Assortment Optimization under the Multinomial Logit Model with Sequential Offerings

Nan Liu[1], Yuhang Ma[2], Huseyin Topaloglu[2]
[1]Carroll School of Management, Boston College, Chestnut Hill, MA 02467, USA
[2]School of Operations Research and Information Engineering, Cornell Tech, New York City, NY 10044, USA
nan.liu@bc.edu, ym367@cornell.edu, ht88@cornell.edu

May 8, 2019

Abstract

We consider assortment optimization problems, where the choice process of a customer takes place in multiple stages. There is a finite number of stages. In each stage, we offer an assortment of products that does not overlap with the assortments offered in the earlier stages. If the customer makes a purchase within the offered assortment, then the customer leaves the system with the purchase. Otherwise, the customer proceeds to the next stage, where we offer another assortment. If the customer reaches the end of the last stage without a purchase, then the customer leaves the system without a purchase. The choice of the customer in each stage is governed by a multinomial logit model. The goal is to find an assortment to offer in each stage to maximize the expected revenue obtained from a customer. For this assortment optimization problem, it turns out that the union of the optimal assortments to offer in each stage is nested by revenue, in the sense that this union includes a certain number of products with the largest revenues. However, it is still difficult to figure out the stage in which a certain product should be offered. In particular, the problem of finding an assortment to offer in each stage to maximize the expected revenue obtained from a customer is NP-hard. We give a fully polynomial-time approximation scheme for the problem when the number of stages is fixed.

1 Introduction

In traditional revenue management models, it is common to model the demand for each product by using an exogenous random variable that does not depend on what other products are made available to the customers. In many retail settings, however, customers choose and substitute among the products that are offered to them, in which case, the demand for a product depends on what other products are made available to the customer. There is a recent surge of revenue management models that explicitly capture such a customer choice process. Nevertheless, much of the work in this stream of literature assumes that the customers view the whole assortment of products offered to them simultaneously, but it is not difficult to run into cases where the customers gradually view the assortment in multiple stages. When selling products in online retail, for example, the firm may display the search results to a customer sequentially through multiple webpages. In this case, the goal is to decide which products to offer on each page of search results and in which order to present the pages, to maximize the expected revenue obtained from a customer. When scheduling healthcare appointments over the phone, a reasonable objective for the service provider is to maximize the probability that a patient books an appointment. To gently guide the patient through the choice process, the service provider may offer sets of appointment slots sequentially. In this case, the goal is to decide which sets of appointment slots to offer to the patient and in which order to offer the sets, to maximize the probability that the patient books an appointment slot. Thus, if the customers gradually view the assortment in multiple stages, then we need to decide not only what assortment of products to offer, but also the order in which we should offer the products. In addition, it may be necessary to limit the number of products offered in each stage, for example, to ensure that the limited space on the webpage can accommodate the products offered in each stage. Similarly, to avoid overwhelming the patient with a large number of options, it may be desirable to limit the total number of appointment slots offered over all stages.

In this paper, we consider assortment optimization problems, where the choice process of a customer takes place in multiple stages. There is a finite number of stages in the choice process, which is fixed a priori. In each stage of the choice process, we offer an assortment of products that does not overlap with the assortments offered in the previous stages. If the customer makes a purchase within the assortment offered in the current stage, then she leaves the system with a purchase. Otherwise, the customer proceeds to the next stage. If the customer reaches the end of the last stage without a purchase, then she leaves the system without a purchase. We use the multinomial logit model to capture the choice process of the customer in each stage. The goal is to find an assortment to offer in each stage to maximize the expected revenue obtained from a customer. In our initial treatment of the assortment optimization problem, we focus on the case where there is no limit on the number of products that we can offer in any stage, but we also discuss the case where there is a limit on the number of products offered in each stage or there is a limit on the total number of products offered over all stages.

Main Contributions. We show that the problem of finding an assortment to offer in each stage to maximize the expected revenue obtained from a customer is NP-hard (Theorem 1). Motivated by this result, we develop a fully polynomial-time approximation scheme (FPTAS) for the problem as follows. First, we cast our assortment optimization problem as maximizing a nonlinear function over a certain feasible set $\mathcal{P}$, but checking whether a given point is in $\mathcal{P}$ requires finding a feasible solution to an intractable multiple knapsack problem (Lemma 2). Second, we give an approximate version of the feasible set $\mathcal{P}$ by aligning the cumulative capacity consumptions of the products in the multiple knapsack to a geometric grid. Using $\widetilde{\mathcal{P}}$ to denote the approximate version of the feasible set, we bound the loss in the expected revenue when we maximize the nonlinear function over the approximate feasible set $\widetilde{\mathcal{P}}$ (Theorem 5). Third, we show that we can use a dynamic program to check whether a given point is in $\widetilde{\mathcal{P}}$ and we enumerate over the elements of $\widetilde{\mathcal{P}}$ to maximize the nonlinear function over the approximate feasible set $\widetilde{\mathcal{P}}$. Accounting for the number of operations to solve the dynamic program, we get our FPTAS (Theorem 7).

Letting n be the number of products among which we choose a sequence of assortments and m be the number of stages, for any $\epsilon \in (0, 1)$, our FPTAS runs in $O(mn^{2m}(\log(n\kappa))^{m-1}\,(\log(n\nu))^m/\epsilon^{2m-1})$ operations to provide a $(1-\epsilon)$-approximate solution. Here, κ is the largest value for the product of the revenue and preference weight of a product and ν is the largest value for the preference weight of a product, after normalizing the smallest revenue and preference weight to one. Thus, for a fixed number of stages, the running time of our FPTAS is polynomial in input size and the reciprocal of the precision. If we have a limit on the number of products offered in each stage, then the term n^{2m} in the running time is replaced by n^{3m}, whereas if we have a limit on the total number of products offered in all stages, then the term n^{2m} in the running time is replaced by n^{2m+1}.

We also provide some insight into the form of the optimal assortment. Gallego et al. (2004) and Talluri and van Ryzin (2004) show that if there is a single stage in the choice process, then the optimal assortment is nested by revenue, including a certain number of products with the largest revenues. So, we can efficiently find the optimal assortment by checking the expected revenue from each nested by revenue assortment. We show that if there are multiple stages, then the union of the optimal assortments to offer in each stage is nested by revenue. However, this result does not allow us to find the optimal assortment efficiently, since it does not characterize the stage in which each product should be offered. Thus, we defer this result to the appendix.

Lastly, in our assortment optimization problem, we fix the assortments that we offer in all of the stages a priori. A natural question is whether there is any value in adjusting the assortment offered in a particular stage based on the choice trajectory of a customer in the previous stages. Note that if a customer is in a particular stage in the choice process, then her choices in all of the previous stages must have been no purchase. Therefore, given that a customer is in a particular stage in the choice process, there is only one possible choice trajectory of the customer in the previous stages, indicating that there is no value in adjusting the assortment offered in a particular stage based on the choice trajectory of a customer in the previous stages.

Literature Review. There is some work on assortment optimization problems, where the assortment is offered in multiple stages. Gallego et al. (2016) study a problem in online retail, where the assortments of products are presented in multiple pages. Each customer picks the number of pages to view according to a distribution that is exogenously fixed and she chooses among all of the products offered on those pages. The authors give an approximation algorithm. Focusing on healthcare appointment scheduling on the phone, Liu et al. (2017) consider the case where the service provider offers assortments of appointment slots to a patient in multiple stages. If a patient is offered an assortment that includes appointment slots she is interested in, then she chooses among them uniformly. The authors characterize the optimal sequence of appointment slots to offer. Flores et al. (2019) work with a two stage multinomial logit model, where there are two disjoint sets of products that can potentially be offered in the two stages. Thus, if a product is offered, then the stage in which it will be offered is fixed a priori. Focusing on the case with two stages, the authors give an efficient optimal algorithm. In our model, if we decide to offer a product, then we still need to choose the stage in which to offer the product. As a result, our problem is NP-hard. Furthermore, we work with multiple stages.

As mentioned above, Gallego et al. (2004) and Talluri and van Ryzin (2004) show that the assortment optimization problem under the multinomial logit model can be solved efficiently when there is a single stage. Rusmevichientong et al. (2010) show how to solve the same assortment optimization problem when there is a cardinality constraint limiting the total number of offered products. Rusmevichientong and Topaloglu (2012) study robust assortment optimization problems under the multinomial logit model when the parameters of the model are not known, but they take values in an uncertainty set. Wang (2012) considers the problem of jointly finding an assortment of products to offer and their corresponding prices, when the customers choose under the multinomial logit model and there is a constraint on the total number of offered products. Davis et al. (2013) give a linear programming formulation when the constraints have a totally unimodular structure. Wang (2013) and Gallego et al. (2015) solve assortment optimization problems under a more general version of the multinomial logit model, where the preference weight of the no purchase option increases as a function of the products that are not offered.

In a mixture of multinomial logit models, we have multiple customer types and customers of different types choose according to different multinomial logit models. Bront et al. (2009), Mendez-Diaz et al. (2010) and Rusmevichientong et al. (2014) show that the assortment optimization problem under a mixture of multinomial logit models is NP-hard, provide integer programming formulations, study special cases admitting efficient solutions and give approximations. Meissner et al. (2012) provide an upper bound on the optimal expected revenue by using a relaxation that offers different assortments to customers of different types. The objective function of our assortment optimization problem is a sum of fractions. Considering assortment optimization problems with multiple customer types, Mittal and Schulz (2013), Desir and Goyal (2014) and Feldman and Topaloglu (2018) design FPTAS for maximizing various sums of fractions. In our FTPAS, we draw on Desir and Goyal (2014), where the authors use the connections of their problem to the knapsack problem by aligning the cumulative capacity consumptions to a geometric grid. Due to the multiple stages in our choice process, the numerators and denominators in our fractions are nonlinear and we need to carefully account for the errors resulting from the geometric grid. The error is exponential in the number of products, but by a judicious choice of the performance guarantee, we get an FPTAS. For representative assortment optimization work under other choice models, we refer the reader to Davis et al. (2014) for the nested logit model, Aouad et al. (2016) for the ranking-based choice model and Blanchet et al. (2016) for the Markov chain choice model.

Organization. In Section 2, we formulate our assortment optimization problem with multiple stages and show that it is NP-hard. In the rest of this paper, we focus on developing an FPTAS. In Section 3, we give an alternative formulation of our assortment optimization problem that maximizes a nonlinear function over a certain feasible set $\mathcal{P}$. Checking whether a given point is in $\mathcal{P}$ is difficult. In Section 4, we give an approximation to the feasible set $\mathcal{P}$. We refer to this approximation as $\widetilde{\mathcal{P}}$. In Section 5, we bound the loss in the expected revenue when we maximize the nonlinear function over the approximate feasible set $\widetilde{\mathcal{P}}$. In Section 6, we give a dynamic program to check whether a given point is in $\widetilde{\mathcal{P}}$. In Section 7, by accounting for the number of operations to solve the dynamic program and to enumerate over the elements of $\widetilde{\mathcal{P}}$, we give our FPTAS. In Section 8, we make extensions to the case where there is a limit on the number of products that we can offer. In Section 9, we give numerical experiments. In Section 10, we conclude.

2 Problem Formulation and Complexity

We have n products indexed by $N = \{1, \ldots, n\}$. In the choice process, we have m stages indexed by $M = \{1, \ldots, m\}$. We use the set $S^k \subseteq N$ to denote the set of products that we offer in stage k. Since we can offer a product in at most one stage, the feasible sets of products that we can offer over all stages are $\mathcal{F} = \{(S^1, \ldots, S^m) : S^k \subseteq N \; \forall\, k \in M,\; S^k \cap S^\ell = \varnothing \; \forall\, k \neq \ell\}$. In the choice process, a customer chooses within the offered set of products in each stage according to the multinomial logit model and the choices of a customer in different stages are independent. In particular, we use $v_i^k > 0$ to denote the preference weight of product i when this product is offered in stage k. Normalizing the preference weight of the no purchase option in each stage to one, if we offer the sets of products $(S^1, \ldots, S^m)$ over all of the stages, then a customer in stage k chooses product i with probability $\mathbf{1}(i \in S^k)\, v_i^k / (1 + \sum_{j \in S^k} v_j^k)$, where $\mathbf{1}(\cdot)$ is the indicator function. A customer in stage k does not make a purchase with probability $1/(1 + \sum_{j \in S^k} v_j^k)$, in which case, she moves on to stage $k+1$. If a customer does not make a purchase by the end of the last stage m, then she leaves the system without a purchase. For a customer to purchase product i in stage k, she needs to not make a purchase in stages $1, \ldots, k-1$ and she needs to purchase product i in stage k. Therefore, if the sets of products offered over all stages are $(S^1, \ldots, S^m)$, then a customer purchases product i in stage k with probability $\prod_{\ell=1}^{k-1} \frac{1}{1+\sum_{j \in S^\ell} v_j^\ell} \times \frac{\mathbf{1}(i \in S^k)\, v_i^k}{1+\sum_{j \in S^k} v_j^k}$. The revenue associated with product i is $r_i > 0$. Our goal is to find sets of products to offer over all stages to maximize the expected revenue obtained from a customer, which yields the problem

$$\begin{aligned}
\widehat{Z} \;&=\; \max_{(S^1,\ldots,S^m) \in \mathcal{F}} \left\{ \sum_{i \in N} \sum_{k \in M} r_i \left\{ \prod_{\ell=1}^{k-1} \frac{1}{1 + \sum_{j \in S^\ell} v_j^\ell} \right\} \frac{\mathbf{1}(i \in S^k)\, v_i^k}{1 + \sum_{j \in S^k} v_j^k} \right\} \\
&= \max_{(S^1,\ldots,S^m) \in \mathcal{F}} \left\{ \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1 + \sum_{i \in S^\ell} v_i^\ell} \right\} \sum_{i \in S^k} r_i\, v_i^k \right\}, \qquad (1)
\end{aligned}$$

where the second equality follows simply by arranging the terms. Note that we can interpret $\prod_{\ell=1}^{k} \frac{1}{1+\sum_{i \in S^\ell} v_i^\ell} \times \sum_{i \in S^k} r_i\, v_i^k$ as the expected revenue in stage k.

Computational Complexity. Next, we characterize the computational complexity of problem (1). In particular, we focus on the feasibility version of problem (1), where the goal is to find a solution $(S^1, \ldots, S^m) \in \mathcal{F}$ with an expected revenue that is no smaller than a given threshold. In the next theorem, we show that the feasibility version of problem (1) is NP-complete.

Theorem 1 *The feasibility version of problem* (1) *is NP-complete.*

Proof. The problem is in NP. We use a reduction from the partition problem, which is a well-known NP-complete problem; see Garey and Johnson (1979). In the partition problem, we have n products indexed by $N = \{1, \ldots, n\}$. The weight of product i is c_i. The weight of each product is an integer and we have $\sum_{i \in N} c_i = 2\,t$. The goal of the partition problem is to find a set of products

S such that $\sum_{i \in S} c_i = \sum_{i \in N \setminus S} c_i = t$. Using the partition problem, we construct an instance of the feasibility version of problem (1) as follows. We have n products indexed by $N = \{1, \ldots, n\}$ and two stages indexed by $M = \{1, 2\}$. The revenue of product i is $r_i = 1$ for all $i \in N$. The preference weight of product i in stage k is $v_i^k = c_i / t$ for all $i \in N$, $k \in M$. The expected revenue threshold is $3/4$. We proceed to showing that there exists $(S_1, S_2) \in \mathcal{F}$ such that the expected revenue from the solution (S_1, S_2) is $3/4$ or more if and only if there exists $S \subseteq N$ such that $\sum_{i \in S} c_i = t$. Noting the expression for the expected revenue in problem (1), for the solution $(S_1, S_2) \in \mathcal{F}$ to provide an expected revenue of $3/4$ or more, this solution must satisfy the inequality

$$\frac{\sum_{i \in S_1} c_i / t}{1 + \sum_{i \in S_1} c_i / t} + \frac{1}{1 + \sum_{i \in S_1} c_i / t} \times \frac{\sum_{i \in S_2} c_i / t}{1 + \sum_{i \in S_2} c_i / t} \geq \frac{3}{4},$$

which is, arranging the terms, equivalent to $(t + \sum_{i \in S_1} c_i)\,(t + \sum_{i \in S_2} c_i) \geq 4\,t^2$. Also, if $(S_1, S_2) \in \mathcal{F}$, then we have $S_1 \cap S_2 = \varnothing$, which implies that $\sum_{i \in S_1} c_i + \sum_{i \in S_2} c_i \leq \sum_{i \in N} c_i = 2\,t$, so that $\sum_{i \in S_2} c_i \leq 2\,t - \sum_{i \in S_1} c_i$. In this case, if $(S_1, S_2) \in \mathcal{F}$, then we have $(t + \sum_{i \in S_1} c_i)\,(t + \sum_{i \in S_2} c_i) \leq (t + \sum_{i \in S_1} c_i)\,(3\,t - \sum_{i \in S_1} c_i) = 4\,t^2 - (\sum_{i \in S_1} c_i - t)^2 \leq 4\,t^2$. Thus, for the solution $(S_1, S_2) \in \mathcal{F}$ to provide an expected revenue of $3/4$ or more, the last chain of inequalities must hold as equalities, which happens only when $\sum_{i \in S_1} c_i = t$ and $\sum_{i \in S_2} c_i = \sum_{i \in N \setminus S_1} c_i = t$. So, there exists $(S_1, S_2) \in \mathcal{F}$ such that the expected revenue from the solution (S_1, S_2) is $3/4$ or more if and only if there exists $S \subseteq N$ such that $\sum_{i \in S} c_i = \sum_{i \in N \setminus S} c_i = t$. $\square$

Thus, problem (1) is NP-hard even when there are only two stages in the choice process with $v_i^1 = v_i^2$ for all $i \in N$ and $r_i = 1$ for all $i \in N$. Note that if the revenues of all products are one, then the objective function of problem (1) is the probability that a customer makes a purchase. Before us, to show that the assortment optimization problem under a mixture of multinomial logit models is NP-hard, Rusmevichientong et al. (2014) use a reduction from the partition problem, but the specifics of their reduction are different. Also, if the revenues of all products are one, then their assortment optimization problem has a trivial optimal solution that offers all products. Our assortment optimization problem is NP-hard even when the revenues of all products are one.

