MANAGEMENT SCIENCE
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0025-1909 | EISSN 1526-5501 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Vivek F. Farias
MIT Sloan, vivekf@mit.edu

Srikanth Jagabathula
EECS, MIT, jskanth@alum.mit.edu

Devavrat Shah
EECS, MIT, devavrat@mit.edu

We consider the problem of static assortment optimization, where the goal is to find the assortment of size at most C that maximizes revenues. This is a fundamental decision problem in the area of Operations Management. It has been shown that this problem is provably hard for most of the important families of parametric of choice models, except the multinomial logit (MNL) model. In addition, most of the approximation schemes proposed in the literature are tailored to a specific parametric structure. We deviate from this and propose a general algorithm to find the optimal assortment assuming access to only a subroutine that gives revenue predictions; this means that the algorithm can be applied with any choice model. We prove that when the underlying choice model is the MNL model, our algorithm can find the optimal assortment efficiently.

1. Introduction

This paper deals with the application of choice models to make decisions. There are several important practical applications where the end-goal is to make a decision, and a choice model is a critical component to making that decision. The main application area of our focus is the set of decision problems faced by operations managers. In this context, a central decision problem is the *static assortment optimization* problem in which the goal is to find the optimal assortment: the assortment of products with the maximum revenue subject to a constraint on the size of the assortment. Solving the decision problem requires two components: (a) a subroutine that uses historical sales transaction data to predict the expected revenues from offering each assortment of products, and (b) an optimization algorithm that uses the subroutine to find the optimal assortment. This paper deals with desigining an efficient optimization algorithm.

As one can imagine, the problems of predicting revenues and finding the optimal assortment are important in their own right, and their consideration is motivated by the fact that any improvements to existing solutions will have significant practical implications. Specifically, solutions to these two problems lead to a solution to the *single-leg, multiple fare-class yield management problem*; this problem is central to the area Revenue Management (RM) and deals with the allocation of aircraft seat capacity to multiple fare classes when customers exhibit choice behavior. In particular, consider an airline selling tickets to a single-leg aircraft. Assume that the airline has already decided the fare classes and is trying to dynamically decide which fare-classes to open as a function of the remaining booking time and the remaining number of seats. This dynamic decision problem can be cast in a reasonably straightforward manner as a dynamic program with one state variable. As shown in Talluri and van Ryzin (2004), the solution to the dynamic program reduces to solving a slight variant of the static assortment optimization problem. Thus, solution to the two problems effectively solves the single-leg, multiple fare-class yield management problem âŁ” a central problem to RM with huge practical implications.

Given the subroutine to predict revenues, we need an efficient algorithm to search for the optimal assortment. In particular, we are interested in solving

$$\underset{|\mathcal{M}|\leq C}{\arg\max} \quad R(\mathcal{M}),$$

where $R(\mathcal{M})$ is the expected revenue from offering assortment $\mathcal{M}$. In this chapter, we assume access to a subroutine that can efficiently generate revenue predictions for each assortment $\mathcal{M}$, and our goal is to design an optimization algorithm that minimizes the number of calls to the subroutine. The revenue predictions can themselves be generated either using a specific parametric choice model or using the nonparametric approach described in the previous chapter. Assuming there are N products and a constraint of C on the size of the optimal assortment, exhaustive search would require $O(N^C)$ calls to the revenue subroutine. Such an exhaustive search is prohibitive in practice whenever N or C is large. Therefore, our goal is to propose an algorithm that can produce a "good" approximation to the optimal assortment with only a "few" calls to the revenue subroutine. Existing approaches focus on exploiting specific parametric structures of choice models to solve the decision problem efficiently. In this context, Rusmevichientong et al. (2010a) have proposed an efficient algorithm to find the optimal assortment in $O(NC)$ operations whenever the underlying model is the MNL model. Unfortunately, beyond the simple case of the MNL model, the optimization problem or its variants are provably hard (like the NL and MMNL models; see Rusmevichientong et al. (2009) and Rusmevichientong et al. (2010b)). In addition, the algorithms proposed in the literature (both exact and approximate) heavily exploit the structure of the assumed choice model; consequently, the existing algorithms – even without any guarantees – cannot be used with other choice models like the probit model or the mixture of MNL models with a continuous mixture. Given these issues, our goal is to design a general optimization scheme that is (a) not tailored to specific parametric structures and (b) requires only a subroutine that gives revenue estimates for assortments.

