

Sinha, P. and Zoltners, A. A. (1979), 'The multiple-choice knapsack problem', *Operations Research* **27**(3), 503–515.

Talluri, K. (2011), A randomized concave programming method for choice network revenue management, Technical report, Universitat Pompeu Fabra, Barcelona, Spain.

Talluri, K. and van Ryzin, G. (2004), 'Revenue management under a general discrete choice model of consumer behavior', *Management Science* **50**(1), 15–33.

Train, K. (2003), *Discrete Choice Methods with Simulation*, Cambridge University Press, Cambridge, UK.

Vossen, T. W. M. and Zhang, D. (2013), Reductions of approximate linear programs for network revenue management, Technical report, University of Colorado at Boulder, Boulder, CO.

Vulcano, G., van Ryzin, G. J. and Ratliff, R. (2012), 'Estimating primary demand for substitutable products from sales transaction data', *Operations Research* **60**(2), 313–334.

Wang, R. (2013), 'Assortment management under the generalized attraction model with a capacity constraint', *Journal of Revenue and Pricing Management* **12**(3), 254–270.

Williamson, D. P. and Shmoys, D. B. (2011), *The Design of Approximation Algorithms*, Cambridge University Press, New York.

Zhang, D. and Adelman, D. (2009), 'An approximate dynamic programming approach to network revenue management with customer choice', *Transportation Science* **42**(3), 381–394.