Random Utility Maximization. In random utility maximization, a customer associates random utilities with the products and the no purchase option, choosing the alternative with the largest utility. We can justify our choice model by using random utility maximization. The utility of purchasing product i in stage k is U_i^k. For the vector of utilities $(U_i^1, \ldots, U_i^m)$ associated with product i in different stages, the marginal distribution of U_i^k is Gumbel with location and scale parameters $(\mu_i^k, 1)$, but the different components of the vector can be dependent. Through the dependence between the utilities associated with a product in different stages, we can capture the situation where if a customer favors a certain product in a certain stage, then she is likely to favor this product in other stages as well. The utility of not purchasing anything in stage k is U_0^k. For the vector of utilities $(U_0^1, \ldots, U_0^m)$ associated with the no purchase option in different stages, the marginal distribution of U_0^k is Gumbel with location and scale parameters $(\mu_0^k, 1)$, but the different components of the vector are independent. Not having any dependence between the utilities associated with the no purchase option in different stages yields a tractable expression for the choice probabilities and it is partially motivated by the fact that the no purchase options in different stages are very different since the tendency of a particular customer not to purchase in a particular stage may depend on how many other stages are to follow, if any. (In any case, one can certainly and admittedly argue that if a customer favors the no purchase option in a certain stage, then she is also likely to favor this option in other stages.) Also, the location parameter of the utility associated with the no purchase option in different stages can be different, once again, indicating that not purchasing in a later stage when the choice process is about to terminate can have a different mean utility implication than not purchasing in an earlier stage. For two products $i \neq j$, the vectors $(U_i^1, \ldots, U_i^m)$ and $(U_j^1, \ldots, U_j^m)$ are independent. Considering the independent random variables $\{X_i : i \in G\}$ for some generic index set G, if X_i has a Gumbel distribution with location and scale parameters $(\beta_i, 1)$, then $\mathbb{P}\{X_i = \max_{j \in G} X_j\} = e^{\beta_i} / \sum_{j \in G} e^{\beta_j}$. If we offer the sets of products $(S^1, \ldots, S^m)$ over all stages, then for a customer to purchase product $i \in S^k$ in stage k, she needs to not make a purchase in stages $1, \ldots, k-1$ and she needs to purchase product i in stage k, in which case, a customer purchases product $i \in S^k$ in stage k with probability

$$\mathbb{P}\Big\{U_0^\ell = \max_{j \in S^\ell \cup \{0\}} U_j^\ell \;\; \forall\, \ell = 1, \ldots, k-1 \text{ and } U_i^k = \max_{j \in S^k \cup \{0\}} U_j^k\Big\} = \prod_{\ell=1}^{k-1} \frac{e^{\mu_0^\ell}}{\sum_{j \in S^\ell \cup \{0\}} e^{\mu_j^\ell}} \times \frac{e^{\mu_i^k}}{\sum_{j \in S^k \cup \{0\}} e^{\mu_j^k}},$$

where we use the fact that the sets of products offered in the different stages are disjoint so that the events on the left side above are independent. Letting $e^{\mu_i^k - \mu_0^k} = v_i^k$ and multiplying the numerator and denominator of all of the fractions on the right side above by $e^{-\mu_0^\ell}$ for $\ell = 1, \ldots, k$, we obtain the choice probability $\prod_{\ell=1}^{k-1} \frac{1}{1+\sum_{j \in S^\ell} v_j^\ell} \times \frac{v_i^k}{1+\sum_{j \in S^k} v_j^k}$.

In the rest of the paper, noting Theorem 1, we focus on developing an FPTAS.

3 Multiple Knapsack Representation

Noting the objective function in (1), intuitively speaking, a good solution should keep the quantity $\sum_{i \in S^k} r_i \, v_i^k$ large and the quantity $\sum_{i \in S^k} v_i^k$ small for all $k \in M$. This observation motivates the following approach. First, we guess lower bounds on the quantity $\sum_{i \in S^k} r_i \, v_i^k$ and upper bounds on the quantity $\sum_{i \in S^k} v_i^k$ for all $k \in M$. Second, we check whether there exists a solution $(S^1, \ldots, S^m) \in \mathcal{F}$ that satisfies our guesses. Carrying out an exhaustive search over our guesses, we pick the best solution. To pursue this approach, we use $\mathcal{P}$ to denote the set of vectors $\boldsymbol{f} = (f^1, \ldots, f^m)$ and $\boldsymbol{h} = (h^1, \ldots, h^m)$ such that there exists a solution $(S^1, \ldots, S^m) \in \mathcal{F}$ satisfying $\sum_{i \in S^k} r_i \, v_i^k \geq f^k$ and $\sum_{i \in S^k} v_i^k \leq h^k$ for all $k \in M$. Thus, $\mathcal{P}$ is given by

$$\mathcal{P} = \Big\{(\boldsymbol{f}, \boldsymbol{h}) \in \Re_+^m \times \Re_+^m \;:\; \exists\, (S^1, \ldots, S^m) \in \mathcal{F} \text{ that satisfies} \\ \sum_{i \in S^k} r_i \, v_i^k \geq f^k \;\; \forall\, k \in M \;\text{ and }\; \sum_{i \in S^k} v_i^k \leq h^k \;\; \forall\, k \in M\Big\}. \tag{2}$$

Note that the two sets of constraints that need to be satisfied by $(S^1, \ldots, S^m)$ above are similar to multiple knapsack constraints. Noting the objective function of problem (1), if we have

$\sum_{i \in S^k} r_i\, v_i^k \geq f^k$ and $\sum_{i \in S^k} v_i^k \leq h^k$ for all $k \in M$, then the solution $(S^1, \ldots, S^m)$ provides an expected revenue of at least $\sum_{k \in M} \prod_{\ell=1}^k \frac{1}{1+h^\ell} \times f^k$. Therefore, we consider the problem

$$\max_{(\boldsymbol{f},\boldsymbol{h}) \in \mathcal{P}} \left\{ \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+h^\ell} \right\} f^k \right\}. \tag{3}$$

In the next lemma, we show that the optimal objective value of problem (1) corresponds to the optimal objective value of problem (3).

Lemma 2 *The optimal objective value of problem* (1) *is equal to the optimal objective value of problem* (3).

Proof. Let $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ be an optimal solution to problem (3) providing the optimal objective value $\widehat{\zeta}$. We have $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathcal{P}$, in which case, by the definition of $\mathcal{P}$, there exists a solution $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ such that $\sum_{i \in \widehat{S}^k} r_i\, v_i^k \geq \widehat{f}^k$ and $\sum_{i \in \widehat{S}^k} v_i^k \leq \widehat{h}^k$ for all $k \in M$. Let $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ be an optimal solution to problem (1) providing the optimal objective value $\widehat{Z}$. Define $\widetilde{f}^k = \sum_{i \in \widetilde{S}^k} r_i\, v_i^k$ and $\widetilde{h}^k = \sum_{i \in \widetilde{S}^k} v_i^k$ for all $k \in M$. Note that $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) \in \mathcal{P}$ since $\sum_{i \in \widetilde{S}^k} r_i\, v_i^k \geq \widetilde{f}^k$ and $\sum_{i \in \widetilde{S}^k} v_i^k \leq \widetilde{h}^k$ for all $k \in M$. Therefore, $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ is a feasible solution to problem (3). Since $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ is an optimal solution to problem (3), we get $\sum_{k \in M} \prod_{\ell=1}^k \frac{1}{1+\widehat{h}^\ell} \times \widehat{f}^k = \widehat{\zeta} \geq \sum_{k \in M} \prod_{\ell=1}^k \frac{1}{1+\widetilde{h}^\ell} \times \widetilde{f}^k$. In this case, noting that we have $\sum_{i \in \widehat{S}^k} r_i\, v_i^k \geq \widehat{f}^k$ and $\sum_{i \in \widehat{S}^k} v_i^k \leq \widehat{h}^k$, the objective value provided by the solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ for problem (1) satisfies

$$\begin{aligned}\sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+\sum_{i \in \widehat{S}^\ell} v_i^\ell} \right\} \textstyle\sum_{i \in \widehat{S}^k} r_i\, v_i^k &\geq \displaystyle\sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+\widehat{h}^k} \right\} \widehat{f}^k \\ &\geq \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+\widetilde{h}^\ell} \right\} \widetilde{f}^k = \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+\sum_{i \in \widetilde{S}^\ell} v_i^\ell} \right\} \textstyle\sum_{i \in \widetilde{S}^k} r_i\, v_i^k = \widehat{Z}.\end{aligned}$$

Since $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$, the left side above is at most the optimal objective value of problem (1), which is $\widehat{Z}$. Thus, all of the inequalities above hold as equalities and we get $\widehat{Z} = \widehat{\zeta}$. $\square$

Using the lemma above, we can try to solve problem (1) in two steps. First, we find an optimal solution $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ to problem (3). Second, we find $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ that satisfies $\sum_{i \in \widehat{S}^k} r_i\, v_i^k \geq \widehat{f}^k$ and $\sum_{i \in \widehat{S}^k} v_i^k \leq \widehat{h}^k$ for all $k \in M$. By the discussion right before problem (3), the expected revenue from the solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ is at least $\sum_{k \in M} \prod_{\ell=1}^k \frac{1}{1+\widehat{h}^\ell} \times \widehat{f}^k$, which is, by Lemma 2, equal to the optimal objective value of problem (1). Both of these two steps are computationally difficult. In particular, the objective function of problem (3) is not necessarily concave and finding $(\widehat{S}^1, \ldots, \widehat{S}^m)$ satisfying the last two inequalities is a combinatorial problem. In our FPTAS, we carry out these two steps approximately. First, we use a geometric grid over $\Re_+^m \times \Re_+^m$ to check the objective value of problem (3) at a limited number of guesses for $(\boldsymbol{f}, \boldsymbol{h})$. Second, we give an approximate version of the set $\mathcal{P}$, in which case, we can use a dynamic program to find $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ that approximately satisfies $\sum_{i \in \widehat{S}^k} r_i\, v_i^k \geq \widehat{f}^k$ and $\sum_{i \in \widehat{S}^k} v_i^k \leq \widehat{h}^k$ for all $k \in M$. In the next section, we construct the geometric grid and the approximate version of the set $\mathcal{P}$.

4 Approximate Feasible Set

To give an approximation to the set $\mathcal{P}$, we begin by computing $\sum_{i \in S^k} r_i\, v_i^k$ and $\sum_{i \in S^k} v_i^k$ recursively. For $f^k \in \Re_+$, $h^k \in \Re_+$ and $S^k \subseteq N$, we define $F_i^k(f^k, S^k)$ and $H_i^k(h^k, S^k)$ recursively as

$$\begin{aligned} F_{i+1}^k(f^k, S^k) &= F_i^k(f^k, S^k) - r_i\, v_i^k\, \mathbf{1}(i \in S^k) \\ H_{i+1}^k(h^k, S^k) &= H_i^k(h^k, S^k) - v_i^k\, \mathbf{1}(i \in S^k) \end{aligned} \tag{4}$$

with the initial condition that $F_1^k(f^k, S^k) = f^k$ and $H_1^k(h^k, S^k) = h^k$. Adding the first equality above over all $i \in N$ and noting that $F_1^k(f^k, S^k) = f^k$, we obtain $F_{n+1}^k(f^k, S^k) = f^k - \sum_{i \in N} r_i\, v_i^k\, \mathbf{1}(i \in S^k) = f^k - \sum_{i \in S^k} r_i\, v_i^k$. Therefore, we have $\sum_{i \in S^k} r_i\, v_i^k \geq f^k$ if and only if we have $F_{n+1}^k(f^k, S^k) \leq 0$. Similarly, we have $\sum_{i \in S^k} v_i^k \leq h^k$ if and only if we have $H_{n+1}^k(h^k, S^k) \geq 0$. In this case, we can replace the condition $\sum_{i \in S^k} r_i\, v_i^k \geq f^k$ and $\sum_{i \in S^k} v_i^k \leq h^k$ for all $k \in M$ in (2) with the condition $F_{n+1}^k(f^k, S^k) \leq 0$ and $H_{n+1}^k(h^k, S^k) \geq 0$ for all $k \in M$ to express the set $\mathcal{P}$ equivalently. In other words, the set $\mathcal{P}$ is also given by

$$\mathcal{P} = \Big\{ (\boldsymbol{f}, \boldsymbol{h}) \in \Re_+^m \times \Re_+^m \ : \ \exists\, (S^1, \ldots, S^m) \in \mathcal{F} \text{ that satisfies} \\ F_{n+1}^k(f^k, S^k) \leq 0 \ \ \forall\, k \in M \ \text{ and } \ H_{n+1}^k(h^k, S^k) \geq 0 \ \ \forall\, k \in M \Big\}. \tag{5}$$

By restricting the values of $F_i^k(f^k, S^k)$ and $H_i^k(h^k, S^k)$ on a geometric grid, we proceed to giving an approximate version of the set $\mathcal{P}$.

For fixed $\rho > 0$, we define $\mathsf{Dom} = \{(1+\rho)^\ell : \ell = \ldots, -1, 0, 1, \ldots\} \cup \{-\infty, 0\}$, which is a geometric grid augmented by the points $\{-\infty, 0\}$. We define the round up operator $\lceil \cdot \rceil$ that rounds its argument up to the nearest element of Dom. In particular, $\lceil x \rceil = \min\{y \in \mathsf{Dom} : y \geq x\}$. Similarly, we define the round down operator $\lfloor \cdot \rfloor$ that rounds its argument down to the nearest element of Dom. Therefore, we have $\lfloor x \rfloor = \max\{y \in \mathsf{Dom} : y \leq x\}$. Note that if $x < 0$, then $\lceil x \rceil = 0$ and $\lfloor x \rfloor = -\infty$. We use $\Phi_i^k(f^k, S^k)$ and $\Gamma_i^k(h^k, S^k)$ to denote approximate versions of $F_i^k(f^k, S^k)$ and $H_i^k(h^k, S^k)$, which are also defined recursively as

$$\begin{aligned} \Phi_{i+1}^k(f^k, S^k) &= \big\lceil \Phi_i^k(f^k, S^k) - r_i\, v_i^k\, \mathbf{1}(i \in S^k) \big\rceil \\ \Gamma_{i+1}^k(h^k, S^k) &= \big\lfloor \Gamma_i^k(h^k, S^k) - v_i^k\, \mathbf{1}(i \in S^k) \big\rfloor \end{aligned} \tag{6}$$

with the initial condition that $\Phi_1^k(f^k, S^k) = f^k$ and $\Gamma_1^k(h^k, S^k) = h^k$. Noting the round up operator in the definition of $\Phi_{i+1}^k(f^k, S^k)$, we get $\Phi_{i+1}^k(f^k, S^k) \geq \Phi_i^k(f^k, S^k) - r_i\, v_i^k\, \mathbf{1}(i \in S^k)$. If we add this inequality over all $i \in N$ and note that $\Phi_1^k(f^k, S^k) = f^k$, then we obtain $\Phi_{n+1}^k(f^k, S^k) \geq f^k - \sum_{i \in N} r_i\, v_i^k\, \mathbf{1}(i \in S^k) = f^k - \sum_{i \in S^k} r_i\, v_i^k$. Therefore, if $\Phi_{n+1}^k(f^k, S^k) \leq 0$, then we have $f^k - \sum_{i \in S^k} r_i\, v_i^k \leq 0$, which implies that $F_{n+1}^k(f^k, S^k) \leq 0$ as well. Using a similar argument, we also obtain $\Gamma_{n+1}^k(h^k, S^k) \leq h^k - \sum_{i \in S^k} v_i^k$. In this case, if $\Gamma_{n+1}^k(h^k, S^k) \geq 0$, then we have $h^k - \sum_{i \in S^k} v_i^k \geq 0$, which implies that $H_{n+1}^k(h^k, S^k) \geq 0$ as well.

By the discussion in the previous paragraph, if the vector $(\boldsymbol{f}, \boldsymbol{h}) \in \Re_+^m \times \Re_+^m$ satisfies $\Phi_{n+1}^k(f^k, S^k) \leq 0$ and $\Gamma_{n+1}^k(h^k, S^k) \geq 0$ for all $k \in M$, then it also satisfies $F_{n+1}^k(f^k, S^k) \leq 0$

and $H^k_{n+1}(h^k, S^k) \geq 0$ for all $k \in M$. Thus, we can define a restricted version of the set $\mathcal{P}$ using $\Phi^k_{n+1}(f^k, S^k)$ and $\Gamma^k_{n+1}(h^k, S^k)$. Denoting this restricted version by $\widetilde{\mathcal{P}}$, we have

$$\widetilde{\mathcal{P}} = \Big\{ (\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}^m_+ \times \mathsf{Dom}^m_+ \;:\; \exists\, (S^1, \ldots, S^m) \in \mathcal{F} \text{ that satisfies} \\ \Phi^k_{n+1}(f^k, S^k) \leq 0 \;\; \forall\, k \in M \;\text{ and }\; \Gamma^k_{n+1}(h^k, S^k) \geq 0 \;\; \forall\, k \in M \Big\}, \quad (7)$$

where we use $\mathsf{Dom}_+ = \mathsf{Dom} \setminus \{-\infty\}$. By the discussion above, $\widetilde{\mathcal{P}} \subseteq \mathcal{P}$. In the next proposition, we show that we can perturb an element of $\mathcal{P}$ to obtain an element of $\widetilde{\mathcal{P}}$.