Overview of our approach. We propose a general set-function optimization algorithm, which given a general function defined over sets, finds an estimate of the set (or assortment) where the function is maximized. This set-function optimization algorithm clearly applies to the static assortment optimization problem, thereby yielding the optimization scheme with the desired properties. Note that since we are considering a very general setup, there is not much structure to exploit. Hence, we adopt the greedy method – the general technique for designing heuristics for optimization problems. However, a naive greedy implementation algorithm fails even in the simple case of the MNL model. Specifically, consider the simpler un-capacitated decision problem. Here, a naive greedy implementation would start with the empty set and incrementally build the solution set by adding at each stage a product that results in the maximum increase in revenue; this process would terminate when addition of a product no longer results in an increase in revenue. It is easy to see that the naive implementation would succeed in solving the decision problem only if the optimal assortments exhibit a nesting property: the optimal assortment of size C_1 is a subset of the optimal assortment of size C_2 whenever $C_1 < C_2$. Unfortunately, the nesting property does not hold even in the case of the MNL model. In order to overcome this issue, we allow for greedy "exchanges" in addition to greedy "additions." Particularly, at every stage, we allow a new product to be either added (which we call an "addition") to the solution set or replace an existing product (which we call an "exchange") in the solution set; the operation at each stage is chosen greedily. The termination condition now becomes an interesting question. As in the naive implementation, we could terminate the process when addition or exchange no longer results in an increase in revenue. However, since we never run out of products for exchanges, the algorithm may take an exponential (in the number of products) number of steps to terminate. We overcome this issue by introducing a control parameter that caps the number of times a product may be involved in exchanges. Calling that parameter b, we show that the algorithms calls the revenue subroutine $O(N^2bC^2)$ times for the capacitated problem. We thus obtain a general algorithm with the desired properties to solve the static assortment optimization problem.

Guarantees for our algorithm. We derive guarantees to establish the usefulness of our optimization procedure. For that, we first consider the case of the MNL model, where the decision problem is well-understood. Specifically, we assume that the underlying choice model is an instance of the MNL family and the revenue subroutine yields revenue estimates for assortments under the specific instance. We can show that the the algorithm we propose, when run with $b \geq C$, succeeds in finding the optimal assortment with $O(N^2C^3)$ calls to the revenue subroutine. Therefore, in the special case when the underlying choice model is the MNL model, our algorithm captures what is already known. It also provides a simpler alternative to the more complicated algorithm proposed by Rusmevichientong et al. (2010a). We also consider the case when noise corrupts the available revenue estimates – a common practical issue. In this case, we show that our algorithm is robust to errors in the revenue estimates produced by the subroutine. Particularly, if the underlying choice model is the MNL model and the revenue estimate produced by the subroutine may not be exact but within a factor $1-\varepsilon$ of the true value, then we can show that our algorithm finds an estimate of the optimal assortment with revenue that is within $1 - f(\varepsilon)$ of the optimal value; here $f(\varepsilon)$ goes to zero with ε and also depends on C and the parameters of the underlying model. In summary, our theoretical analysis shows that our algorithm finds the exact optimal solution in the noiseless case or a solution with provable guarantees in the noisy case, whenever the underlying choice model is the MNL model. In this sense, our results subsume what is already known in the context of the MNL model.

In the context of the more complicated models like the nested logit (NL) and the mixtures of MNL models, the decision problem is provably hard. As discussed above, even obtaining a PTAS can be very complicated and requires careful exploitation of the structure. We however believe that it is possible to obtain “good” approximations to the optimal assortments in practice.

Organization. Next, we describe in detail the optimization algorithm we propose and the guarantees we can provide. The rest of the chapter is organized as follows. The optimization algorithm, which we call GreedyOPT is described in Section 2. We then describe the precise guarantees we can provide on the algorithm in Section 3. Finally, we present the proofs of our results in Section 4 before concluding in Section 5.

2. Description of GreedyOPT

We now provide the detailed description of our optimization algorithm GreedyOPT. As noted above, most of the algorithms proposed in the literature – both exact and approximate – are based on heavily exploiting the structure of the assumed choice model. Unfortunately, since we are considering a very general setup, there is not much structure to exploit. Hence, we adopt the greedy method – the general technique for designing heuristics for optimization problems.

A naive greedy implementation however fails even in the simple case of the MNL model. Specifically, consider the simpler un-capacitated decision problem. Here, a naive greedy implementation would start with the empty set and incrementally build the solution set by adding at each stage a product that results in the maximum increase in revenue; this process would terminate when addition of a product no longer results in an increase in revenue. It is easy to see that the naive implementation would succeed in solving the decision problem only if the optimal assortments exhibit a nesting property: the optimal assortment of size C_1 is a subset of the optimal assortment of size C_2 whenever $C_1 < C_2$. Unfortunately, the nesting property does not hold even in the case of the MNL model.

In order to overcome this issues associated with the naive greedy implementation, we allow for greedy “exchanges” in addition to greedy “additions.” Particularly, at every stage, we allow a new product to be either added (which we call an “addition”) to the solution set or replace an existing product (which we call an “exchange”) in the solution set; the operation at each stage is chosen greedily. The termination condition now becomes an interesting question. As in the naive implementation, we could terminate the process when addition or exchange no longer results in an increase in revenue. However, since we never run of products for exchanges, the algorithm may take an exponential (in the number of products) number of steps to terminate. We overcome this issue by introducing a control parameter that caps the number of times a product may be involved in exchanges. Calling that parameter b, we show that the algorithms calls the revenue subroutine $O(N^2bC^2)$ times for the capacitated problem. We thus obtain a general algorithm with the desired properties to solve the static assortment optimization problem.