Proposition 3 *For any $f^k \in \mathsf{Dom}_+$, $h^k \in \mathsf{Dom}_+$ and $S^k \subseteq N$, if $F^k_{n+1}(f^k, S^k) \leq 0$, then we have $\Phi^k_{n+1}(f^k/(1+\rho)^{|S^k|}, S^k) \leq 0$. Also, if $H^k_{n+1}(h^k, S^k) \geq 0$, then we have $\Gamma^k_{n+1}((1+\rho)^{|S^k|}\, h^k, S^k) \geq 0$.*

Proof. Fix f^k and S^k. For notational brevity, we let $\Phi^k_i = \Phi^k_i(f^k/(1+\rho)^{|S^k|}, S^k)$ and $S^k_i = S^k \cap \{i, \ldots, n\}$. We follow the convention that $S^k_{n+1} = \varnothing$. By the definition of $\Phi^k_i(f^k/(1+\rho)^{|S^k|}, S^k)$, we have $\Phi^k_{i+1} = \lceil \Phi^k_i - r_i\, v^k_i\, \mathbf{1}(i \in S^k) \rceil$ with $\Phi^k_1 = f^k/(1+\rho)^{|S^k|}$. Noting the definition of S^k_i, we have $i \in S^k$ if and only if $i \in S^k_i$. Also, we have $S^k_1 = S^k$. Therefore, we can write the recursion that we use to compute Φ^k_i equivalently as $\Phi^k_{i+1} = \lceil \Phi^k_i - r_i\, v^k_i\, \mathbf{1}(i \in S^k_i) \rceil$ with $\Phi^k_1 = f^k/(1+\rho)^{|S^k_1|}$. We use induction over the products to show that $\Phi^k_i \leq \sum_{j \in S^k_i} r_j\, v^k_j/(1+\rho)^{|S^k_i|}$ or $\Phi^k_i = 0$ for all $i \in N \cup \{n+1\}$. Since $F^k_{n+1}(f^k, S^k) \leq 0$, by the discussion at the beginning of this section, we have $\sum_{j \in S^k} r_j\, v^k_j \geq f^k$. In this case, noting that $\Phi^k_1 = f^k/(1+\rho)^{|S^k_1|}$, we obtain $\Phi^k_1 = f^k/(1+\rho)^{|S^k_1|} \leq \sum_{j \in S^k_1} r_j\, v^k_j/(1+\rho)^{|S^k_1|}$, which implies that the result holds for product 1. Next, we assume that the result holds for product i, so that $\Phi^k_i \leq \sum_{j \in S^k_i} r_j\, v^k_j/(1+\rho)^{|S^k_i|}$ or $\Phi^k_i = 0$. If $\Phi^k_i = 0$, then $\Phi^k_i - r_i\, v^k_i\, \mathbf{1}(i \in S^k_i) \leq 0$, which implies that $\Phi^k_{i+1} = \lceil \Phi^k_i - r_i\, v^k_i\, \mathbf{1}(i \in S^k_i) \rceil = 0$, in which case, the result holds for product $i+1$ as well. Thus, we assume that $\Phi^k_i \leq \sum_{j \in S^k_i} r_j\, v^k_j/(1+\rho)^{|S^k_i|}$ in the rest of the induction argument. Note that if $\Phi^k_i \leq \sum_{j \in S^k_i} r_j\, v^k_j/(1+\rho)^{|S^k_i|}$, then we have

$$\Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \leq \frac{\sum_{j \in S^k_i} r_j\, v^k_j}{(1+\rho)^{|S^k_i|}} - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \\ = \frac{\sum_{j \in S^k_{i+1}} r_j\, v^k_j + \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i}{(1+\rho)^{|S^k_{i+1}| + \mathbf{1}(i \in S^k_i)}} - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \leq \frac{\sum_{j \in S^k_{i+1}} r_j\, v^k_j}{(1+\rho)^{|S^k_{i+1}| + \mathbf{1}(i \in S^k_i)}}, \quad (8)$$

where the equality uses the fact that if $i \in S^k_i$, then $S^k_i = S^k_{i+1} \cup \{i\}$ and the second inequality uses the fact that $1/(1+\rho)^{|S^k_{i+1}| + \mathbf{1}(i \in S^k_i)} \leq 1$.

If $\Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \leq 0$, then $\Phi^k_{i+1} = \lceil \Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \rceil = 0$ and the result holds for product $i+1$ as well. So, we consider the chain of inequalities in (8) under the assumption that $\Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i > 0$. First, we consider the case $i \notin S^k_i$. By (8), we obtain $\Phi^k_i \leq \sum_{j \in S^k_{i+1}} r_j\, v^k_j/(1+\rho)^{|S^k_{i+1}|}$. Furthermore, we have $\Phi^k_{i+1} = \lceil \Phi^k_i - r_i\, v^k_i\, \mathbf{1}(i \in S^k_i) \rceil = \lceil \Phi^k_i \rceil$. Lastly, since $f^k \in \mathsf{Dom}_+$, we get $\Phi^k_1 = f^k/(1+\rho)^{|S^k_1|} \in \mathsf{Dom}_+$. Since $\Phi^k_{j+1} = \lceil \Phi^k_j - r_j\, v^k_j\, \mathbf{1}(j \in S^k_j) \rceil$ for all $j \in N$ and $\Phi^k_1 \in \mathsf{Dom}_+$, we obtain $\Phi^k_j \in \mathsf{Dom}$ for all $j \in N$, so that $\lceil \Phi^k_j \rceil = \Phi^k_j$. Therefore, we have $\Phi^k_{i+1} = \lceil \Phi^k_i \rceil = \Phi^k_i \leq \sum_{j\in S^k_{i+1}} r_j\, v^k_j/(1+\rho)^{|S^k_{i+1}|}$, in which case, the result holds for product $i+1$ as well. Second, we consider the case $i \in S^k_i$. By (8), we have $\Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \leq \sum_{j\in S^k_{i+1}} r_j\, v^k_j/(1+\rho)^{|S^k_{i+1}|+1}$. For $x \geq 0$, note that $\lceil x \rceil \leq (1+\rho)\, x$. In this case, since we assume that $\Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i > 0$, by the last inequality, we obtain $\Phi^k_{i+1} = \lceil \Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \rceil \leq (1+\rho) \times (\Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i) \leq \sum_{j\in S^k_{i+1}} r_j\, v^k_j/(1+\rho)^{|S^k_{i+1}|}$, which implies that the result holds for product $i+1$ as well, completing the induction argument. Therefore, the discussion so far establishes that $\Phi^k_i \leq \sum_{j\in S^k_i} r_j\, v^k_j/(1+\rho)^{|S^k_i|}$ or $\Phi^k_i = 0$ for all $i \in N \cup \{n+1\}$. Using this result with $i = n+1$ and noting that $S^k_{n+1} = \varnothing$, we get $\Phi^k_{n+1} \leq 0$ or $\Phi^k_{n+1} = 0$. So, $\Phi^k_{n+1} = \Phi^k_{n+1}(f^k/(1+\rho)^{|S^k|}, S^k) \leq 0$, showing the first statement in the proposition. The second statement uses a similar reasoning. □

By the proposition above, if $\rho > 0$ is small, then we can perturb an element of $\mathcal{P}$ by a small amount to obtain an element of $\widetilde{\mathcal{P}}$. Noting the discussion at the end of Section 3, we can solve problem (1) by obtaining an optimal solution $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ to problem (3) and finding $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ such that $F^k_{n+1}(\widehat{f}^k, \widehat{S}^k) \leq 0$ and $H^k_{n+1}(\widehat{f}^k, \widehat{S}^k) \geq 0$ for all $k \in M$. Replacing $\mathcal{P}$ with $\widetilde{\mathcal{P}}$, $F^k_{n+1}(f^k, S^k)$ with $\Phi^k_{n+1}(f^k, S^k)$ and $H^k_{n+1}(h^k, S^k)$ with $\Gamma^k_{n+1}(f^k, S^k)$, we can approximately solve problem (1) by obtaining an optimal solution $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ to the problem $\max_{(\boldsymbol{f},\boldsymbol{h})\in\widetilde{\mathcal{P}}} \left\{ \sum_{k\in M} \prod_{\ell=1}^k \frac{1}{1+h^\ell} \times f^k \right\}$ and finding $(\widetilde{S}^1, \ldots, \widetilde{S}^m) \in \mathcal{F}$ such that $\Phi^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ and $\Gamma^k_{n+1}(\widetilde{h}^k, \widetilde{S}^k) \geq 0$ for all $k \in M$. In the next section, we give a performance guarantee for this approach. Proposition 3 plays an important role in coming up with this performance guarantee.

5 Performance Guarantee

To obtain a solution to problem (1) with a performance guarantee, we propose the following algorithm, referred to as Approx.

Step 1. Solve the problem $\max_{(\boldsymbol{f},\boldsymbol{h})\in\widetilde{\mathcal{P}}} \left\{ \sum_{k\in M} \prod_{\ell=1}^k \frac{1}{1+h^\ell} \times f^k \right\}$ and denote an optimal solution to this problem by $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$.

Step 2. Since $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$, there exists a solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m) \in \mathcal{F}$ such that $\Phi^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ and $\Gamma^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \geq 0$ for all $k \in M$. Return one such solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$.

In this section, we give a performance guarantee for the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ provided by the Approx algorithm. In the next section, we give an approach that allows us to execute the Approx algorithm efficiently when the number of stages is fixed. Putting these two results together yields an FPTAS for problem (1). We proceed to giving a performance guarantee for the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ provided by the Approx algorithm. We let $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathcal{P}$ be an optimal solution to problem (3). By Lemma 2, the optimal objective value of problem (3) is equal to the optimal objective value of problem (1). Since $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathcal{P}$, by the alternative definition of $\mathcal{P}$ in (5), there exists $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ such that $F^k_{n+1}(\widehat{f}^k, \widehat{S}^k) \leq 0$ and $H^k_{n+1}(\widehat{h}^k, \widehat{S}^k) \geq 0$ for all $k \in M$. We define $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}})$ as $\overline{f}^k = \lfloor \widehat{f}^k \rfloor/(1+\rho)^{|\widehat{S}^k|}$ and $\overline{h}^k = (1+\rho)^{|\widehat{S}^k|}\, \lceil \widehat{h}^k \rceil$ for all $k \in M$.

In the next lemma, we show that $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}})$ is feasible to the problem in Step 1 above.

Lemma 4 *For any* $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \Re^m_+ \times \Re^m_+$ *and* $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ *that satisfies* $F^k_{n+1}(\widehat{f}^k, \widehat{S}^k) \leq 0$ *and* $H^k_{n+1}(\widehat{h}^k, \widehat{S}^k) \geq 0$ *for all* $k \in M$*, let* $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}})$ *be such that* $\overline{f}^k = \lfloor \widehat{f}^k \rfloor / (1+\rho)^{|\widehat{S}^k|}$ *and* $\overline{h}^k = (1+\rho)^{|\widehat{S}^k|} \lceil \widehat{h}^k \rceil$ *for all* $k \in M$*. Then, we have* $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$*.*

Proof. Adding the equality in (4) over all $i \in N$ and noting that $F^k_1(f^k, S^k) = f^k$, we obtain $F^k_{n+1}(f^k, S^k) = f^k - \sum_{i \in S^k} r_i \, v^k_i$, which implies that $F^k_{n+1}(f^k, S^k)$ is increasing in f^k. In this case, noting that $F^k_{n+1}(\widehat{f}^k, \widehat{S}^k) \leq 0$ and $\lfloor f^k \rfloor \leq f^k$, we obtain $F^k_{n+1}(\lfloor \widehat{f}^k \rfloor, \widehat{S}^k) \leq F^k_{n+1}(\widehat{f}^k, \widehat{S}^k) \leq 0$ as well. Therefore, we have $F^k_{n+1}(\lfloor \widehat{f}^k \rfloor, \widehat{S}^k) \leq 0$, in which case, using the fact that $\lfloor \widehat{f}^k \rfloor \in \mathsf{Dom}_+$, by Proposition 3, we obtain $\Phi^k_{n+1}(\overline{f}^k, \widehat{S}^k) = \Phi^k_{n+1}(\lfloor \widehat{f}^k \rfloor / (1+\rho)^{|\widehat{S}^k|}, \widehat{S}^k) \leq 0$. Using a similar reasoning, we also have $\Gamma^k_{n+1}(\overline{h}^k, \widehat{S}^k) = \Gamma^k_{n+1}((1+\rho)^{|\widehat{S}^k|} \lceil \widehat{h}^k \rceil, \widehat{S}^k) \geq 0$. In this case, there exists $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ such that $\Phi^k_{n+1}(\overline{f}^k, \widehat{S}^k) \leq 0$ and $\Gamma^k_{n+1}(\overline{h}^k, \widehat{S}^k) \geq 0$ for all $k \in M$, so that noting the definition of $\widetilde{\mathcal{P}}$ in (7), we have $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$. □

When $\rho > 0$ is close to zero, $(1+\rho)^{|S^k|}$ is close to one. Thus, by Lemma 4, we can scale any solution $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathcal{P}$ by a factor close to one to obtain a solution $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$, as long as $\rho > 0$ is small. In other words, given a solution $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathcal{P}$, which is optimal to problem (3), we can scale this solution by a factor close to one to obtain a solution $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$, which is feasible to the problem in Step 1 of the Approx algorithm. In the next theorem, we use this observation to give a performance guarantee for the solution provided by the Approx algorithm. In this theorem and throughout the rest of the paper, we use $\textsc{Rev}(S^1, \ldots, S^m)$ to denote the objective function of problem (1), which is the expected revenue from the solution $(S^1, \ldots, S^m)$.

Theorem 5 *Letting* $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ *be the output of the* Approx *algorithm and* $\widehat{Z}$ *be the optimal objective value of problem* (1)*, we have* $\textsc{Rev}(\widetilde{S}^1, \ldots, \widetilde{S}^m) \geq \widehat{Z}/(1+\rho)^{3n+1}$*.*

Proof. We let $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ be an optimal solution to the problem in Step 1 of the Approx algorithm. By (6), we have $\Phi^k_{i+1}(\widetilde{f}^k, \widetilde{S}^k) \geq \Phi^k_i(\widetilde{f}^k, \widetilde{S}^k) - r_i \, v^k_i \, \mathbf{1}(i \in \widetilde{S}^k)$ for all $i \in N$, $k \in M$, in which case, adding this inequality over all $i \in N$ and noting that $\Phi^k_1(\widetilde{f}^k, \widetilde{S}^k) = \widetilde{f}^k$, we obtain $\Phi^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \geq \widetilde{f}^k - \sum_{i \in N} r_i \, v^k_i \, \mathbf{1}(i \in \widetilde{S}^k) = \widetilde{f}^k - \sum_{i \in \widetilde{S}^k} r_i \, v^k_i$ for all $k \in M$. Furthermore, by the definition of $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ in Step 2 of the Approx algorithm, we also have $\Phi^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ for all $k \in M$. In this case, we obtain $\widetilde{f}^k - \sum_{i \in \widetilde{S}^k} r_i \, v^k_i \leq \Phi^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ so that $\sum_{i \in \widetilde{S}^k} r_i \, v^k_i \geq \widetilde{f}^k$ for all $k \in M$. Using a similar reasoning, we have $\sum_{i \in \widetilde{S}^k} v^k_i \leq \widetilde{h}^k$ for all $k \in M$ as well. Therefore, the expected revenue from the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ satisfies

$$\textsc{Rev}(\widetilde{S}^1, \ldots, \widetilde{S}^m) \;=\; \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1 + \sum_{i \in \widetilde{S}^\ell} v^\ell_i} \right\} \sum_{i \in \widetilde{S}^k} r_i \, v^k_i \;\geq\; \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1 + \widetilde{h}^\ell} \right\} \widetilde{f}^k, \tag{9}$$

which shows that the optimal objective value of the problem in Step 1 of the Approx algorithm is a lower bound on the expected revenue from the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$.

Next, we construct a lower bound on the optimal objective value of the problem in Step 1 of the Approx algorithm by giving a feasible solution to this problem. Using $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to denote an optimal solution to problem (1), we let $\widehat{f}^k = \sum_{i \in \widehat{S}^k} r_i\, v_i^k$ and $\widehat{h}^k = \sum_{i \in \widehat{S}^k} v_i^k$ for all $k \in M$. By the discussion that follows the definition of $F_i^k(f^k, S^k)$ in (4), we have $F_{n+1}^k(f^k, S^k) = f^k - \sum_{i \in S^k} r_i\, v_i^k$. Thus, we have $F_{n+1}^k(\widehat{f}^k, \widehat{S}^k) = \widehat{f}^k - \sum_{i \in \widehat{S}^k} r_i\, v_i^k = 0$ for all $k \in M$. Using a similar reasoning, we also have $H_{n+1}^k(\widehat{h}^k, \widehat{S}^k) = 0$ for all $k \in M$. In this case, letting $\overline{f}^k = \lfloor \widehat{f}^k \rfloor / (1+\rho)^{|\widehat{S}^k|}$ and $\overline{h}^k = (1+\rho)^{|\widehat{S}^k|} \lceil \widehat{h}^k \rceil$ for all $k \in M$, by Lemma 4, we obtain $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$, which implies that $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}})$ is a feasible solution to the problem in Step 1 of the Approx algorithm. Noting that $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ is an optimal solution to this problem, we obtain

$$
\begin{aligned}
\sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+\widetilde{h}^\ell} \right\} \widetilde{f}^k &\geq \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+\overline{h}^\ell} \right\} \overline{f}^k \\
&= \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+(1+\rho)^{|\widehat{S}^\ell|} \lceil \widehat{h}^\ell \rceil} \right\} \frac{\lfloor \widehat{f}^k \rfloor}{(1+\rho)^{|\widehat{S}^k|}} \\
&\geq \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+(1+\rho)^{|\widehat{S}^\ell| + \mathbf{1}(\widehat{h}^\ell > 0)}\, \widehat{h}^\ell} \right\} \frac{\widehat{f}^k}{(1+\rho)^{|\widehat{S}^k|+1}}, \qquad (10)
\end{aligned}
$$

where the last equality is by the fact that $\lceil x \rceil \leq (1+\rho)^{\mathbf{1}(x>0)}\, x$ and $\lfloor x \rfloor \geq x/(1+\rho)$ for any $x \in \Re_+$. Since $\widehat{h}^k = \sum_{i \in \widehat{S}^k} v_i^k$, we have $\widehat{h}^k > 0$ if and only if $\widehat{S}^k \neq \varnothing$. Therefore, we obtain

$$
\begin{aligned}
&\sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+(1+\rho)^{|\widehat{S}^\ell| + \mathbf{1}(\widehat{h}^\ell > 0)}\, \widehat{h}^\ell} \right\} \frac{\widehat{f}^k}{(1+\rho)^{|\widehat{S}^k|+1}} \\
&\qquad = \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+(1+\rho)^{|\widehat{S}^\ell| + \mathbf{1}(\widehat{S}^\ell \neq \varnothing)}\, \widehat{h}^\ell} \right\} \frac{\widehat{f}^k}{(1+\rho)^{|\widehat{S}^k|+1}} \\
&\qquad \geq \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{(1+\rho)^{|\widehat{S}^\ell| + \mathbf{1}(\widehat{S}^\ell \neq \varnothing)}\, (1+\widehat{h}^\ell)} \right\} \frac{\widehat{f}^k}{(1+\rho)^{|\widehat{S}^k|+1}}. \qquad (11)
\end{aligned}
$$

Since $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$, the sets $\widehat{S}^1, \ldots, \widehat{S}^m$ are disjoint. Therefore, we have $\sum_{\ell=1}^k |\widehat{S}^\ell| \leq n$ and $\sum_{\ell=1}^k \mathbf{1}(\widehat{S}^\ell \neq \varnothing) \leq n$ for all $k \in M$. Also, $|\widehat{S}^k| \leq n$. So, we have

$$
\begin{aligned}
&\sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{(1+\rho)^{|\widehat{S}^\ell| + \mathbf{1}(\widehat{S}^\ell \neq \varnothing)}\, (1+\widehat{h}^\ell)} \right\} \frac{\widehat{f}^k}{(1+\rho)^{|\widehat{S}^k|+1}} \\
&\qquad \geq \sum_{k \in M} \frac{1}{(1+\rho)^{2n}} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+\widehat{h}^\ell} \right\} \frac{\widehat{f}^k}{(1+\rho)^{n+1}} \\
&\qquad = \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+\sum_{i \in \widehat{S}^\ell} v_i^\ell} \right\} \frac{\sum_{i \in \widehat{S}^k} r_i\, v_i^k}{(1+\rho)^{3n+1}} = \frac{\widehat{Z}}{(1+\rho)^{3n+1}}, \qquad (12)
\end{aligned}
$$

where the first equality is by the definition of $(\widehat{f}^k, \widehat{h}^k)$ and the second equality holds as $(\widehat{S}^1, \ldots, \widehat{S}^m)$ is an optimal solution to problem (1). The desired result follows by (9), (10), (11) and (12). $\square$

Thus, the expected revenue from the solution provided by the Approx algorithm deviates from the optimal expected revenue by no more than a factor of $(1+\rho)^{3n+1}$. For any $\epsilon \in (0,1)$, consider executing the Approx algorithm with $\rho = \epsilon/(8n)$. Since $\epsilon < 1$, we have $(1+\rho)^{3n+1} \leq (1+\rho)^{4n} = (1+\frac{\epsilon}{8n})^{4n} \leq \exp(\epsilon/2) \leq 1+\epsilon$, so that $\text{Rev}(\widetilde{S}^1, \ldots, \widetilde{S}^m) \geq \widehat{Z}/(1+\rho)^{3n+1} \geq \widehat{Z}/(1+\epsilon) \geq (1-\epsilon)\,\widehat{Z}$. Thus, the expected revenue from the solution provided by the Approx algorithm is at least $1-\epsilon$ fraction of the optimal expected revenue. Although the solution provided by the Approx algorithm has a performance guarantee, it is not yet clear that we can execute the Approx algorithm efficiently. In the next section, we give a dynamic program that allows us to execute the Approx algorithm efficiently when the number of stages is fixed. By accounting for the number of operations to solve the dynamic program, we ultimately obtain our FPTAS.

6 Dynamic Programming Formulation

To execute the Approx algorithm efficiently, we make use of two observations. First, we can use a dynamic program to check whether a given value of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$, allowing us to check the feasibility of a solution to the problem in Step 1 of the Approx algorithm. Second, we can bound the components of an optimal solution to the problem in Step 1 of the Approx algorithm. Using the bound, the number of values of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ that can possibly be an optimal solution becomes polynomial in input size, when the number of stages is fixed. In this case, we can execute Step 1 of the Approx algorithm by checking whether each value of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ that can be possibly be an optimal solution to the problem in this step satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$ and by picking one that provides the best expected revenue. In Step 2 of the Approx algorithm, we need to find a solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m) \in \mathcal{F}$ such that $\Phi_{n+1}^k(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ and $\Gamma_{n+1}^k(\widetilde{f}^k, \widetilde{S}^k) \geq 0$ for all $k \in M$. Noting (7), checking whether a given value of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$ requires finding a solution $(S^1, \ldots, S^m) \in \mathcal{F}$ such that $\Phi_{n+1}^k(f^k, S^k) \leq 0$ and $\Gamma_{n+1}^k(f^k, S^k) \geq 0$ for all $k \in M$. Therefore, we can use the dynamic program that we use in Step 1 of the Approx algorithm to execute Step 2 as well.

We proceed to giving a dynamic program that allows us to check whether a given value of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ is feasible to the problem in Step 1 of the Approx algorithm. Consider a fixed value of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$. Noting (6), the values of $\Phi_i^k(f^k, S^k)$ and $\Gamma_i^k(h^k, S^k)$ depend on the decisions that we make for the products $\{1, \ldots, i-1\}$, but not on the decisions that we make for the products $\{i, \ldots, n\}$. In our dynamic program, the decision epochs correspond to the products. At the decision epoch corresponding to product i, we choose the stage at which we should offer product i. Note that we may decide not to offer product i at all. In particular, to capture the decisions that we make at this decision epoch, we use the vector $\boldsymbol{x}_i = (\boldsymbol{x}_i^1, \ldots, \boldsymbol{x}_i^m) \in \{0,1\}^m$, where $x_i^k = 1$ if and only if we offer product i in stage k. Since we can offer a product in no more than one stage, the decision should satisfy $\sum_{k \in M} x_i^k \leq 1$. At the decision epoch corresponding to product i, we have already made the decisions for the products $\{1, \ldots, i-1\}$. Therefore, the state variable at the decision epoch corresponding to product i are the values of $\Phi_i^k(f^k, S^k)$ and $\Gamma_i^k(h^k, S^k)$ for all $k \in M$, which are determined by the decisions that we make for the products in $\{1, \ldots, i-1\}$. Given that $\Phi_i^k(f^k, S^k) = f_i^k$ and $\Gamma_i^k(h^k, S^k) = h_i^k$ for all $k \in M$, by (6), after we make the decision for product i, we can compute $\Phi^k_{i+1}(f^k, S^k)$ and $\Gamma^k_{i+1}(h^k, S^k)$ as $\lceil f^k_i - r_i\, v^k_i\, x^k_i \rceil$ and $\lfloor f^k_i - v^k_i\, x^k_i \rfloor$ for all $k \in M$. To capture the state at the decision epoch corresponding to product i, we define the vectors $\boldsymbol{f}_i = (f^1_i, \ldots, f^m_i)$ and $\boldsymbol{h}_i = (h^1_i, \ldots, h^m_i)$. Thus, letting $\boldsymbol{e}^k \in \{0,1\}^m$ be a unit vector with a one in the k-th component, to check whether a fixed value of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}^m_+ \times \mathsf{Dom}^m_+$ satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$, we can solve the dynamic program

$$V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) \;=\; \max_{\substack{\boldsymbol{x}_i \in \{0,1\}^m : \\ \sum_{k \in M} x^k_i \leq 1}} \left\{ V_{i+1}\left(\left\lceil \boldsymbol{f}_i - \sum_{k \in M} \boldsymbol{e}^k\, r_i\, v^k_i\, x^k_i \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k\, v^k_i\, x^k_i \right\rfloor \right) \right\}, \tag{13}$$

with the boundary condition that $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = 0$ if $f^k_{n+1} \leq 0$ and $h^k_{n+1} \geq 0$ for all $k \in M$. Otherwise, we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = -\infty$. Once we compute the value functions $\{V_i(\cdot,\cdot) : i \in N\}$ through the dynamic program above, for a given value of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}^m_+ \times \mathsf{Dom}^m_+$, we have $V_1(\boldsymbol{f}, \boldsymbol{h}) = 0$ if and only if $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$. Note that we apply the operators $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ on the vectors in (13) componentwise.

The state variable $(\boldsymbol{f}_i, \boldsymbol{h}_i)$ in the dynamic program in (13) takes values in the set $\mathsf{Dom}^m_+ \times \mathsf{Dom}^m_+$. Therefore, the number of possible values for the state variable is countable but not yet finite. Next, we give a natural bound on the state variable in this dynamic program, in which case, the number of possible values for the state variable becomes finite. Thus, we can solve the dynamic program in (13) in finite number of operations. In the next lemma, along with the discussion that follows this lemma, we show that we do not need to consider the values of the state variable whose components exceed a certain upper bound. In this lemma and throughout the rest of the paper, for notational brevity, we let $R_{\max} = \max\{r_i\, v^k_i : i \in N,\ k \in M\}$, $R_{\min} = \min\{r_i\, v^k_i : i \in N,\ k \in M\}$, $V_{\max} = \max\{v^k_i : i \in N,\ k \in M\}$ and $V_{\min} = \min\{v^k_i : i \in N,\ k \in M\}$. Also, we define the function $\Delta(\rho, n) = ((1+\rho)^n - 1)/\rho$. Note that $\Delta(\rho, n) \geq (1 + \rho\, n - 1)/\rho = n$.

Lemma 6 *For any $\{\boldsymbol{x}_i : i \in N\}$ and $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \Re^m_+ \times \Re^m_+$, assume that $\{(\boldsymbol{f}_i, \boldsymbol{h}_i) : i \in N\}$ are given by $f^k_{i+1} = \lceil f^k_i - r_i\, v^k_i\, x^k_i \rceil$ and $h^k_{i+1} = \lfloor h^k_i - v^k_i\, x^k_i \rfloor$ for all $i \in N$, $k \in M$ with $f^k_1 = \widehat{f}^k$ and $h^k_1 = \widehat{h}^k$. If $\widehat{f}^k > \lceil n R_{\max} \rceil$, then we have $f^k_{n+1} > 0$. Similarly, if $\widehat{h}^k \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$, then we have $h^k_{n+1} \geq 0$.*

The proof of the lemma above follows from an induction over the products and we defer it to Appendix A. The value function $V_{n+1}(\cdot,\cdot)$ in (13) takes the value 0 or $-\infty$, depending only on the signs of the components of the state variable. Furthermore, given the state variable $(\boldsymbol{f}_i, \boldsymbol{h}_i)$ in the decision epoch corresponding to product i, each component of the state variable at the next decision epoch is computed by using the recursion $f^k_{i+1} = \lceil f^k_i - r_i\, v^k_i\, x^k_i \rceil$ and $h^k_{i+1} = \lfloor h^k_i - v^k_i\, x^k_i \rfloor$ for all $k \in M$. Therefore, if we start with the initial state $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$, then each component of the state variable is computed by using the recursion in Lemma 6. In this case, by Lemma 6, if $\widehat{f}^k > \lceil n R_{\max} \rceil$ for some $k \in M$ in the initial state variable $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$, then the same component of the state variable $(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1})$ at the final decision epoch always satisfies $f^k_{n+1} > 0$, irrespective of the decisions that we take in the intermediate decision epochs. Thus, noting the boundary condition in the dynamic program in (13), we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = -\infty$ irrespective of our decisions, which implies that $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) = -\infty$. In other words, we can immediately deduce that $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) = -\infty$, whenever $\widehat{f}^k > \lceil n R_{\max} \rceil$ for some $k \in M$. We do not need to compute $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ explicitly whenever $\widehat{f}^k > \lceil n R_{\max} \rceil$ for some $k \in M$. On the other hand, by Lemma 6, if $\widehat{h}^k \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$ for some $k \in M$ in the initial state variable $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$, then the same component of the state variable $(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1})$ at the final decision epoch always satisfies $h^k_{n+1} \geq 0$, again, irrespective of the decisions that we take in the intermediate decision epochs. Thus, since the value function $V_{n+1}(\cdot, \cdot)$ only depends on the signs of the components of the state variable, as long as $\widehat{h}^k \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$ in the initial state variable, the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ does not depend on the specific value of $\widehat{h}^k$. In other words, if we have $\widehat{h}^k > \lceil \Delta(\rho, n)\, V_{\max} \rceil$ for some $k \in M$ in the initial state variable $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$, then we can bump the value of this component of the state variable down to $\lceil \Delta(\rho, n)\, V_{\max} \rceil$ without changing the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$. Therefore, we do not need to compute $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ explicitly either whenever $\widehat{h}^k > \lceil \Delta(\rho, n)\, V_{\max} \rceil$ for some $k \in M$. In this case, we do not need to compute the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ when $\widehat{f}^k > \lceil n R_{\max} \rceil$ or $h^k > \lceil \Delta(\rho, n)\, V_{\max} \rceil$ for some $k \in M$.

Also, since $r_i\, v^k_i \geq R_{\min}$, if $0 < \widehat{f}^k < \lfloor R_{\min} \rfloor$ for some $k \in M$ in the initial state variable $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$, then offering any of the products at any decision epoch sets the value of this component of the state variable to zero at the subsequent decision epochs. However, if $\widehat{f}^k = \lfloor R_{\min} \rfloor$ for some $k \in M$, then offering any of the products at any decision epoch also sets the value of this component of the state variable to zero at the subsequent decision epochs. Noting that the value function $V_{n+1}(\cdot, \cdot)$ only depends on the signs of the components of the state variable, if $0 < \widehat{f}^k < \lfloor R_{\min} \rfloor$ for some $k \in M$, then we can bump the value of this component of the state variable up to $\lfloor R_{\min} \rfloor$ without changing the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$. In other words, we do not need to compute $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ explicitly whenever $0 < \widehat{f}^k < \lfloor R_{\min} \rfloor$ for some $k \in M$. Using a similar argument, we do not need to compute $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ explicitly whenever $0 < \widehat{h}^k < \lfloor V_{\min} \rfloor$ for some $k \in M$. So, we do not need to compute the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ when $0 < \widehat{f}^k < \lfloor R_{\min} \rfloor$ or $0 < \widehat{h}^k < \lfloor V_{\min} \rfloor$ for some $k \in M$.

Putting the discussion in the previous two paragraphs together, we only need to compute the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ for the values of the initial state variable $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathsf{Dom}^m_+ \times \mathsf{Dom}^m_+$ that satisfies $\widehat{f}^k \in \{0\} \cup [\lfloor R_{\min} \rfloor, \lceil n R_{\max} \rceil]$ and $\widehat{h}^k \in \{0\} \cup [\lfloor V_{\min} \rfloor, \lceil \Delta(\rho, n)\, V_{\max} \rceil]$ for all $k \in M$. Once we compute the value function at these values of the state variable, we can immediately deduce the value function at other values of the state variable. This discussion also indicates that there exists an optimal solution $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ to the problem in Step 1 of the Approx algorithm that satisfies $\widetilde{f}^k \in \{0\} \cup [\lfloor R_{\min} \rfloor, \lceil n R_{\max} \rceil]$ and $\widetilde{h}^k \in \{0\} \cup [\lfloor V_{\min} \rfloor, \lceil \Delta(\rho, n)\, V_{\max} \rceil]$ for all $k \in M$. In particular, if $\widetilde{f}^k > \lceil n R_{\max} \rceil$ for some $k \in M$, then $V_1(\widetilde{f}, \widetilde{h}) = -\infty$ by our earlier discussion, so $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) \notin \widetilde{\mathcal{P}}$, which indicates that $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ is not feasible to the problem in Step 1. On the other hand, an optimal solution $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ to the problem in Step 1 satisfies $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$ so that $V_1(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) = 0$. If we have $\widetilde{h}^k \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$ for some $k \in M$, then we, by our earlier discussion, can bump the value of $\widetilde{h}^k$ down to $\lceil \Delta(\rho, n)\, V_{\max} \rceil$ without changing the value of $V_1(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ from zero. Therefore, the solution that we obtain in this way is still feasible to the problem in Step 1. Furthermore, since the objective function of this problem is decreasing in h^k, the solution that we obtain in this way is also an optimal solution. Similarly if we have $0 < \widetilde{f}^k < \lfloor R_{\min} \rfloor$ for some $k \in M$, then we can bump the value of $\widetilde{f}^k$ up to $\lfloor R_{\min} \rfloor$ without changing the value of $V_1(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ from zero. Therefore, the solution that we obtain in this way is still feasible to the problem in Step 1. Furthermore, the objective function of this problem is increasing in f^k, indicating that the solution that we obtain in this way is also an optimal solution. Lastly, using a similar argument, if $0 < \widetilde{h}^k < \lfloor V_{\min} \rfloor$ for some $k \in M$, then we can bump the value of $\widetilde{h}^k$ down to zero and still obtain an optimal solution to the problem in Step 1. So, there exists a finite number of possible solutions to the problem in Step 1 of the Approx algorithm and we can solve the dynamic program in (13) to check whether each one of these solutions is feasible. In the next section, we use this observation to give our FPTAS.

7 Fully Polynomial-Time Approximation Scheme

At the end of Section 5, we discuss that if we execute the Approx algorithm with $\rho = \epsilon/(8n)$ for some $\epsilon \in (0, 1)$, then we obtain a solution to problem (1) that provides an expected revenue deviating from the optimal expected revenue by no more than a factor of $1 - \epsilon$. Next, we discuss that if we execute the Approx algorithm with $\rho = \epsilon/(8n)$ for some $\epsilon \in (0, 1)$, then the running time is polynomial in input size, when the number of stages is fixed. In this way, we obtain our FPTAS. We know that there exists an optimal solution $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ to the problem in Step 1 of the Approx algorithm that satisfies $\widetilde{f}^k \in \{0\} \cup [\lfloor R_{\min} \rfloor, \lceil n R_{\max} \rceil]$ and $\widetilde{h}^k \in \{0\} \cup [\lfloor V_{\min} \rfloor, \lceil \Delta(\rho, n)\, V_{\max} \rceil]$ for all $k \in M$. Noting the definition of Dom, the number of possible values of $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ that lie in these intervals is given by

$$O\left(\left(\frac{\log(n \frac{R_{\max}}{R_{\min}})}{\log(1+\rho)} \right)^m \times \left(\frac{\log(\Delta(\rho, n) \frac{V_{\max}}{V_{\min}})}{\log(1+\rho)} \right)^m \right) = O\left(\left(\frac{\log(n \frac{R_{\max}}{R_{\min}}) \times \log(\Delta(\rho, n) \frac{V_{\max}}{V_{\min}})}{\rho^2} \right)^m \right). \quad (14)$$

To check whether a value of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ is feasible to the problem in Step 1 of the Approx algorithm, we can use the value function $V_1(\boldsymbol{f}, \boldsymbol{h})$. We know that we only need to compute the value function $V_1(\boldsymbol{f}, \boldsymbol{h})$ for the values of the initial state variable $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ that satisfies $f^k \in \{0\} \cup [\lfloor R_{\min} \rfloor, \lceil n R_{\max} \rceil]$ and $h^k \in \{0\} \cup [\lfloor V_{\min} \rfloor, \lceil \Delta(\rho, n)\, V_{\max} \rceil]$ for all $k \in M$. Each component of the state variable in the dynamic program in (13) decreases as we move from one decision epoch to the next. Therefore, if a component of a state variable turns negative, then it never turns positive in a subsequent decision epoch. Since $V_{n+1}(\cdot, \cdot)$ only depends on the signs of the components of the state variable, the number of possible values for the state variable at each decision epoch is also given by the expression in (14).