The formal description of the algorithm is provided in Figures 2 and 2. For convenience, whenever an exchange takes place, we call the product that is removed as the product that is *exchanged-out* and the product that is introduced as the product that is *exchanged-in*. Now, the algorithm takes as inputs the capacity C, the initial assortment size S, and a bound b on the number of exchange-outs. The algorithm incrementally builds the solution assortment. Specifically, it searches over all assortments of size S. For each such assortment, the algorithm calls the subroutine GreedyADD-EXCHANGE (formally described in Figure 2) at most $C - S$ times to construct an assortment of size at most C. Of all such constructed assortments, the algorithm returns the one with the maximum revenue.

Figure 1 GreedyOPT

Input: Initial size S, capacity constraint C such that $1 \leq S \leq C \leq N$, and revenue function $R(\cdot)$.

Output: Estimate of optimal assortment $\hat{M}^{\text{OPT}}$ of size $|\hat{M}^{\text{OPT}}| \leq C$

Algorithm:

Initialization: $\hat{M}^{\text{OPT}} \leftarrow \emptyset$
for each $\mathcal{M} \subset \mathcal{N}$ such that $|\mathcal{M}| = S$
    for $S+1 \leq i \leq C$
        $\mathcal{M} \leftarrow$GreedyADD-EXCHANGE$(\mathcal{M}, \mathcal{N}, b, R(\cdot))$
    end for
    if $R(\hat{M}^{\text{OPT}}) < R(\mathcal{M})$
        $\hat{M}^{\text{OPT}} \leftarrow \mathcal{M}$
    end if
end for
Output: $\hat{M}^{\text{OPT}}$

Running-time complexity: It is easy to see that the number of times GreedyOPT calls the revenue function $R(\cdot)$ is equal to $(C-S)\binom{N}{S}$ times the number of times GreedyADD-EXCHANGE calls the revenue function. In order to count the number of times GreedyADD-EXCHANGE calls the revenue function $R(\cdot)$, we first count the number of times the while loop in GreedyADD-EXCHANGE is executed. The number of times the while loop runs is bounded above by the maximum number of iterations before the set $\tilde{\mathcal{N}}$ becomes empty. In each iteration either an addition or an exchange takes place. Since there is at most one addition that can take place and $|\tilde{\mathcal{N}}|$ decreases by 1 whenever exchange-outs(i) of a product i reaches b, it follows that the while loop runs for at most $Nb+1$ iterations. In each iteration of the while loop, the revenue function is called at most $O(CN)$ times. Thus, GreedyADD-EXCHANGE calls the revenue function at most $O(CbN^2)$ times. Since $\binom{N}{S} = O(N^S)$, we can now conclude that GreedyOPT calls the revenue function $O(C^2bN^{S+2})$. The choice of S will depend on the accuracy of revenue estimates we have access to. Next, we provide guarantees on GreedyOPT, which provide guidance on the choice of S.

Figure 2 GreedyADD-EXCHANGE

Input: assortment $\mathcal{M}$, product universe $\mathcal{N}$, revenue function $R(\cdot)$, maximum number of exhange-outs b

Output: Estimate of optimal assortment of size at most $|\mathcal{M}|+1$

Algorithm:
Initialization: $\hat{\mathcal{M}} \leftarrow \mathcal{M}$, $\tilde{\mathcal{N}} \leftarrow \mathcal{N}$, exchange-outs$(i)=0$ for each $i \in \mathcal{N}$
while $\tilde{\mathcal{N}} \neq \emptyset$
  //try exchanging products
  $i^*, j^* = \arg\max_{i \in \hat{\mathcal{M}}, j \in \tilde{\mathcal{N}}} R\left((\hat{\mathcal{M}} \setminus \{i\}) \cup \{j\}\right)$
  $\tilde{\mathcal{M}}_{\text{exchange}} \leftarrow (\hat{\mathcal{M}} \setminus \{i\}) \cup \{j\}$

  // try adding a product
  $k^* = \arg\max_{k \in \tilde{\mathcal{N}}} R(\hat{\mathcal{M}} \cup \{k\})$
  $\tilde{\mathcal{M}}_{\text{add}} \leftarrow \hat{\mathcal{M}} \cup \{k^*\}$

  if $|\hat{\mathcal{M}}| < |\mathcal{M}|+1$ **and** $R(\tilde{\mathcal{M}}_{\text{add}} > R(\mathcal{M})$ **and** $R(\tilde{\mathcal{M}}_{\text{add}}) > R(\tilde{\mathcal{M}}_{\text{exchange}})$
    // add the product k^*
    $\hat{\mathcal{M}} \leftarrow \tilde{\mathcal{M}}_{\text{add}}$
    $\tilde{\mathcal{N}} \leftarrow \tilde{\mathcal{N}} \setminus \{k^*\}$
  else if $R(\tilde{\mathcal{M}}_{\text{exchange}}) > R(\mathcal{M})$
    // exchange products i^* and j^*
    $\hat{\mathcal{M}} \leftarrow \tilde{\mathcal{M}}_{\text{exchange}}$
    exchange-outs$(i^*) \leftarrow$ exchange-outs$(i^*)+1$
    if exchange-outs$(i) \geq b$
      $\tilde{\mathcal{N}} \leftarrow \tilde{\mathcal{N}} \setminus \{j^*\}$
    else
      $\tilde{\mathcal{N}} \leftarrow \left(\tilde{\mathcal{N}} \setminus \{j^*\}\right) \cup \{i^*\}$
  else
    break from while
  end if
end while
Output: $\hat{\mathcal{M}}$