There are $O(n)$ decision epochs in the dynamic program in (13). We can compute the value functions $\{V_i(\cdot, \cdot) : i \in N\}$ starting from the last decision epoch and moving backwards over the decision epochs. Computation of the value function at a particular state takes $O(m)$ operations, since there are m stages in which we can offer a product. Thus, noting the number of possible values for the state variable in (14), we can execute Step 1 of the Approx algorithm in $O(mn(\log(n \frac{R_{\max}}{R_{\min}}))^m\, (\log(\Delta(\rho, n) \frac{V_{\max}}{V_{\min}}))^m / \rho^{2m})$ operations. On the other hand, to execute Step 2 of the Approx algorithm, once we obtain an optimal solution $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ to the problem in Step 1, we can follow the optimal state and action trajectory in the dynamic program in (13). In particular, letting $(\widetilde{\boldsymbol{f}}_1, \widetilde{\boldsymbol{h}}_1) = (\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$, we can compute $\widetilde{\boldsymbol{x}}_i$ and $(\widetilde{\boldsymbol{f}}_i, \widetilde{\boldsymbol{h}}_i)$ recursively as $\widetilde{\boldsymbol{x}}_i = \arg\max\{V_{i+1}(\lceil \widetilde{\boldsymbol{f}}_i - \sum_{k\in M} \boldsymbol{e}^k\, r_i\, v_i^k\, x_i^k \rceil, \lfloor \widetilde{\boldsymbol{h}}_i - \sum_{k\in M} \boldsymbol{e}^k\, v_i^k\, x_i^k \rfloor) : \sum_{k\in M} x_i^k \leq 1,\ \boldsymbol{x}_i \in \{0,1\}^m\}$ with $\widetilde{\boldsymbol{f}}_{i+1} = \lceil \widetilde{\boldsymbol{f}}_i - \sum_{k\in M} \boldsymbol{e}^k\, r_i\, v_i^k\, x_i^k \rceil$ and $\widetilde{\boldsymbol{h}}_{i+1} = \lfloor \widetilde{\boldsymbol{h}}_i - \sum_{k\in M} \boldsymbol{e}^k\, v_i^k\, x_i^k \rfloor$ for all $i \in N$. In this case, letting $\widetilde{S}^k = \{i \in N : \widetilde{x}_i^k = 1\}$ for all $k \in M$, the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m) \in \mathcal{F}$ satisfies $\Phi_{n+1}^k(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ and $\Gamma_{n+1}^k(\widetilde{f}^k, \widetilde{S}^k) \geq 0$ for all $k \in M$. The number of operations required to execute Step 2 of the Approx algorithm is dominated by that required to execute Step 1. In the next theorem, we build on this discussion to give an FPTAS for problem (1).

Theorem 7 *For any $\epsilon \in (0,1)$, we can find a solution to problem (1) such that the expected revenue from this solution deviates from the optimal objective value of problem (1) by at most a factor of $1-\epsilon$ and the number of operations required to obtain this solution is $O(mn^{2m+1} (\log(n \frac{R_{\max}}{R_{\min}}))^m\, (\log(n \frac{V_{\max}}{V_{\min}}))^m / \epsilon^{2m})$.*

Proof. Consider executing the Approx algorithm with $\rho = \epsilon/(8n)$. By the discussion at the end Section 5, this algorithm returns a solution such that the expected revenue from this solution deviates from the optimal objective value of problem (1) by at most a factor of $1-\epsilon$. On the other hand, by the discussion right before the theorem, we can execute the Approx algorithm in $O(mn (\log(n \frac{R_{\max}}{R_{\min}}))^m\, (\log(\Delta(\rho, n) \frac{V_{\max}}{V_{\min}}))^m / \rho^{2m})$ operations. Noting that $\exp(x/2) \leq 1 + x$ for all $x \in (0,1)$ and $2\rho\, n = \epsilon/4 < 1$, we have $\Delta(\rho, n) = ((1+\rho)^n - 1)/\rho \leq (\exp(\rho\, n) - 1)/\rho = (1 + 2\rho\, n - 1)/\rho = 2n$. In this case, replacing ρ with $\epsilon/(8n)$ and $\Delta(\rho, n)$ with $2n$, we obtain the number of operations to execute the Approx algorithm with $\rho = \epsilon/(8n)$. □

The number of operations in Theorem 7 is polynomial in the input size and $1/\epsilon$ when the number of stages m is fixed, yielding an FPTAS for problem (1) for fixed number of stages. The state variable in the dynamic program in (13) takes values in $\mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$. It turns out we can formulate an equivalent dynamic program, where the state variable takes values in $\mathsf{Dom}_+^{m-1} \times \mathsf{Dom}_+^m$. Using the latter dynamic program improves the number of operations in our FPTAS. To formulate the equivalent dynamic program, we choose one stage arbitrarily. We choose the first stage in the discussion that follows. We partition the vector $\boldsymbol{f} = (f^1, \ldots, f^m)$ into the scalar f^1 and the vector $\boldsymbol{f}^{-1} = (f^2, \ldots, f^m)$. Therefore, we can write $\boldsymbol{f} = (f^1, \boldsymbol{f}^{-1})$. It is not difficult to use induction over the products to show that $\Phi_{n+1}^k(f^k, S^k)$ is increasing in f^k. In particular, since $\Phi_1^k(f^k, S^k) = f^k$, $\Phi_1^k(f^k, S^k)$ is increasing f^k. If we assume that $\Phi_i^k(f^k, S^k)$ is increasing in f^k and note that $\lceil x \rceil$ in increasing in x, then (6) implies that $\Phi_{i+1}^k(f^k, S^k)$ is increasing in f^k as well, completing the induction argument. Therefore, if $\Phi_{i+1}^k(\widehat{f}^k, S^k) \leq 0$, then we have $\Phi_{i+1}^k(f^k, S^k) \leq 0$ for all $f^k \leq \widehat{f}^k$. Similarly, if $\Phi_{i+1}^k(\widehat{f}^k, S^k) > 0$, then we have $\Phi_{i+1}^k(f^k, S^k) > 0$ for all $f^k > \widehat{f}^k$.

In this case, consider a fixed value of $(\boldsymbol{f}^{-1}, \boldsymbol{h}) \in \mathsf{Dom}_+^{m-1} \times \mathsf{Dom}_+^m$. Noting the definition of $\widetilde{\mathcal{P}}$ in (7), depending on the value of $(\boldsymbol{f}^{-1}, \boldsymbol{h}) \in \mathsf{Dom}_+^{m-1} \times \mathsf{Dom}_+^m$, there exists a threshold $T(\boldsymbol{f}^{-1}, \boldsymbol{h})$ such that we have $((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) \in \widetilde{\mathcal{P}}$ for all $f^1 \in \mathsf{Dom}_+$ that satisfies $f^1 \leq T(\boldsymbol{f}^{-1}, \boldsymbol{h})$, whereas we have $((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) \notin \widetilde{\mathcal{P}}$ for all $f^1 \in \mathsf{Dom}_+$ that satisfies $f^1 > T(\boldsymbol{f}^{-1}, \boldsymbol{h})$. In other words, we have $V_1((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) = 0$ if $f^1 \le T(\boldsymbol{f}^{-1}, \boldsymbol{h})$, whereas we have $V_1((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) = -\infty$ if $f^1 > T(\boldsymbol{f}^{-1}, \boldsymbol{h})$. Thus, if we can compute the threshold $T(\boldsymbol{f}^{-1}, \boldsymbol{h})$ for all $(\boldsymbol{f}^{-1}, \boldsymbol{h}) \in \mathsf{Dom}_+^{m-1} \times \mathsf{Dom}_+^m$, then we do not need to solve the dynamic program in (13). In the rest of this section, we give a dynamic program to compute the threshold $T(\boldsymbol{f}^{-1}, \boldsymbol{h})$. In particular, letting $M^{-1} = M \setminus \{1\}$ for notational brevity, we consider the dynamic program

$$J_i(\boldsymbol{f}_i^{-1}, \boldsymbol{h}_i) = \max_{\substack{\boldsymbol{x}_i \in \{0,1\}^m : \\ \sum_{k \in M} x_i^k \le 1}} \left\{ r_i\, v_i^1\, x_i^1 + \left\lfloor J_{i+1}\left(\left\lceil \boldsymbol{f}_i^{-1} - \sum_{k \in M^{-1}} \boldsymbol{e}^k\, r_i\, v_i^k\, x_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k\, v_i^k\, x_i^k \right\rfloor \right) \right\rfloor \right\}, \quad (15)$$

with the boundary condition that $J_{n+1}(\boldsymbol{f}_{n+1}^{-1}, \boldsymbol{h}_{n+1}) = 0$ if $f_{n+1}^k \le 0$ for all $k \in M^{-1}$ and $h_{n+1}^k \ge 0$ for all $k \in M$. Otherwise, we have $J_{n+1}(\boldsymbol{f}_{n+1}^{-1}, \boldsymbol{h}_{n+1}) = -\infty$. In Appendix B, we show that we have $V_1((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) = 0$ if $f^1 \le \lfloor J_1(\boldsymbol{f}^{-1}, \boldsymbol{h}) \rfloor$, whereas we have $V_1((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) = -\infty$ if $f^1 > \lfloor J_1(\boldsymbol{f}^{-1}, \boldsymbol{h}) \rfloor$. Therefore, we can use $\lfloor J_1(\boldsymbol{f}^{-1}, \boldsymbol{h}) \rfloor$ as the threshold $T(\boldsymbol{f}^{-1}, \boldsymbol{h})$, preventing the need to compute $V_1(\boldsymbol{f}, \boldsymbol{h})$ for all $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$.

Working with the dynamic program in (15), we can use precisely the same argument earlier in this section to give an FPTAS for problem (1). This FPTAS provides a solution such that the expected revenue from this solution deviates from the optimal objective value of problem (1) by at most a factor of $1 - \epsilon$ and the number of operations required to obtain this solution is $O(mn^{2m} (\log(n \frac{R_{\max}}{R_{\min}}))^{m-1} (\log(n \frac{V_{\max}}{V_{\min}}))^m / \epsilon^{2m-1})$. Note that although working with the dynamic program in (15) allows us to obtain a more efficient FPTAS, the dynamic program in (13) is substantially more interpretable than the one in (15). Therefore, we chose to use the dynamic program in (13) in our initial presentation of our FPTAS.

Prior to our work, Desir and Goyal (2014) consider assortment optimization problems under a mixture of multinomial logit models. The authors develop an FPTAS by using the connections of their problem to the knapsack problem and aligning the cumulative capacity consumptions to a geometric grid. Due to the multiple stages in the choice process in our assortment optimization problem, we need to be careful to characterize the error resulting from aligning the cumulative capacity consumptions to a geometric grid, as in Proposition 3. By Theorem 5, for a fixed grid size ρ, the error is exponential in the number of products. As in Theorem 7, choosing the grid size ρ such that it differs from the precision ϵ by a factor of n, we get our FPTAS.

Lastly, although our assortment optimization problem is NP-hard, we can provide some structure for the form of the optimal solution. In particular, we can show that the union of the optimal sets to offer in the different stages is nested by revenue, including a certain number of products with the largest revenues. In other words, there exists an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (1) such that $\cup_{k \in M} \widehat{S}^k = \{i \in N : r_i \ge \widehat{\zeta}\}$ for some constant $\widehat{\zeta}$. While this result intuitively suggests that we should give more priority to offering the products with larger revenues, it does not allow us to find the optimal solution efficiently, since this result does not characterize the stage in which each product should be offered. We discuss this result in Appendix C.

8 Constraints on the Offered Sets of Products

In this section, we consider two types of constraints on the sets of products that we can offer to the customers. First, in online retail, for example, we may display the search results to a customer sequentially through multiple webpages. If there is limited space on the webpage, then it may be desirable to limit the number of products offered in each stage. We refer to this type of constraints as cardinality constraints *within* stages. Note that not all online retail applications require cardinality constraints within stages. In particular, it may be at our discretion to decide how many products to display on each page, in which case, it may not be necessary to limit the number of products that we offer in each stage. Second, in scheduling healthcare appointments over the phone, for example, to gently guide the patient through the choice process, we may offer sets of appointment slots sequentially. To avoid overwhelming the patient with a large number of options, it may be desirable to limit the total number of appointment slots offered over all stages. We refer to this type of constraints as cardinality constraints *across* stages.

Cardinality Constraints within Stages. We let C^k be the maximum number of products that we can offer in stage k. Therefore, the feasible sets of products that we can offer over all stages are $\mathcal{F} = \{(S^1, \ldots, S^m) : S^k \subseteq N \;\forall\, k \in M,\; |S^k| \le C^k \;\forall\, k \in M,\; S^k \cap S^\ell = \varnothing \;\forall\, k \ne \ell\}$. The development in Sections 3, 4 and 5 does not change at all, as long as we use this definition of $\mathcal{F}$ under cardinality constraints within stages. All we need to do is to interpret all occurrences of $\mathcal{F}$ as the one under cardinality constraints within stages. We slightly modify the dynamic program in (13) that we use to check whether a fixed value of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$. In particular, we let c_i^k be the number of products among $\{1, \ldots, i-1\}$ that we offer in stage k. Defining the vector $\boldsymbol{c}_i = (c_i^1, \ldots, c_i^m)$, we use the dynamic program

$$V_i(\boldsymbol{f}_i, \boldsymbol{h}_i, \boldsymbol{c}_i) \;=\; \max_{\substack{\boldsymbol{x}_i \,\in\, \{0,1\}^m \,:\\ \sum_{k \in M} x_i^k \,\le\, 1}} \left\{ V_{i+1}\left(\left\lceil \boldsymbol{f}_i - \sum_{k \in M} \boldsymbol{e}^k\, r_i\, v_i^k\, x_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k\, v_i^k\, x_i^k \right\rfloor, \boldsymbol{c}_i + \sum_{k \in M} \boldsymbol{e}^k\, x_i^k \right) \right\},$$

with the boundary condition that $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}, \boldsymbol{c}_{n+1}) = 0$ if $f_{n+1}^k \le 0$, $h_{n+1}^k \ge 0$ and $c_{n+1}^k \le C^k$ for all $k \in M$. Otherwise, we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}, \boldsymbol{c}_{n+1}) = -\infty$. In this case, we have $V_1(\boldsymbol{f}, \boldsymbol{h}, \boldsymbol{0}) = 0$ if and only if $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$, where $\boldsymbol{0} \in \mathbb{Z}_+^m$ is a vector of all zeros. We can slightly modify the discussion in Section 7 to construct our FPTAS. In particular, using the dynamic program above, under cardinality constraints within stages, we can execute the Approx algorithm in $O(mn^{m+1} (\log(n \frac{R_{\max}}{R_{\min}}))^m\, (\log(\Delta(\rho, n) \frac{V_{\max}}{V_{\min}}))^m / \rho^{2m})$ operations. Choosing $\rho = \epsilon/(8n)$, we obtain an FPTAS with a running time of $O(mn^{3m+1} (\log(n \frac{R_{\max}}{R_{\min}}))^m\, (\log(n \frac{V_{\max}}{V_{\min}}))^m / \epsilon^{2m})$ to obtain a $(1-\epsilon)$-approximate solution. Lastly, using the same approach in the dynamic program in (15), we can reduce this running time to $O(mn^{3m} (\log(n \frac{R_{\max}}{R_{\min}}))^{m-1}\, (\log(n \frac{V_{\max}}{V_{\min}}))^m / \epsilon^{2m-1})$.

Cardinality Constraints across Stages. We let C be the maximum total number of products that we can offer over all stages. Therefore, the feasible sets of products that we can offer over all stages are $\mathcal{F} = \{(S^1, \ldots, S^m) : S^k \subseteq N \;\forall\, k \in M,\; |\cup_{k \in M} S^k| \le C,\; S^k \cap S^\ell = \varnothing \;\forall\, k \ne \ell\}$. Once again, as long as we use the definition of $\mathcal{F}$ under cardinality constraints across stages, the development in Sections 3, 4 and 5 does not change at all. We slightly modify the dynamic program in (13) that we use to check whether a fixed value of $(\boldsymbol{f}, \boldsymbol{h}) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$. We use c_i to denote the total number of products among $\{1, \ldots, i-1\}$ that we offer in any of the stages. In this case, we use the dynamic program

$$V_i(\boldsymbol{f}_i, \boldsymbol{h}_i, c_i) \;=\; \max_{\substack{\boldsymbol{x}_i \,\in\, \{0,1\}^m \,: \\ \sum_{k \in M} x_i^k \,\leq\, 1}} \left\{ V_{i+1}\left(\left\lceil \boldsymbol{f}_i - \sum_{k \in M} \boldsymbol{e}^k \, r_i \, v_i^k \, x_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k \, v_i^k \, x_i^k \right\rfloor, c_i + \sum_{k \in M} x_i^k \right) \right\},$$

with the boundary condition that $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}, c_{n+1}) = 0$ if $f_{n+1}^k \leq 0$ and $h_{n+1}^k \geq 0$ for all $k \in M$ and $c_{n+1} \leq C$. Otherwise, we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}, c_{n+1}) = -\infty$. Thus, we have $V_1(\boldsymbol{f}, \boldsymbol{h}, 0) = 0$ if and only if $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$. We can construct our FPTAS by slightly modifying the discussion in Section 7. We can use the dynamic program above to execute the Approx algorithm in $O(mn^2 (\log(n \frac{R_{\max}}{R_{\min}}))^m \, (\log(\Delta(\rho, n) \frac{V_{\max}}{V_{\min}}))^m / \rho^{2m})$ operations under cardinality constraints across stages. To obtain an FPTAS, we choose $\rho = \epsilon/(8n)$, in which case, to obtain a $(1-\epsilon)$-approximate solution, our FPTAS has a running time of $O(mn^{2m+2} (\log(n \frac{R_{\max}}{R_{\min}}))^m \, (\log(n \frac{V_{\max}}{V_{\min}}))^m / \epsilon^{2m})$. We can reduce this running time to $O(mn^{2m+1} (\log(n \frac{R_{\max}}{R_{\min}}))^{m-1} \, (\log(n \frac{V_{\max}}{V_{\min}}))^m / \epsilon^{2m-1})$ by using the same approach in the dynamic program in (15). It is also not too difficult to combine the discussion in this paragraph with the one in the previous paragraph to limit the number of products offered in each stage, as well as the total number of products offered over all stages, in which case, we have joint cardinality constraints within and across stages.

Space Constraints across Stages. Naturally, we can consider the case where each product occupies a certain amount of space and we limit the total space consumption of the products offered in each stage or over all stages. We refer to these types of constraints as space constraints within or across stages. We can extend the discussion in this section to space constraints across stages, but the extension to space constraints within stages appears to be difficult. In particular, under space constraints across stages, we let w_i be the space consumption of product i and T be the limit on the total space consumption of the products offered over all stages. The development in Sections 3, 4 and 5 still does not change at all. Under space constraints across stages, in the dynamic program in (13), the value function $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i)$ would correspond to the minimum total space consumption for the products in $\{i, \ldots, n\}$ to ensure that $\Phi_{n+1}^k(f^k, S^k) \leq 0$ and $\Gamma_{n+1}(h^k, S^k) \geq 0$ for all $k \in M$, given that the decisions that we make for the products in $\{1, \ldots, i-1\}$ satisfy $f_i^k = \Phi_i^k(f^k, S^k)$ and $h_i^k = \Gamma_i^k(h^k, S^k)$ for all $k \in M$; see Desir and Goyal (2014). Therefore, we have

$$V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) \;=\; \min_{\substack{\boldsymbol{x}_i \,\in\, \{0,1\}^m \,: \\ \sum_{k \in M} x_i^k \,\leq\, 1}} \left\{ w_i \sum_{k \in M} x_i^k + V_{i+1}\left(\left\lceil \boldsymbol{f}_i - \sum_{k \in M} \boldsymbol{e}^k \, r_i \, v_i^k \, x_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k \, v_i^k \, x_i^k \right\rfloor \right) \right\},$$

with the boundary condition that $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = 0$ if $f_{n+1}^k \leq 0$ and $h_{n+1}^k \geq 0$ for all $k \in M$. Otherwise, we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = +\infty$. In this case, we have $V_1(\boldsymbol{f}, \boldsymbol{h}) \leq T$ if and only if $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$. The number of possible values for the state variable in the dynamic program above is the same as that for the dynamic program in (13). Thus, the number of possible values for the state variable in the dynamic program above is also given by the expression in (14), in which case, we can solve the dynamic program above in $O(mn(\log(n\frac{R_{\max}}{R_{\min}}))^m\,(\log(\Delta(\rho,n)\frac{V_{\max}}{V_{\min}}))^m/\rho^{2m})$ operations. Using the same discussion in Section 7 and earlier in this section, under space constraints across stages, we can choose $\rho = \epsilon/(8n)$ to construct an FPTAS with a running time of $O(mn^{2m+1}(\log(n\frac{R_{\max}}{R_{\min}}))^m\,(\log(n\frac{V_{\max}}{V_{\min}}))^m/\epsilon^{2m})$. We cannot reduce this running time by using the approach that we use in the dynamic program in (15), since we cannot characterize the value function above by using a threshold on one component of the state variable, as is done right before the dynamic program in (15). Note that the running time of our FPTAS under space constraints across stages is slightly worse than that under cardinality constraints across stages.

Space Constraints within Stages. Extending our approach to space constraints within stages appears to be difficult. Under space constraints within stages, similar to our approach under cardinality constraints within stages, we need an additional m-dimensional state variable in our dynamic program, which keeps track of the cumulative space consumption of the products offered in each stage. To obtain an FPTAS with a running time that is polynomial in the input size, we need to discretize the components of the additional state variable by using a geometric grid, but if we use such a discretization, then we cannot guarantee that we satisfy the constraints on the space consumptions of the products offered in each stage. It is simple to construct an FPTAS to obtain a solution that satisfies the constraints on the space consumptions with a multiplicative error of $1+\epsilon$ in running time that is polynomial in $1/\epsilon$, but a feasible solution, by its definition, must satisfy the hard space constraints. In the previous paragraph, under space constraints across stages, we do not have to use an additional state variable in our dynamic program. In particular, under space constraints across stages, we need to keep track of the total space consumption of the products offered over all stages, which is a scalar. In this case, we can "overload" the value function so that the value of the value function, itself, corresponds to the cumulative space consumption of the products offered over all stages. Under space constraints within stages, however, we need to keep track of the space consumption of the products offered in each stage separately, which is an m-dimensional quantity, preventing us from "overloading" the value function.

9 Numerical Experiments

We give two sets of numerical experiments to test the effectiveness of our FTPAS. In the first set of numerical experiments, we work with randomly generated test problems. In the second set of numerical experiments, we use the data coming from a survey on the appointment slot choices of the patients in a clinic. We provide all of our test problems as online supplement. Throughout, our goal is to understand how the practical performance of our FPTAS compares with its theoretical guarantee. We begin by formulating a linear program that provides an upper bound on the optimal expected revenue in problem (1). So, we can compare the upper bound on the optimal expected revenue with the expected revenue from the solution obtained by our FPTAS to assess the ex post optimality gap of the solution. Following the linear program, we give our numerical experiments.

9.1 Upper Bound on the Optimal Expected Revenue

We construct a linear program that we can use to obtain an upper bound on the optimal expected revenue in problem (1). All test problems in our numerical experiments have two stages. So, for notational brevity, we give our linear program for the case with two stages, but we discuss the extension to more than two stages at the end of this section. To formulate our linear program, we use the decision variable $x_i^k \in \{0,1\}$, where $x_i^k = 1$ if and only if we offer product i in stage k. Noting that we have $m = 2$ stages, we write problem (1) equivalently as

$$\widehat{Z} \quad = \max_{\{\boldsymbol{x}_i \,:\, i \in N\} \in \{0,1\}^{n\times m}} \left\{ \frac{\sum_{i\in N} r_i\, v_i^1\, x_i^1}{1+\sum_{i\in N} v_i^1\, x_i^1} + \frac{1}{1+\sum_{i\in N} v_i^1\, x_i^1}\, \frac{\sum_{i\in N} r_i\, v_i^2\, x_i^2}{1+\sum_{i\in N} v_i^2\, x_i^2} \;:\; \sum_{k\in M} x_i^k \le 1 \;\forall\, i \in N \right\}. \tag{16}$$

Our linear program is based on guessing the values of $\sum_{i\in N} v_i^1\, x_i^1$ and $\sum_{i\in N} r_i\, v_i^2\, x_i^2/(1+\sum_{i\in N} v_i^2\, x_i^2)$ above in an optimal solution.

Using $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ to denote an optimal solution to problem (16), we let the intervals $[\underline{\nu}, \overline{\nu}]$ and $[\underline{\sigma}, \overline{\sigma}]$ be such that $\sum_{i\in N} v_i^1\, \widehat{x}_i^1 \in [\underline{\nu}, \overline{\nu}]$ and $\frac{\sum_{i\in N} r_i\, v_i^2\, \widehat{x}_i^2}{1+\sum_{i\in N} v_i^2\, \widehat{x}_i^2} \in [\underline{\sigma}, \overline{\sigma}]$. Consider the problem

$$\widehat{Z}(\underline{\nu}, \overline{\nu}, \underline{\sigma}, \overline{\sigma}) \quad = \max_{\{\boldsymbol{x}_i \,:\, i \in N\} \in [0,1]^{n\times m}} \left\{ \frac{\sum_{i\in N} r_i\, v_i^1\, x_i^1}{1+\underline{\nu}} + \frac{1}{1+\underline{\nu}}\, \overline{\sigma} \;:\; \sum_{k\in M} x_i^k \le 1 \;\forall\, i \in N, \;\; \sum_{i\in N} v_i^1\, x_i^1 \le \overline{\nu},\; \frac{\sum_{i\in N} r_i\, v_i^2\, x_i^2}{1+\sum_{i\in N} v_i^2\, x_i^2} \ge \underline{\sigma} \right\}. \tag{17}$$

In the problem above, the values of $\underline{\nu}, \overline{\nu}, \underline{\sigma}$ and $\overline{\sigma}$ are fixed. Therefore, the objective along with the first and second constraints are linear in the decision variables. We can write the third constraint as $\sum_{i\in N}(r_i - \underline{\sigma})\, v_i^2\, x_i^2 \ge \underline{\sigma}$, so that the third constraint is linear in the decision variables as well. Since the decision variable x_i^k takes values in the interval $[0,1]$ for all $i \in N$, $k \in M$, problem (17) is a linear program. We proceed to argue that the optimal objective value of the linear program in (17) is an upper bound on the optimal objective value of problem (16). Letting $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ be an optimal solution to problem (16), by the definition of the intervals $[\underline{\nu}, \overline{\nu}]$ and $[\underline{\sigma}, \overline{\sigma}]$, we have $\sum_{i\in N} v_i^1\, \widehat{x}_i^1 \in [\underline{\nu}, \overline{\nu}]$ and $\frac{\sum_{i\in N} r_i\, v_i^2\, \widehat{x}_i^2}{1+\sum_{i\in N} v_i^2\, \widehat{x}_i^2} \in [\underline{\sigma}, \overline{\sigma}]$. Since $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ is an optimal solution to problem (16), we also have $\sum_{k\in M} \widehat{x}_i^k \le 1$ for all $i \in N$. Therefore, the solution $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ is feasible to problem (17). Since $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ is an optimal solution to problem (16), noting that $\sum_{i\in N} v_i^1\, \widehat{x}_i^1 \ge \underline{\nu}$ and $\frac{\sum_{i\in N} r_i\, v_i^2\, \widehat{x}_i^2}{1+\sum_{i\in N} v_i^2\, \widehat{x}_i^2} \le \overline{\sigma}$, the optimal objective value of problem (16) satisfies

$$\widehat{Z} = \frac{\sum_{i\in N} r_i\, v_i^1\, \widehat{x}_i^1}{1+\sum_{i\in N} v_i^1\, \widehat{x}_i^1} + \frac{1}{1+\sum_{i\in N} v_i^1\, \widehat{x}_i^1}\, \frac{\sum_{i\in N} r_i\, v_i^2\, \widehat{x}_i^2}{1+\sum_{i\in N} v_i^2\, \widehat{x}_i^2} \le \frac{\sum_{i\in N} r_i\, v_i^1\, \widehat{x}_i^1}{1+\underline{\nu}} + \frac{1}{1+\underline{\nu}}\, \overline{\sigma}.$$

The expression on the right side above is the objective value of the linear program in (17) evaluated at the solution $\{\widehat{\boldsymbol{x}}_i : i \in N\}$. Therefore, there exists a feasible solution to the linear program in (17) with the corresponding objective value that is no less than the optimal objective value of problem (16). Therefore, the optimal objective value $\widehat{Z}(\underline{\nu}, \overline{\nu}, \underline{\sigma}, \overline{\sigma})$ of the linear program in (17) is no less than the optimal objective value of problem (16), as desired. We cannot use the linear program in (17) immediately to obtain an upper bound on the optimal objective value of problem (16) since we cannot come up with the intervals $[\underline{\nu}, \overline{\nu}]$ and $[\underline{\sigma}, \overline{\sigma}]$ without knowing an optimal solution to problem (16). To get around this difficulty, we solve the linear program in (17) for multiple guesses for the intervals that can potentially include the values $\sum_{i \in N} v_i^1 \, \widehat{x}_i^1$ and $\frac{\sum_{i \in N} r_i \, v_i^2 \, \widehat{x}_i^2}{1 + \sum_{i \in N} v_i^2 \, \widehat{x}_i^2}$.

Let $V_{\max} = \max\{v_i^1 : i \in N\}$ and $r_{\max} = \max\{r_i : i \in N\}$. So, the value of $\sum_{i \in N} v_i^1 \, \widehat{x}_i^1$ cannot exceed $n V_{\max}$. Viewing $\frac{\sum_{i \in N} r_i \, v_i^2 \, \widehat{x}_i^2}{1 + \sum_{i \in N} v_i^2 \, \widehat{x}_i^2}$ as the weighted average of the revenues of the products, its value cannot exceed $r_{\max}$. We partition the interval $[0, n V_{\max}]$ by using the $K+1$ points $0 = \nu^0 \leq \nu^1 \leq \ldots \leq \nu^K = n V^{\max}$. Similarly, we partition the interval $[0, r_{\max}]$ by using the $L+1$ points $0 = \sigma^0 \leq \sigma^1 \leq \ldots \leq \sigma^L = r^{\max}$. Even if we do not know an optimal solution $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ to problem (16), the value $\sum_{i \in N} v_i^1 \, \widehat{x}_i^1$ lies in one of the intervals $\{[\nu^{k-1}, \nu^k] : k = 1, \ldots, K\}$, whereas the value $\frac{\sum_{i \in N} r_i \, v_i^2 \, \widehat{x}_i^2}{1 + \sum_{i \in N} v_i^2 \, \widehat{x}_i^2}$ lies in one of the intervals $\{[\sigma^{\ell-1}, \sigma^\ell] : \ell = 1, \ldots, L\}$. Thus, if we solve the linear program in (17) with $[\underline{\nu}, \overline{\nu}] = [\nu^{k-1}, \nu^k]$ and $[\underline{\sigma}, \overline{\sigma}] = [\sigma^{\ell-1}, \sigma^\ell]$ to compute $\widehat{Z}(\nu^{k-1}, \nu^k, \sigma^{\ell-1}, \sigma^\ell)$ for all $k = 1, \ldots, K$, $\ell = 1, \ldots, L$, then $\max\{\widehat{Z}(\nu^{k-1}, \nu^k, \sigma^{\ell-1}, \sigma^\ell) : k = 1, \ldots, K, \ \ell = 1, \ldots, L\}$ is an upper bound on the optimal objective value of problem (16).

We obtain an upper bound on the optimal objective value of problem (16) for any choice of the points $0 = \nu^0 \leq \nu^1 \leq \ldots \leq \nu^K = n V^{\max}$ and $0 = \sigma^0 \leq \sigma^1 \leq \ldots \leq \sigma^L = r^{\max}$. In our numerical experiments, we choose these points such that $\nu^k - \nu^{k-1} = \sigma^\ell - \sigma^{\ell-1} = 0.01$. Using our approach, we obtain reasonably tight upper bounds for our test problems, but a theoretical characterization of the tightness of the upper bounds requires characterizing the integrality gap of a multiple knapsack problem, which is difficult. We can extend our approach to the case where there are more than two stages in the choice process of the customers. This extension requires guessing the intervals that can potentially include the values $\sum_{i \in N} v_i^k \, \widehat{x}_i^k$ for all $k = 1, \ldots, m-1$, $\sum_{i \in N} r_i \, v_i^k \, \widehat{x}_i^k$ for all $k = 2, \ldots, m-1$ and $\frac{\sum_{i \in N} r_i \, v_i^m \, \widehat{x}_i^m}{1 + \sum_{i \in N} v_i^m \, \widehat{x}_i^m}$. Therefore, the number of operations to compute an upper bound increases exponentially with the number of stages.

9.2 Randomly Generated Test Problems

In this section, we work with a large number of test problems that are randomly generated and test the performance of our FPTAS on these test problems.

Experimental Setup. We have $n = 18$ products and $m = 2$ stages in all of our test problems. The preference weight of a product is the same in both stages. Using v_i to denote the preference weight of product i, to generate the preference weights of the products, we sample θ_i from the uniform distribution over $[1, 10]$ for all $i \in N$ and set the preference weight of product i as $v_i = (1 - P_0) \, \theta_i / (P_0 \sum_{j \in N} \theta_j)$, where P_0 is a parameter that we vary. So, if we offer all products in a particular stage, then the probability of no purchase is $1/(1 + \sum_{i \in N} v_i) = 1/(1 + (1 - P_0)/P_0) =$

P_0. Thus, the parameter P_0 controls the likelihood that a customer does not make a purchase in a stage. To generate the revenues of the products, for all $i \in N$, we set $r_i = 0.3$ or $r_i = 1$ with equal probabilities. We also tested our FPTAS with revenues of the products uniformly generated over a bounded interval and its performance was even better. After generating the revenues $\{r_i : i \in N\}$ and preference weights $\{v_i : i \in N\}$ as described, we use two approaches to finalize their values. In the first approach, we simply leave the generated revenues and preference weights as they are. So, there is no relationship between the revenue and preference weight of a product. In the second approach, we reindex the revenues $\{r_i : i \in N\}$ and preference weights $\{v_i : i \in N\}$ so that $r_1 \geq r_2 \geq \ldots \geq r_n$ and $v_1 \leq v_2 \leq \ldots \leq v_n$. Thus, the more expensive products have smaller preference weights. We use $T \in \{N, O\}$ to denote the approach to finalize the revenues and preference weights, where N corresponds to no relationship between the revenues and preference weights and O corresponds to ordering the revenues and preference weights. Varying $P_0 \in \{0.05, 0.1, 0.2, 0.3\}$ and $T \in \{N, O\}$, we obtain eight parameter settings. In each parameter setting, we generate 50 individual test problems by using the approach just discussed.

Benchmark. As a benchmark, we use an iterative exchange heuristic. In this heuristic, we represent a solution by using the vector $\boldsymbol{z} = (z_1, \ldots, z_n) \in \{0, 1, 2\}^n$, where we have $z_i = 0$ if we do not offer product i, $z_i = 1$ if we offer product i in stage 1 and $z_i = 2$ if we offer product i in stage 2. In the iterative exchange heuristic, at the first iteration, we have the solution $\boldsymbol{z}^1 = (0, \ldots, 0) \in \mathbb{Z}_+^n$, which does not offer any of the products. At iteration ℓ, given that we have the solution $\boldsymbol{z}^\ell$, we check whether replacing each component z_i^ℓ of the solution $\boldsymbol{z}^\ell$ by an element of $\{0, 1, 2\} \setminus \{z_i^\ell\}$ improves the expected revenue from the solution $\boldsymbol{z}^\ell$. When we find one such component z_i^ℓ and one such element of $\{0, 1, 2\} \setminus \{z_i^\ell\}$, we replace this component by this element to obtain a solution with a larger expected revenue, which yields the solution $\boldsymbol{z}^{\ell+1}$ that we have at iteration $\ell + 1$. If replacing any component z_i^ℓ of the solution $\boldsymbol{z}^\ell$ by any element of $\{0, 1, 2\} \setminus \{z_i^\ell\}$ does not improve the expected revenue from the solution $\boldsymbol{z}^\ell$, then we stop.