3. Theoretical guarantees for GreedyOPT

We now give a precise description of the main results we can establish for the GreedyOPT algorithm. Specifically, suppose that the underlying choice model is an MNL model with weights $w_0 = 1$ for product 0 and w_i for product $i \in \mathcal{N}$; recall that the choice probabilities are given by

$$\mathbb{P}(i|\mathcal{M}) = \frac{w_i}{1+\sum_{j \in \mathcal{M}} w_j}.$$

Note that 1 appears in the denominator because of the no-purchase option. In particular, the probability that an arriving customer leaves without purchasing anything when assortment $\mathcal{M}$ is on offer is given by

$$\mathbb{P}(0|\mathcal{M}) = \frac{1}{1+\sum_{i \in \mathcal{M}} w_i}.$$

Let $R(\mathcal{M})$ denote the expected revenue from assortment $\mathcal{M}$. Under the MNL model, we have

$$R(\mathcal{M}) = \frac{\sum_{i\in\mathcal{M}} p_i w_i}{1+\sum_{i\in\mathcal{M}} w_i},$$

where p_i is the price or the revenue obtained from the sale of product i.

We now have the following theorem when the revenue subroutine provides exact revenues:

Theorem 1. *Suppose the underlying model is the MNL model with weights $w_1, w_2, \ldots, w_N$ and the revenue subroutine provides exact revenues. Then, for any $S \geq 0$ and $b \geq C+1$, the* GreedyOPT *algorithm finds the optimal solution to* Capacitated OPT *problem.*

Therefore, taking $S=0$ and $b=C+1$, GreedyOPT finds the optimal assortment of size at most C by calling the revenue function $O(N^2C^3)$. Thus, our algorithm provides a simpler alternative to the more complicated algorithm proposed by Rusmevichientong et al. (2010a).

We next show that the GreedyOPT algorithm is robust to errors in the available revenue estimates. Specifically, we consider the more realistic setting where one has access to only approximate estimates of revenues i.e., we assume access to a function $\tilde{R}(\cdot)$ such that for any assortment $\mathcal{M}$ we have

$$(1-\varepsilon(\mathcal{M}))R(\mathcal{M}) \leq \tilde{R}(\mathcal{M}) \leq R(\mathcal{M})$$

for some parameter $0 < \varepsilon(\mathcal{M}) < 1$. Naturally, the parameter $\varepsilon(\mathcal{M})$ determines the quality of revenue estimates we have available. Assuming that we have access to only approximate revenues, we find the optimal assortment by running GreedyOPT with approximate revenues. In order to describe the result, we need some notation. For any assortment $\mathcal{M}$, let $w(\mathcal{M})$ denote $1+\sum_{i\in\mathcal{M}} w_i$. Further, let

$$\varepsilon_{\max} \stackrel{\text{def}}{=} \max_{\mathcal{M}:\, |\mathcal{M}|\leq C} \varepsilon(\mathcal{M}) \quad \text{and} \quad W_C^{\max} \stackrel{\text{def}}{=} \max_{\mathcal{M}:\, |\mathcal{M}|\leq C} w(\mathcal{M}).$$

Finally, we defer to the next section the precise definitions of two quantities $\bar{C}(\delta_C)$ and δ_C that we need to describe the theorem; it suffices to say that as $\varepsilon_{\max} \to 0$, we have $\delta_C \to 0$ and $\bar{C}(\delta_C) \to C$.

With these definitions, we can now state our result.

Theorem 2. *Let M_C^{OPT} denote the optimal assortment of size at most C and $\hat{M}_C^{\text{OPT}}$ denote the estimate of the optimal assortment produced by* GreedyOPT *when run with inputs $S \geq 0$ and $b \geq \bar{C}(2\delta_C)+1$. Then, we must have*

$$\frac{R(M_C^{\text{OPT}}) - R(\hat{M}_C^{\text{OPT}})}{R(M_C^{\text{OPT}})} \leq f(w, \varepsilon_{\max}),$$

where w denotes the vector of weights $(w_1, w_2, \ldots, w_N)$ and

$$f(w, \varepsilon_{\max}) \stackrel{\text{def}}{=} \frac{W_C^{\max}}{w(M_C^{\text{OPT}})} \eta(\varepsilon_{\max})$$

with $\eta(\varepsilon_{\max}) \stackrel{\text{def}}{=} 4C\varepsilon_{\max}/(1-\varepsilon_{\max})$.