Numerical Results. In our numerical experiments, we use our FPTAS to obtain solutions with performance guarantees of $\epsilon = 1/4$ and $\epsilon = 1/2$. Therefore, the expected revenues from the solutions obtained by our FPTAS are guaranteed to be at least 75% and 50% of the optimal expected revenue. By the discussion at the end of Section 5, to obtain such performance guarantees, we need to execute the Approx algorithm with $\rho = \epsilon/(8n)$. We give our main numerical results in Table 1. In this table, the first column shows the parameter setting by using the pair (P_0, T), where P_0 and T are as discussed in our experimental setup. There are three blocks of four columns in the rest of the table. The first block of columns focuses on the performance of our FPTAS with $\epsilon = 1/4$, corresponding to a 75% performance guarantee. Recall that we generate 50 test problems in each parameter setting. For each test problem, we use our FPTAS to obtain an approximate solution. For each test problem, we also use the linear program in (17) to obtain an upper bound on the optimal expected revenue. In the first block of columns in Table 1, the first column shows the average percent gap between the upper bound on the optimal expected revenue and the expected revenue from the solution obtained by our FPTAS, where the average is computed over the 50 test problems

Par.	FPTAS with $\epsilon = 1/4$				Iterative Exchange Heuristic				FPTAS with $\epsilon = 1/2$			
(P_0, T)	Avg.	Max.	75th	95th	Avg.	Max.	75th	95th	Avg.	Max.	75th	95th
$(N, 0.05)$	0.47%	1.39%	0.63%	1.02%	0.73%	11.59%	0.67%	1.15%	0.47%	1.39%	0.63%	1.02%
$(O, 0.05)$	0.56%	1.77%	0.73%	1.17%	1.34%	12.75%	0.75%	6.69%	0.56%	1.77%	0.73%	1.17%
$(N, 0.1)$	0.71%	1.69%	0.82%	1.12%	2.22%	17.93%	1.01%	14.73%	0.71%	1.69%	0.82%	1.12%
$(O, 0.1)$	0.88%	2.34%	0.92%	2.00%	2.51%	16.24%	2.34%	10.43%	0.88%	2.34%	0.92%	2.00%
$(N, 0.2)$	1.08%	1.76%	1.21%	1.67%	2.61%	13.94%	2.54%	9.59%	1.08%	1.76%	1.21%	1.67%
$(O, 0.2)$	1.39%	2.73%	1.72%	1.95%	2.23%	10.58%	1.76%	6.87%	1.39%	2.73%	1.72%	1.95%
$(N, 0.3)$	1.75%	3.59%	1.95%	3.06%	2.35%	8.33%	2.89%	6.08%	1.75%	3.59%	1.95%	3.06%
$(O, 0.3)$	1.77%	3.54%	2.10%	3.06%	2.15%	7.43%	2.59%	4.98%	1.77%	3.54%	2.10%	3.06%
Avg.	1.08%	2.35%	1.26%	1.88%	2.02%	12.35%	1.82%	7.57%	1.08%	2.35%	1.26%	1.88%

Table 1: Performance of our FPTAS and the iterative exchange heuristic.

in a parameter setting. In particular, for test problem k, letting Rev^k be the expected revenue from the solution obtained by our FPTAS and Upp^k be the upper bound on the optimal expected revenue, the first column shows the average of the data $\{100 \times (\mathsf{Upp}^k - \mathsf{Rev}^k)/\mathsf{Upp}^k : k = 1, \ldots, 50\}$. The second, third and fourth columns show the maximum, 75th percentile and 95th percentile of the same data. The second and third blocks of columns in Table 1 have the same interpretation as the first block, but the second block focuses on the performance of the iterative exchange heuristic, whereas the third block focuses on the performance of our FPTAS with $\epsilon = 1/2$.

We make three observations in the results in Table 1. First, our FPTAS is able to obtain high quality solutions and its practical performance can be substantially better than its theoretical performance guarantee. When we execute our FPTAS with $\epsilon = 1/4$ corresponding to a performance guarantee of 75%, the average optimality gap of the solutions that we obtain is no larger than 1.08%. Second, the performance of the iterative exchange heuristic noticeably lags behind that of our FPTAS. Also, the heuristic can be unreliable. In particular, there are test problems where we have a gap of as much as 17.93% between the upper bound on the optimal expected revenue and the expected revenue from the solution obtained by the heuristic. Third, the performance of our FPTAS with $\epsilon = 1/4$ and $\epsilon = 1/2$ is indistinguishable up to the two decimal digits that we report, but we occasionally get different solutions with $\epsilon = 1/4$ and $\epsilon = 1/2$. Intuitively, our theoretical analysis accumulates the error in the dynamic program in (13) rather conservatively. Thus, even when we use a large value of ϵ corresponding to a crude performance guarantee in our FPTAS, we may still get solutions with high quality, but of course, we cannot give a priori guarantees beyond 75% and 50%, unless we tighten our theoretical analysis further, which appears to be difficult.

Another interesting question is the benefit in the expected revenue that we obtain by letting the customers make choices in two stages. To answer this question, we compute the optimal solution when we offer products in only one of the two stages, but not in the other stage. Since the preference weights of the products in the two stages are the same, this approach is equivalent to computing the optimal set of products to offer in the first stage when we do not offer any products in the second stage. In this case, the expected revenue takes the same form as that under the standard multinomial logit model and there exist efficient algorithms to find the optimal

Par. (P_0, T)	Performance Gap Avg.	Max.	75th	95th
$(N, 0.05)$	7.71%	32.53%	9.21%	11.99%
$(O, 0.05)$	10.29%	39.27%	10.16%	26.59%
$(N, 0.1)$	12.53%	36.14%	12.43%	25.96%
$(O, 0.1)$	15.72%	36.33%	18.88%	35.14%
$(N, 0.2)$	15.26%	31.95%	17.58%	27.45%
$(O, 0.2)$	20.00%	34.59%	28.68%	34.11%
$(N, 0.3)$	16.61%	29.22%	22.95%	26.01%
$(O, 0.3)$	18.11%	30.08%	24.74%	29.20%
Avg.	14.53%	33.76%	18.08%	27.06%

Table 2: Performance improvement when we make offers in two stages rather than in one stage.

solution; see Talluri and van Ryzin (2004). For all of the test problems in our experimental setup, we computed the optimal solution when we offer products only in the first stage. We give our numerical results in Table 2. The first column in this table shows the parameter setting by using the pair (P_0, T). For test problem k, we let Rev^k be the expected revenue from the solution obtained by our FPTAS with $\epsilon = 1/4$ and One^k be the expected revenue from the optimal solution that we obtain when we offer products only in the first stage. In this case, the second, third, fourth and fifth columns in Table 2 show the average, maximum, 75th percentile and 95th percentile of the data $\{100 \times (\mathsf{Rev}^k - \mathsf{One}^k)/\mathsf{Rev}^k : k = 1, \ldots, 50\}$, providing summary statistics for the percent improvement in the expected revenue when we offer products in two stages. Over all of our test problems, offering products in two stages provides an average improvement of 14.53% in the expected revenue. There are test problems where the improvement reaches 39.27%.

The CPU time for the iterative exchange heuristic is on the order of milliseconds. If we use our FPTAS with $\epsilon = 1/2$, then the average CPU time per test problem is 19.8 seconds, whereas if we use our FPTAS with $\epsilon = 1/4$, then the average CPU time per test problem increases by about a factor of ten. In our theoretical running time analysis, decreasing ϵ by a factor of two increases the running time by a factor of eight. The additional CPU time is spent on allocating memory, which is not accounted for in our theoretical running time analysis. Ultimately, our FPTAS has longer CPU times, but it can provide significantly better solutions than the iterative exchange heuristic. Also, the CPU times for our FPTAS are reasonable if we compute solutions in an offline fashion, which is the case in many applications. Lastly, our FPTAS can continue to provide high solution quality with even larger values of ϵ. In particular, we repeated our numerical experiments with $\epsilon = 3/4$ and the performance of our FPTAS on all our test problems remained almost unchanged, whereas its average CPU time per test problem was 6.0 seconds.

9.3 Test Problems Based on a Survey for Appointment Slot Choices

In this section, we build on the survey conducted by Feldman et al. (2014) regarding the appointment slot choices of the patients visiting Farrell Community Health Center in New York City. In their work, Feldman et al. (2014) use the data provided by the survey to fit a multinomial logit model with a single stage. In our numerical experiments, we augment the data provided by their survey to use it under the multinomial logit model with two stages.

Experimental Setup. We particularly focus on one question in the survey. The question describes a number of symptoms that the patient is hypothetically going through, including heavy cough and sharp chest pain. There are six possible days and three possible time blocks on each day, resulting in 18 possible appointment slots. Among the 18 possible appointment slots, the question offers the patient a randomly chosen set of six appointment slots, along with the option of seeking care elsewhere. The patient picks one of the offered alternatives. In the survey, the patients are offered one set of appointment slots, all in one stage. To use the data provided by the survey under the multinomial logit model with two stages, we artificially augment the data as follows. We randomly split the set of six appointment slots offered to each patient into two partitions, each containing three appointment slots, so that the first partition is offered in the first stage, whereas the second partition is offered in the second stage. So, for each patient, we have the two sets of appointment slots offered in the two stages, along with the choice of the patient. Using maximum likelihood estimation, we fit a multinomial logit model with two stages. Feldman et al. (2014) focus on the appointment slot choices among the different days, but their survey also includes more granular data on the appointment slot choices among the different time blocks on each day, which is what we use. Also, since we augmented the data, we caution the reader against comparing the results of our numerical experiments with the current operations of the clinic.

In the appointment scheduling setting, a possible objective is to maximize the probability that a patient schedules an appointment, but for purposes of quality and continuity of care, it is preferable to get the patients to schedule earlier appointments. We use two revenue structures in our numerical experiments. In the first revenue structure, we set $r_i = 1$ for all $i \in N$. In this case, we maximize the probability that a patient schedules an appointment without making a distinction between scheduling earlier or later appointments. In the second revenue structure, we use N_1 to denote the appointment slots in the first three days and N_2 to denote the appointment slots in the last three days. We set $r_i = 1$ for all $i \in N_1$ and $r_i = 0.3$ for all $i \in N_2$. In this case, we put a larger weight on scheduling earlier appointments. We use $R \in \{U, E\}$ to denote the revenue structure, where U corresponds to having a uniform revenue of one from each appointment and E corresponds to putting a larger weight on scheduling earlier appointments. Also, recall that we augment the data provided by the survey by randomly splitting the set of appointment slots offered to each patient into two partitions. Using six different random seeds to carry out the splitting process, we obtain six different datasets. We fit a multinomial logit model with two stages to each one of the datasets, yielding six multinomial logit models. The approach that we use to fit the multinomial logit model is similar to the one in Vulcano et al. (2012). Letting ℓ denote the multinomial logit model that we obtain, varying $R \in \{U, E\}$ and $\ell \in \{1, \ldots, 6\}$, we obtain 12 test problems.

Numerical Results. We give our numerical results in Table 3. In this table, the first column shows the parameter setting for each of the 12 test problems by using the pair (R, ℓ) with $R \in \{U, E\}$

Par. (R,ℓ)	FPTAS $\epsilon=1/4$	Single Stage	Par. (R,ℓ)	FPTAS $\epsilon=1/4$	Single Stage
$(U,1)$	0.64%	10.15%	$(U,4)$	0.65%	10.28%
$(E,1)$	0.74%	10.90%	$(E,4)$	0.75%	11.03%
$(U,2)$	0.66%	10.24%	$(U,5)$	0.66%	10.36%
$(E,2)$	0.75%	10.98%	$(E,5)$	0.79%	11.12%
$(U,3)$	0.65%	10.23%	$(U,6)$	0.63%	10.12%
$(E,3)$	0.77%	11.00%	$(E,6)$	0.75%	10.91%
Avg.	0.70%	10.58%	Avg.	0.71%	10.64%

Table 3: Performance of our FPTAS and the approach that offers appointment slots only in the first stage.

and $\ell \in \{1,\ldots,6\}$. The second column focuses on the performance of our FPTAS with $\epsilon = 1/4$, corresponding to a 75% performance guarantee. In particular, the second column shows the percent gap between the upper bound on the optimal expected revenue and the expected revenue from the solution obtained by our FPTAS. The third column focuses on the expected revenues that we obtain when we offer appointment slots only in the first stage, but not in the second stage. In particular, the third column shows the percent gap between the upper bound on the optimal expected revenue and the expected revenue from the optimal solution under the constraint that we offer appointment slots only in the first stage. We executed our FPTAS with $\epsilon = 1/2$ and $\epsilon = 3/4$, as well as the iterative exchange heuristic. Using $\epsilon = 1/2$ or $\epsilon = 3/4$ for our FPTAS did not change the performance of our FPTAS. In 10 out of the 12 test problems, the performance of the iterative exchange heuristic was identical to that of our FPTAS. In the remaining two test problems, the expected revenue obtained by the iterative exchange heuristic lagged behind that obtained by our FPTAS, but by less than 0.01%. The results in Table 3 indicate that our FPTAS can obtain solutions with high quality and its performance in practice can be significantly better than what is predicted by its theoretical performance guarantee. In the table, the solutions obtained by our FPTAS have optimality gaps no larger than 0.79%. On the other hand, offering appointment slots only in the first stage may incur losses in the expected revenues by more than 10%. Closing this section, we note that the goal of the numerical experiments that we give in this section is to test the quality of the solutions obtained by our FPTAS and the benefits provided by taking advantage of the second stage in the choice process. In particular, since the data that we use is based on a survey that offers the sets of appointment slots altogether in one stage and we artificially augment the data to fit a multinomial logit model with two stages, our numerical experiments do not shed light into the benefits of sequential offerings in a clinical setting. Testing the clinical benefits of sequential offerings is outside our scope, which requires a survey where we collect data by actually offering the sets of appointment slots in multiple stages.

10 Conclusions

In this paper, we studied assortment optimization problems under the multinomial logit model, where the choice process of the customer takes place in multiple stages. There are several directions for future work. The running time of our FPTAS depends on $R_{\max}/R_{\min}$ and $V_{\max}/V_{\min}$. It would be interesting to see whether one can develop a strongly polynomial time algorithm, whose running time does not depend on these quantities, possibly by building on the fact that the union of the optimal assortments offered in all stages is nested by revenue. Also, the running time of our FPTAS is polynomial in the number of products but exponential in the number of stages. A useful line of research is to develop algorithms with running times polynomial in the number of stages. Our effort in this regard has been unfruitful so far and designing algorithms with running times polynomial in the number of stages appears to need a new line of attack. Lastly, as discussed in Section 8, our approach does not immediately extend to space constraints within stages. It is interesting to study approximation schemes under space constraints within stages.

Acknowledgements. The authors thank the associate editor and two anonymous referees whose comments improved the paper substantially.

References

Aouad, A., Farias, V. and Levi, R. (2016), Assortment optimization under consider-then-choose choice models, Technical report, MIT, Massachusetts, MA.

Blanchet, J., Gallego, G. and Goyal, V. (2016), 'A Markov chain approximation to choice modeling', *Operations Research* **64**(4), 886–905.

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. (2014), Near-optimal algorithms for capacity constrained assortment optimization, Technical report, Columbia University, New York, NY.

Feldman, J., Liu, N., Topaloglu, H. and Ziya, S. (2014), 'Appointment scheduling under patient preference and no-show behavior', *Operations Research* **62**(4), 794–811.

Feldman, J. and Topaloglu, H. (2018), 'Technical note: Capacitated assortment optimization under the multinomial logit model with nested consideration sets', *Operations Research* **66**(2), 380–391.

Flores, A., Berbeglia, G. and van Hentenryck, P. (2019), 'Assortment optimization under the sequential multinomial logit model', *European Journal of Operational Research* **273**(3), 1052–1064.

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., Li, A., Truong, V. A. and Wang, X. (2016), Approximation algorithms for product framing and pricing, Technical report, Columbia University, New York, NY.

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

Garey, M. and Johnson, D. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, NY.

Liu, N., van de Ven, P. M. and Zhang, B. (2017), Managing appointment booking under customer choices, Technical report, Boston College, Boston, MA.

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.

Mittal, S. and Schulz, A. S. (2013), 'A general framework for designing approximation schemes for combinatorial optimization problems with many objectives combined into one', *Operations Research* **61**(2), 389–397.

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. (2014), 'Assortment optimization under the multinomial logit model with random choice parameters', *Production and Operations Management* **23**(11), 2023–2039.

Rusmevichientong, P. and Topaloglu, H. (2012), 'Robust assortment optimization in revenue management under the multinomial logit choice model', *Operations Research* **60**(4), 865–882.

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

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. (2012), 'Capacitated assortment and price optimization under the multinomial logit model', *Operations Research Letters* **40**(6), 492–497.

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

A Appendix: Upper Bound on State Variable

In this section, we give a proof for Lemma 6. First, we show that if $\widehat{f}^k > \lceil nR_{\max} \rceil$, then we have $f^k_{n+1} > 0$. Since $f^k_{i+1} = \lceil f^k_i - r_i\, v^k_i\, x^k_i \rceil$, we have $f^k_{i+1} \geq f^k_i - r_i\, v^k_i\, x^k_i$. Adding this inequality over all $i \in N$ and noting that $f^k_1 = \widehat{f}^k$, along with the definition of $R_{\max}$, we obtain $f^k_{n+1} \geq \widehat{f}^k - \sum_{i \in N} r_i\, v^k_i\, x^k_i \geq \widehat{f}^k - nR_{\max} \geq \widehat{f}^k - \lceil nR_{\max} \rceil$. In this case, the last inequality implies that if $\widehat{f}^k > \lceil nR_{\max} \rceil$, then we have $f^k_{n+1} > 0$. Second, we show that if $\widehat{h}^k \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$, then we have $h^k_{n+1} \geq 0$. We claim that if $h^k_i \geq \Delta(\rho, n+1-i)\, V_{\max}$, then $h^k_{i+1} \geq \Delta(\rho, n-i)\, V_{\max}$. To see the claim, if $h^k_i \geq \Delta(\rho, n+1-i)\, V_{\max}$, then we have

$$h^k_{i+1} = \lfloor h^k_i - v^k_i\, x^k_i \rfloor \geq \frac{1}{1+\rho}\,(h^k_i - v^k_i\, x^k_i) \geq \frac{1}{1+\rho}\,(\Delta(\rho, n+1-i) - 1)\, V_{\max}$$

$$= \frac{\frac{(1+\rho)^{n+1-i}-1}{\rho} - 1}{1+\rho}\, V_{\max} = \frac{(1+\rho)^{n-i} - 1}{\rho}\, V_{\max} = \Delta(\rho, n-i)\, V_{\max},$$

where the first inequality holds since $h_i - v^k_i\, x^k_i \geq \Delta(\rho, n+1-i)\, V_{\max} - V_{\max} \geq 0$ and $\lfloor x \rfloor \geq x/(1+\rho)$ for any $x \in \Re_+$ and the second equality is by the definition of $\Delta(\rho, n)$. The chain of inequalities above establishes the claim. Using the claim, if $h^k_1 \geq \Delta(\rho, n)\, V_{\max}$, then $h^k_2 \geq \Delta(\rho, n-1)\, V_{\max}$, but using the claim once more, if $h^k_2 \geq \Delta(\rho, n-1)\, V_{\max}$, then $h^k_3 \geq \Delta(\rho, n-2)\, V_{\max}$. Using the claim successively, if $h^k_1 \geq \Delta(\rho, n)\, V_{\max}$, then $h^k_{n+1} \geq \Delta(\rho, 0)\, V_{\max}$. Since $h^k_1 = \widehat{h}^k$, if $\widehat{h}^k \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$, then $h^k_1 = \widehat{h}^k \geq \Delta(\rho, n)\, V_{\max}$, but if $h^k_1 \geq \Delta(\rho, n)\, V_{\max}$, then $h^k_{n+1} \geq \Delta(\rho, 0)\, V_{\max}$. By the definition of $\Delta(\rho, n)$, we have $\Delta(\rho, 0) = 0$, so it follows that if $\widehat{h}^k \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$, then $h^k_{n+1} \geq 0$.

B Appendix: Computation of Thresholds

In the next lemma, we show the relationship between the value functions $\{V_i(\cdot, \cdot) : i \in N\}$ and $\{J_i(\cdot, \cdot) : i \in N\}$ that are computed through the dynamic programs in (13) and (15).

Lemma 8 *For any $(\boldsymbol{f}_i, \boldsymbol{h}_i) \in \mathsf{Dom}^m_+ \times \mathsf{Dom}^m_+$ and $i \in N$, if $f^1_i \leq \lfloor J_i(\boldsymbol{f}^{-1}_i, \boldsymbol{h}_i) \rfloor$, then we have $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = 0$. Similarly, if $f^1_i > \lfloor J_i(\boldsymbol{f}^{-1}_i, \boldsymbol{h}_i) \rfloor$, then we have $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = -\infty$.*

Proof. We use induction over the products to show that if $f^1_i \leq \lfloor J_i(\boldsymbol{f}^{-1}_i, \boldsymbol{h}_i) \rfloor$, then we have $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = 0$ for any $(\boldsymbol{f}_i, \boldsymbol{h}_i) \in \mathsf{Dom}^m_+ \times \mathsf{Dom}^m_+$ and $i \in N \cup \{n+1\}$. For any $(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) \in \mathsf{Dom}^m_+ \times \mathsf{Dom}^m_+$, since $f^1_{n+1} \geq 0$ and $J_{n+1}(\cdot, \cdot)$ takes only the value zero or $-\infty$, if $f^1_{n+1} \leq \lfloor J_{n+1}(\boldsymbol{f}^{-1}_{n+1}, \boldsymbol{h}_{n+1}) \rfloor$, then we must have $J_{n+1}(\boldsymbol{f}^{-1}_{n+1}, \boldsymbol{h}_{n+1}) = 0$ and $f^1_{n+1} \leq 0$. By the boundary condition of the dynamic program in (15), if $J_{n+1}(\boldsymbol{f}^{-1}_{n+1}, \boldsymbol{h}_{n+1}) = 0$, then we must have $f^k_{n+1} \leq 0$ for all $k \in M^{-1}$ and $h^k_{n+1} \geq 0$ for all $k \in M$. Thus, if $f^1_{n+1} \leq \lfloor J_{n+1}(\boldsymbol{f}^{-1}_{n+1}, \boldsymbol{h}_{n+1}) \rfloor$, then we must have $f^1_{n+1} \leq 0$, $f^k_{n+1} \leq 0$ for all $k \in M^{-1}$ and $h^k_{n+1} \geq 0$ for all $k \in M$, in which case, by the boundary condition of the dynamic program in (13), we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = 0$. Therefore, the result holds for product $n+1$. Next, we assume that the result holds for product $i+1$ and we show that the result holds for product i. Consider $(\boldsymbol{f}_i, \boldsymbol{h}_i) \in \mathsf{Dom}^m_+ \times \mathsf{Dom}^m_+$ such that $f^1_i \leq \lfloor J_i(\boldsymbol{f}^{-1}_i, \boldsymbol{h}_i) \rfloor$. We use $\widehat{\boldsymbol{x}}_i$ to denote an optimal solution to the problem on the right side of (15). Since $f_i^1 \le \lfloor J_i(\boldsymbol{f}_i^{-1}, \boldsymbol{h}_i) \rfloor$, noting the dynamic program in (15), we have

$$f_i^1 \le \left\lfloor r_i\, v_i^1\, \widehat{x}_i^1 + \left\lfloor J_{i+1}\left(\left\lceil \boldsymbol{f}_i^{-1} - \sum_{k \in M^{-1}} \boldsymbol{e}^k\, r_i\, v_i^k\, \widehat{x}_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k\, v_i^k\, \widehat{x}_i^k \right\rfloor \right) \right\rfloor \right\rfloor.$$

By a simple lemma, given as Lemma 9 below, for any $a, b \in \mathsf{Dom}$ and $\alpha \in \Re$, we have $a \le \lfloor \alpha + b \rfloor$ if and only if $\lceil a - \alpha \rceil \le b$. Thus, the inequality above implies that we have

$$\lceil f_i^1 - r_i\, v_i^1\, \widehat{x}_i^1 \rceil \le \left\lfloor J_{i+1}\left(\left\lceil \boldsymbol{f}_i^{-1} - \sum_{k \in M^{-1}} \boldsymbol{e}^k\, r_i\, v_i^k\, \widehat{x}_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k\, v_i^k\, \widehat{x}_i^k \right\rfloor \right) \right\rfloor.$$

For notational brevity, we let $f_{i+1}^k = \lceil f_i^k - r_i\, v_i^k\, \widehat{x}_i^k \rceil$ and $h_{i+1}^k = \lfloor h_i^k - v_i^k\, \widehat{x}_i^k \rfloor$ for all $k \in M$, in which case, the inequality above is equivalent to $f_{i+1}^1 \le \lfloor J_{i+1}(\boldsymbol{f}_{i+1}^{-1}, \boldsymbol{h}_{i+1}) \rfloor$, but by the induction argument, if $f_{i+1}^1 \le \lfloor J_{i+1}(\boldsymbol{f}_{i+1}^{-1}, \boldsymbol{h}_{i+1}) \rfloor$, then $V_{i+1}(\boldsymbol{f}_{i+1}, \boldsymbol{h}_{i+1}) = 0$. Therefore, noting the definitions of f_{i+1}^k and h_{i+1}^k, we have $V_{i+1}(\boldsymbol{f}_{i+1}, \boldsymbol{h}_{i+1}) = V_{i+1}(\lceil \boldsymbol{f}^k - \sum_{k \in M} \boldsymbol{e}^k\, r_i\, v_i^k\, \widehat{x}_i^k \rceil, \lfloor \boldsymbol{h}^k - \sum_{k \in M} \boldsymbol{e}^k\, v_i^k\, \widehat{x}_i^k \rfloor) = 0$. By last equality, the solution $\widehat{\boldsymbol{x}}_i$ provides an objective value of zero for the problem on the right side of (13). Since $V_i(\cdot, \cdot)$ takes only the value zero or $-\infty$ and there exists a solution to the problem on the right side of (13) that provides an objective value of zero, we must have $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = 0$, completing the induction argument. The discussion so far shows that if $f_i^1 \le \lfloor J_i(\boldsymbol{f}_i^{-1}, \boldsymbol{h}_i) \rfloor$, then we have $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = 0$ for any $(\boldsymbol{f}_i, \boldsymbol{h}_i) \in \mathsf{Dom}_+^m \times \mathsf{Dom}_+^m$ and $i \in N \cup \{n+1\}$, establishing the first statement in the lemma. The second statement uses a similar reasoning. $\square$

In the next lemma, we show a result that we use in the proof of Lemma 8.

Lemma 9 *For any $a, b \in \mathsf{Dom}$ and $\alpha \in \Re$, we have $a \le \lfloor \alpha + b \rfloor$ if and only if $\lceil a - \alpha \rceil \le b$.*

Proof. First, we show that if $a \le \lfloor \alpha + b \rfloor$, then we have $\lceil a - \alpha \rceil \le b$. If $a \le \lfloor \alpha + b \rfloor$, then $a \le \alpha + b$, so that $a - \alpha \le b$. Since $b \in \mathsf{Dom}$, having $a - \alpha \le b$ implies that $\lceil a - \alpha \rceil \le b$, as desired. Second, we show that if $\lceil a - \alpha \rceil \le b$, then we have $a \le \lfloor \alpha + b \rfloor$. If $\lceil a - \alpha \rceil \le b$, then $a - \alpha \le b$, so that $a \le \alpha + b$. Since $a \in \mathsf{Dom}$, having $a \le \alpha + b$ implies that $a \le \lfloor \alpha + b \rfloor$, as desired. $\square$

C Appendix: Nested by Revenue Sets

We show that there exists an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (1) that satisfies $\cup_{k \in M} \widehat{S}^k = \{i \in N : r_i \ge \widehat{\zeta}\}$ for some constant $\widehat{\zeta}$. In other words, the union of the sets offered over all stages is a nested by revenue set. Therefore, if we index the products such that $r_1 \ge r_2 \ge \ldots \ge r_n$, then an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (1) is of the form $\cup_{k \in M} \widehat{S}^k = \{1, \ldots, i\}$ for some $i \in N$. Although this result gives some insight into the structure of the optimal solution, it does not allow us to obtain an optimal solution to problem (1) efficiently, since this result does not specify the stage in which each product should be offered. To show that there exists an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (1) that satisfies $\cup_{k \in M} \widehat{S}^k = \{i \in N : r_i \ge \widehat{\zeta}\}$ for some constant $\widehat{\zeta}$, we use a recursive version of the objective function of problem (1). We use $R^\nu(S^\nu, \ldots, S^m)$ to denote the expected revenue obtained from a customer starting her choice process in stage ν when we offer the sets $S^\nu, \ldots, S^m$ in stages $\nu, \ldots, m$. Thus, noting the expected revenue expression in (1) and focusing only on the stages $\nu, \ldots, m$, $R^\nu(S^\nu, \ldots, S^m)$ is given by

$$R^\nu(S^\nu, \ldots, S^m) = \sum_{k=\nu}^{m} \left\{ \prod_{\ell=\nu}^{k} \frac{1}{1 + \sum_{i \in S^\ell} v_i^\ell} \right\} \sum_{i \in S^k} r_i \, v_i^k. \tag{18}$$

Comparing the expression above with (1), note that $R^1(S^1, \ldots, S^m)$ corresponds to the objective function of problem (1). In the next proposition, we show that there exists an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (1) that satisfies $\cup_{k \in M} \widehat{S}^k = \{i \in N : r_i \geq \widehat{\zeta}\}$ for some constant $\widehat{\zeta}$ and the constant $\widehat{\zeta}$ is given by $\min\{R^k(\widehat{S}^k, \ldots, \widehat{S}^m) : k \in M\}$.

Proposition 10 *There exists an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (1) that satisfies $\cup_{k \in M} \widehat{S}^k = \{i \in N : r_i \geq \widehat{\zeta}\}$, where $\widehat{\zeta} = \min\{R^k(\widehat{S}^k, \ldots, \widehat{S}^m) : k \in M\}$.*

Proof. Let $(\widehat{S}^1, \ldots, \widehat{S}^m)$ be an optimal solution to problem (1) with the largest cardinality, so that if $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ is another optimal solution, then $| \cup_{k \in M} \widehat{S}^k| \geq | \cup_{k \in M} \widetilde{S}^k|$. By (18), we have

$$\begin{aligned} R^\nu(S^\nu, \ldots, S^m) &= \frac{\sum_{i \in S^\nu} r_i \, v_i^\nu}{1 + \sum_{i \in S^\nu} v_i^\nu} + \frac{1}{1 + \sum_{i \in S^\nu} v_i^\nu} \sum_{k=\nu+1}^{m} \left\{ \prod_{\ell=\nu+1}^{k} \frac{1}{1 + \sum_{i \in S^\ell} v_i^\ell} \right\} \sum_{i \in S^k} r_i \, v_i^k \\ &= \frac{\sum_{i \in S^\nu} r_i \, v_i^\nu + R^{\nu+1}(S^{\nu+1}, \ldots, S^m)}{1 + \sum_{i \in S^\nu} v_i^\nu}. \end{aligned} \tag{19}$$

First, we show that $\cup_{k \in M} \widehat{S}^k \supseteq \{i \in N : r_i \geq \widehat{\zeta}\}$. To get a contradiction assume that there exists a product j such that $j \in \{i \in N : r_i \geq \widehat{\zeta}\}$ and $j \notin \cup_{k \in M} \widehat{S}^k$. For notational brevity, we let $\widehat{R}^k = R^k(\widehat{S}^k, \ldots, \widehat{S}^m)$. Since $j \in \{i \in N : r_i \geq \widehat{\zeta}\}$ and $\widehat{\zeta} = \min\{\widehat{R}^k : k \in M\}$, we have $r_j \geq \widehat{R}^\ell$ for some $\ell \in M$. Furthermore, since $j \notin \cup_{k \in M} \widehat{S}^k$, we have $j \notin \widehat{S}^\ell$. We define a solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ to problem (1) as $\widetilde{S}^k = \widehat{S}^k$ for all $k \in M \setminus \{\ell\}$ and $\widetilde{S}^\ell = \widehat{S}^\ell \cup \{j\}$. Since $j \notin \cup_{k \in M} \widehat{S}^k$, the product j is not offered in any stage in the solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$. Therefore, the product j is only offered in stage ℓ in the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$, so $(\widetilde{S}^1, \ldots, \widetilde{S}^m) \in \mathcal{F}$. Letting $\widetilde{R}^k = R(\widetilde{S}^k, \ldots, \widetilde{S}^m)$ for notational brevity, we observe that $\widehat{R}^k = \widetilde{R}^k$ for all $k = \ell + 1, \ldots, m$, since $R^k(S^k, \ldots, S^m)$ depends on $S^k, \ldots, S^m$ and $\widehat{S}^k = \widetilde{S}^k$ for all $k = \ell + 1, \ldots, m$. In this case, we have

$$\begin{aligned} \widetilde{R}^\ell - \widehat{R}^\ell = \frac{\sum_{i \in \widetilde{S}^\ell} r_i \, v_i^\ell + \widetilde{R}^{\ell+1}}{1 + \sum_{i \in \widetilde{S}^\ell} v_i^\ell} - \widehat{R}^\ell &= \frac{\sum_{i \in \widehat{S}^\ell} r_i \, v_i^\ell + r_j \, v_j^\ell + \widehat{R}^{\ell+1}}{1 + \sum_{i \in \widehat{S}^\ell} v_i^\ell + v_j^\ell} - \widehat{R}^\ell \\ &= \frac{\widehat{R}^\ell \, (1 + \sum_{i \in \widehat{S}^\ell} v_i^\ell) + r_j \, v_j^\ell}{1 + \sum_{i \in \widehat{S}^\ell} v_i^\ell + v_j^\ell} - \widehat{R}^\ell = \frac{(r_j - \widehat{R}^\ell) \, v_j^\ell}{1 + \sum_{i \in \widehat{S}^\ell} v_i^\ell + v_j^\ell} \geq 0, \end{aligned}$$

where the second equality uses the fact that $\widetilde{S}^\ell = \widehat{S}^\ell \cup \{j\}$ and $\widetilde{R}^{\ell+1} = \widehat{R}^{\ell+1}$, the third equality uses the fact $\widehat{R}^\ell \, (1 + \sum_{i \in \widehat{S}^\ell} v_i^\ell) = \sum_{i \in \widehat{S}^\ell} r_i \, v_i^\ell + \widehat{R}^{\ell+1}$ by (19) and the inequality follows from the fact that $r_j \geq \widehat{R}^\ell$. Therefore, we obtain $\widetilde{R}^\ell \geq \widehat{R}^\ell$. By (19), for all $k = 1, \ldots, \ell - 1$, we have $\widetilde{R}^k = (\sum_{i \in \widetilde{S}^k} r_i\, v_i^k + \widetilde{R}^{k+1})/(1 + \sum_{i \in \widetilde{S}^k} v_i^k) = (\sum_{i \in \widehat{S}^k} r_i\, v_i^k + \widetilde{R}^{k+1})/(1 + \sum_{i \in \widehat{S}^k} v_i^k)$, where we use the fact that $\widetilde{S}^k = \widehat{S}^k$. Similarly, we have $\widehat{R}^k = (\sum_{i \in \widehat{S}^k} r_i\, v_i^k + \widehat{R}^{k+1})/(1 + \sum_{i \in \widehat{S}^k} v_i^k)$ for all $k = 1, \ldots, \ell - 1$. Subtracting the two equalities, we obtain $\widetilde{R}^k - \widehat{R}^k = (\widetilde{R}^{k+1} - \widehat{R}^{k+1})/(1 + \sum_{i \in \widehat{S}^k} v_j^k)$ for all $k = 1, \ldots, \ell - 1$. In this case, having $\widetilde{R}^\ell \geq \widehat{R}^\ell$ implies that $\widetilde{R}^1 \geq \widehat{R}^1$. Therefore, the objective value provided by the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ for problem (1) is at least as large as the one provided by the solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$. Furthermore, $\cup_{k \in M} \widetilde{S}^k = \cup_{k \in M} \widehat{S}^k \cup \{j\}$, which contradicts the fact that $(\widehat{S}^1, \ldots, \widehat{S}^m)$ is an optimal solution to problem (1) with the largest cardinality.

Second, we show that $\cup_{k \in M} \widehat{S}^k \subseteq \{i \in N : r_i \geq \widehat{\zeta}\}$. To get a contradiction, assume that there exists a product j such that $j \in \cup_{k \in M} \widehat{S}^k$ and $j \notin \{i \in N : r_i \geq \widehat{\zeta}\}$. Since $j \in \cup_{k \in M} \widehat{S}^k$, we have $j \in \widehat{S}^\ell$ for some $\ell \in M$. Also, noting that $j \notin \{i \in N : r_i \geq \widehat{\zeta}\}$, we have $r_j < \widehat{\zeta} = \min\{\widehat{R}^k : k \in M\}$, which implies that $r_j < \widehat{R}^\ell$. We define a solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ to problem (1) as $\widetilde{S}^k = \widehat{S}^k$ for all $k \in M \setminus \{\ell\}$ and $\widetilde{S}^\ell = \widehat{S}^\ell \setminus \{j\}$. Since $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ and $\widetilde{S}^k \subseteq \widehat{S}^k$ for all $k \in M$, each product is offered in at most one stage in the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$, so $(\widetilde{S}^1, \ldots, \widetilde{S}^m) \in \mathcal{F}$. Using the same argument in the previous paragraph and noting that $r_j < \widehat{R}^\ell$, we can show that $\widetilde{R}^1 > \widehat{R}^1$, contradicting the fact that $(\widehat{S}^1, \ldots, \widehat{S}^m)$ is an optimal solution to problem (1). □

The proposition above indicates that there exists an optimal solution to problem (1) that has a structure similar to that of an optimal solution when there is a single stage in the choice process. This structure is adequate to obtain an optimal solution efficiently when there is a single stage, but it is not adequate even when there are as few as two stages.