It is easy to see that the algorithm calls the revenue function $O(N^2C^2\bar{C}(2\delta_C))$ times. Note that as $\varepsilon_{\max} \to 0$, $\eta(\varepsilon_{\max})$ and hence $f(w, \varepsilon_{\max})$ go to zero. In addition, it follows from our definitions that as $\varepsilon_{\max} \to 0$, $\bar{C}(2\delta_C) \to C$. Consequently, taking the error in revenues $\varepsilon_{\max} = 0$ yields in Theorem 2 yields the result of Theorem 1 as the special result. Therefore, we only prove Theorem 2 in the next section.

4. Proofs of the main results

In this section we prove Theorem 2; specifically, we establish that the revenues of the optimal assortment and the estimate of the optimal assortment produced by GreedyOPT are "close". In order to establish this result, for the rest of the section, fix a capacity C. Let M^{OPT} and $\hat{M}^{\text{OPT}}$ respectively denote the optimal assortment and the estimate of the optimal assortment produced by Greedy-OPT. Then, our goal is to show that $R(M^{\text{OPT}})$ and $R(\hat{M}^{\text{OPT}})$ are "close" to each other. We assume that the underlying choice model is the MNL model with parameters $w_1, w_2, \ldots, w_N$. Recall that for any assortment $\mathcal{M}$,

$$R(\mathcal{M}) = \frac{\sum\limits_{i\in\mathcal{M}} p_i w_i}{1 + \sum\limits_{i\in\mathcal{M}} w_i},$$

where p_i is the price of product i. The term in the denominator makes comparison of the revenues of two different assortment difficult. Therefore, instead of dealing with the revenues of the assortment directly, we consider the following transformation of the revenues of assortments: for any assortment $\mathcal{M}$ and number $u \in \mathbb{R}$,

$$\begin{aligned} R(\mathcal{M}) - u = \frac{\sum\limits_{i\in\mathcal{M}} p_i w_i}{1 + \sum\limits_{i\in\mathcal{M}} w_i} - u &= \frac{\left(\sum\limits_{i\in\mathcal{M}} (p_i - u) w_i\right) - u}{1 + \sum\limits_{i\in\mathcal{M}} w_i} \\ &= \frac{H_{\mathcal{M}}(u) - u}{w(\mathcal{M})}, \end{aligned}$$

where $H_{\mathcal{M}} \colon \mathbb{R} \to \mathbb{R}$ is a function defined as $H_{\mathcal{M}}(u) = \sum_{i\in\mathcal{M}} (p_i - u) w_i$ and $w(\mathcal{M}) \stackrel{\text{def}}{=} 1 + \sum_{i\in\mathcal{M}} w_i$. We can now write

$$H_{\mathcal{M}}(u) = u + w(\mathcal{M})(R(\mathcal{M}) - u). \tag{1}$$

It is clear that $H_{\mathcal{M}}(\cdot)$ is directly related to the revenue $R(\mathcal{M})$. Moreover, as will become apparent soon, it is easier to compare the transformations $H_{\mathcal{M}_1}(\cdot)$ and $H_{\mathcal{M}_2}(\cdot)$ of two assortments $\mathcal{M}_1$ and $\mathcal{M}_2$ than their revenues $R(\mathcal{M}_1)$ and $R(\mathcal{M}_2)$. Specifically, we can establish the properties stated in the following proposition.

Proposition 1. *For any two assortments $\mathcal{M}_1$ and $\mathcal{M}_2$, let $H_1(\cdot)$ and $H_2(\cdot)$ respectively denote the functions $H_{\mathcal{M}_1}(\cdot)$ and $H_{\mathcal{M}_2}(\cdot)$. Further, let u_1 and u_2 denote the revenues $R(\mathcal{M}_1)$ and $R(\mathcal{M}_2)$ respectively. We then have*
1. $H_1(u_2) \geq H_2(u_2) \iff R(\mathcal{M}_1) \geq R(\mathcal{M}_2)$.
2. $H_1(u_2) \geq (1 + \delta(\mathcal{M}_1)) H_2(u_2) \implies \tilde{R}(\mathcal{M}_1) \geq \tilde{R}(\mathcal{M}_2)$,

where $\delta(\mathcal{M}_1) \stackrel{\text{def}}{=} \varepsilon(\mathcal{M}_1) w(\mathcal{M}_1)/(1 - \varepsilon(\mathcal{M}_1))$.

W e prove each of the properties in turn. First note that for any assortment $\mathcal{M}$ with revenue $R(\mathcal{M}) = u$, it immediately follows from our definitions that $H_{\mathcal{M}}(u) = u + w(\mathcal{M})(R(\mathcal{M}) - u) = u$. The first property now follows from a straightforward expansion of the terms involved:

$$\begin{aligned} H_1(u_2) \geq H_2(u_2) &\iff u_2 + w(\mathcal{M}_1)(u_1 - u_2) \geq u_2 \\ &\iff u_1 \geq u_2 \\ &\iff R(\mathcal{M}_1) \geq R(\mathcal{M}_2), \end{aligned}$$

where the second equivalence follows from the fact that $w(\mathcal{M}_1) > 0$. The second property can also be obtained through a similar straightforward expansion of the terms. In particular,

$$H_1(u_2) \geq (1 + \varepsilon(\mathcal{M}_1)) H_2(u_2) \iff u_2 + w(\mathcal{M}_1)(u_1 - u_2) \geq (1 + \delta(\mathcal{M}_1)) u_2$$

$$\begin{aligned}
&\iff u_1 \geq \left(1+\frac{\delta(\mathcal{M}_1)}{w(\mathcal{M}_1)}\right)u_2\\
&\iff u_1 \geq \left(1+\frac{\varepsilon(\mathcal{M}_1)}{1-\varepsilon(\mathcal{M}_1)}\right)u_2\\
&\iff (1-\varepsilon(\mathcal{M}_1))u_1 \geq u_2, \qquad (2)
\end{aligned}$$

where the second equivalence follows from the definition of $\delta(\mathcal{M}_1)$. Moreover, it follows from our definitions that $\tilde{R}(\mathcal{M}_1) \geq (1-\varepsilon(\mathcal{M}_1))u_1$ and $u_2 \geq \tilde{R}(\mathcal{M}_2)$. We now conclude from (2) that

$$\tilde{R}(\mathcal{M}_1) \geq (1-\varepsilon(\mathcal{M}_1))u_1 \geq u_2 \geq \tilde{R}(\mathcal{M}_2).$$

The result of the proposition now follows.

The above proposition establishes that if the transformation $H_{\mathcal{M}}(\cdot)$ of one assortment is "sufficiently" larger than the other, then it follows that the revenues of one assortment should be larger than the revenues of the other. Therefore, instead of keeping track of the revenues of the assortments in our algorithm, we keep track of their respective transformations $H_{\mathcal{M}}(\cdot)$.

Next, we establish a loop-invariance property that arises due to greedy additions and exchanges in our algorithms. We make use of this property to prove our theorems. In order to state the proposition, we introduce the following notation:

$$\delta_C \stackrel{\text{def}}{=} \max_{\mathcal{M}:\,|\mathcal{M}|} \delta(\mathcal{M}) = \max_{\mathcal{M}:\,|\mathcal{M}|} w(\mathcal{M})\frac{\varepsilon(\mathcal{M})}{1-\varepsilon(\mathcal{M})}.$$

We then have

PROPOSITION 2. *Consider an iteration t of the while loop of the* GREEDYADD-EXCHANGE *algorithm. Let $\mathcal{M}_t$ and $\mathcal{M}_{t+1}$ denote the estimates of the optimal assortments at the beginning and the end of iteration t. Let $\mathcal{N}_t$ denote the universe of products at the beginning of iteration t. Then,*

1. *if a greedy exchange takes place i.e., $\mathcal{M}_{t+1} = (\mathcal{M}_t \setminus \{i^*\}) \cup \{j^*\}$, then for $u = R(\mathcal{M}_{t+1})$, we must have*

$$\begin{aligned}
&h_{i^*}(u) \leq h_i(u) + \delta_C u, && \textit{for all } i \in \mathcal{M}_t\\
&h_{j^*}(u) \geq h_j(u) - \delta_C u, && \textit{for all } j \in \mathcal{N}_t \setminus \mathcal{M}_t;
\end{aligned}$$

2. *if an addition takes place i.e., $\mathcal{M}_{t+1} = \mathcal{M}_t \cup \{j^*\}$, then for $u = R(\mathcal{M}_{t+1})$ we must have*

$$h_{j^*}(u) \geq h_j(u) - \delta_C u, \quad \textit{for all } j \in \mathcal{N}_t \setminus \mathcal{M}_t.$$

W e prove this proposition by contradiction. First consider the case when exchange happens i.e., $\mathcal{M}_{t+1} = (\mathcal{M}_t \setminus \{i^*\}) \cup \{j^*\}$. Note that for any assortment $\mathcal{M} = (\mathcal{M}_t \setminus \{i\}) \cup \{j\}$ with $i \in \mathcal{M}_t$ and $j \in \mathcal{N}_t \setminus \mathcal{M}_t$, letting u denote $R(\mathcal{M}_{t+1})$, we can write

$$H_{\mathcal{M}}(u) - H_{\mathcal{M}_{t+1}}(u) = h_j(u) - h_{j^*}(u) + h_{i^*}(u) - h_i(u). \qquad (3)$$

Now, if the hypothesis of the proposition pertaining to exchange is false, then at least one of the following should be true: either (1) there exists a product $i \in \mathcal{M}_t$ and $i \neq i^*$ such that $h_{i^*}(u) > h_i(u) + \delta_C u$, or (2) there exists a product $j \in \mathcal{N}_t \setminus \mathcal{M}_t$ and $j \neq j^*$ such that $h_{j^*}(u) < h_j(u) + \delta_C u$. In the first case when $h_{i^*}(u) > h_i(u) + \delta_C u$, by taking $j = j^*$, we can write from (3) that $H_{\mathcal{M}}(u) - H_{\mathcal{M}_{t+1}}(u) > \delta_C u$. Similarly, in the second case when $h_{j^*}(u) < h_j(u) + \delta_C u$, by taking $i = i^*$, we can write from (3) that $H_{\mathcal{M}}(u) - H_{\mathcal{M}_{t+1}}(u) > \delta_C u$. Therefore, in both the cases, we have exhibited an assortment $\mathcal{M}$

distinct from $\mathcal{M}_{t+1}$ that can be obtained from $\mathcal{M}_t$ through an exchange and has the property that $H_{\mathcal{M}}(u) - H_{\mathcal{M}_{t+1}}(u) > \delta_C u$. We can now write

$$
\begin{aligned}
& && H_{\mathcal{M}}(u) > H_{\mathcal{M}_{t+1}}(u) + \delta_C u && && \text{(4a)}\\
&\Longrightarrow && H_{\mathcal{M}}(u) > H_{\mathcal{M}_{t+1}}(u) + \delta(\mathcal{M}) u && \text{since } \delta_C \geq \delta(\mathcal{M}) \text{ by definition} && \text{(4b)}\\
&\Longrightarrow && H_{\mathcal{M}}(u) > (1+\delta(\mathcal{M})) H_{\mathcal{M}_{t+1}}(u) && \text{since } H_{\mathcal{M}_{t+1}}(u) = u \text{ by definition} && \text{(4c)}\\
&\Longrightarrow && \tilde{R}(\mathcal{M}) > \tilde{R}(\mathcal{M}_{t+1}) && \text{by Proposition 1.} && \text{(4d)}
\end{aligned}
$$

This clearly contradicts the fact that $\mathcal{M}_{t+1}$ is chosen greedily.

The case when addition happens can be proved in the exact similar way. Particularly, suppose there exists a product $j \in \mathcal{N}_t \setminus \mathcal{M}_t$ and $j \neq j^*$ such that $h_{j^*}(u) < h_j(u) - \delta_C u$, where $u = R(\mathcal{M}_{t+1})$ with $\mathcal{M}_{t+1} = \mathcal{M}_t \cup \{j\}$. Letting $\mathcal{M}$ denote the set $\mathcal{M}_t \cup \{j\}$, we can then write

$$H_{\mathcal{M}}(u) - H_{\mathcal{M}_{t+1}}(u) = h_j(u) - h_{j^*}(u) > \delta_C u.$$

This implies – following the sequence of arguments in (4) – that $\tilde{R}(\mathcal{M}) > \tilde{R}(\mathcal{M}_t)$, contradicting the fact that $\mathcal{M}_{t+1}$ is chosen greedily.

The result of the proposition now follows.

The above proposition establishes a key loop-invariance property that results from greedy additions and exchanges. Specifically, let u denote the revenue of the estimate of the optimal assortment obtained at the end of an iteration of the while loop in GreedyADD-EXCHANGE. Then, the proposition establishes that whenever a product j^* is introduced (either through addition or an exchange-in) greedily, it must be that $h_{j^*}(u)$ is "close" to the maximum $h_j(u)$ of all products j that have been considered for an addition or exchange-in. Similarly, the product i^* that is greedily exchanged-out must be such that $h_{i^*}(u)$ is "close" to the minimum $h_i(u)$ of all products i that have been considered for an exchange-out.

Using the propositions above, we can establish a key property of the subroutine GreedyADD-EXCHANGE. For that, we need the following notation. For any u, define

$$B_S(u) \overset{\text{def}}{=} \arg\max_{\mathcal{M}\colon |\mathcal{M}| \leq S} H_{\mathcal{M}}(u) = \arg\max_{\mathcal{M}\colon |\mathcal{M}| \leq S} \sum_{i \in \mathcal{M}} h_i(u).$$

It is easy to see from the above definition that $B_S(u)$ consists of the top at most C products according to $h_i(u)$ such that $h_i(u) > 0$. Since $h_i(\cdot)$ is monotonically decreasing, it is easy to see that

$$|B_S(u_1)| \geq |B_S(u_2)|, \quad \text{whenever } u_1 \leq u_2. \tag{5}$$

Under appropriate technical assumptions, Rusmevichientong et al. (2010a) showed that for any $1 \leq S \leq N$, the optimal assortment of size at most S under the MNL model is one of the assortments in the collection $\mathcal{B}_S \overset{\text{def}}{=} \{B_S(u)\colon u \in \mathbb{R}\}$. In fact the authors show that if u_S denotes the optimal revenue, then $B_S(u_S)$ is the optimal assortment. An immediate consequence of this result and (5) is that for any $u \leq u_S$

$$S \geq |B_S(u)| \geq |M_S^{\text{OPT}}|. \tag{6}$$

It has been established by Rusmevichientong et al. (2010a) that there can be at most $O(NC)$ distinct assortments in the collection $\mathcal{B}_S$ allowing one to find the optimal assortment by restricting one's search to $O(NC)$ assortments. The following lemma shows that the assortment found by the subroutine GreedyADD-EXCHANGE is "close" to one of the assortments in $\mathcal{B}_S$. Before we describe the lemma, we need the following notation. For any $\delta > 0$ and $u \in \mathbb{R}$, let

$$i_S(u) \overset{\text{def}}{=} \min_{i \in B_S(u)} h_i(u).$$

Moreover, let

$$\bar{B}_S(\delta,u) \stackrel{\text{def}}{=} B_S(u) \cup \{j \in \mathcal{N} \setminus B_S(u) \colon h_{i_S(u)}(u) - h_j(u) \le \delta u\},$$

Also, let

$$\bar{C}(\delta) \stackrel{\text{def}}{=} \max_{u \in \mathbb{R}_+} |\bar{B}_S(\delta,u)|,$$

We then have

Lemma 1. *Suppose* GreedyADD-EXCHANGE *is run with some input assortment $\mathcal{M}$ and $b \ge \bar{C}(\delta_C)+2$, where $C \ge S+1$. Further, suppose that $|M^{\mathrm{OPT}}_{S+1}| = S+1$. Then, there exists an iteration t^* of the while loop such that if $\mathcal{M}^*$ denotes the assortment $\mathcal{M}_{t^*+1}$ and u^* denotes $R(\mathcal{M}^*)$, then*

$$H_{B(u^*)}(u^*) - H_{\mathcal{M}^*}(u^*) \le 2\tilde{C}_{u^*}\delta_C u^*,$$

where $B(u^*)$ denotes the assortment $B_{S+1}(u^*)$ and $\tilde{C}^*$ is a constant denoting $1 + |B(u^*) \setminus \mathcal{M}^*|$.

We defer the proof of Lemma 1 to the end of the section. We now present the proof of Theorem 2.

4.1. Proof of Theorem 2

Let M^{OPT}_C denote the true optimal assortment, and $\hat{M}^{\mathrm{OPT}}_C$ denote the estimate of the optimal assortment produced by GreedyOPT. Furthermore, let $C^* \le C$ denote the size of M^{OPT}_C. It follows from Lemma 1 that in the C^*th invocation of the subroutine GreedyADD-EXCHANGE, there exists an assortment $\mathcal{M}^*$ such that $\tilde{R}(\hat{M}^{\mathrm{OPT}}_C) > \tilde{R}(\mathcal{M}^*)$ and $\mathcal{M}^*$ is such that

$$H_{B(u^*)}(u^*) - H_{\mathcal{M}^*}(u^*) \le 2\tilde{C}_{u^*}\delta_C u^*,$$

where $\tilde{C}^*$ denotes $|B(u^*) \setminus \mathcal{M}^*| + 1$ and $B(u^*)$ denotes the set $B_{C^*}(u^*)$. It follows by the definition of $B(u^*)$ that $H_{B(u^*)}(u^*) \ge H_{M^{\mathrm{OPT}}_C}(u^*)$. Thus, we can write

$$H_{M^{\mathrm{OPT}}_C}(u^*) - H_{\mathcal{M}^*}(u^*) \le 2\tilde{C}_{u^*}\delta_C u^* \le 2C\delta_C u^*. \tag{7}$$

Let u_C denote $R(M^{\mathrm{OPT}}_C)$. Then, it follows by definition that $H_{M^{\mathrm{OPT}}_C}(u_C) = u_C$. Thus,

$$H_{M^{\mathrm{OPT}}_C}(u_C) - H_{M^{\mathrm{OPT}}_C}(u^*) = \sum_{j \in M^{\mathrm{OPT}}_C} w_j(u^* - u_C) = (u^* - u_C)(w(M^{\mathrm{OPT}}_C) - 1).$$

Since $H_{M^{\mathrm{OPT}}_C}(u_C) = u_C$, we can write

$$H_{M^{\mathrm{OPT}}_C}(u^*) = u_C + (u_C - u^*)(w(M^{\mathrm{OPT}}_C) - 1). \tag{8}$$

Since $H_{\mathcal{M}^*}(u^*) = u^*$, it now follows from (7) and (8) that

$$\begin{aligned}
& (u_C - u^*)(w(M^{\mathrm{OPT}}_C) - 1) + u_C - u^* \le 2C\delta_C u^* \\
\implies \quad & (u_C - u^*)w(M^{\mathrm{OPT}}_C) \le 2C\delta_C u^* \\
\implies \quad & u_C \le (1+\tilde{\varepsilon})u^*,
\end{aligned} \tag{9}$$

where $\tilde{\varepsilon} \stackrel{\text{def}}{=} 2C\delta_C / w(M^{\mathrm{OPT}}_C)$. Now since $\tilde{R}(\hat{M}^{\mathrm{OPT}}_C) > \tilde{R}(\mathcal{M}^*)$, it follows that

$$(1 - \varepsilon(\mathcal{M}^*))u^* \le \tilde{R}(\mathcal{M}^*) < \tilde{R}(\hat{M}^{\mathrm{OPT}}) \le \hat{u}_C,$$

where $\hat{u}_C$ denotes $R(\hat{M}^{\mathrm{OPT}}_C)$. It now follows from (9) that

$$u_C \le (1+\tilde{\varepsilon})u^* \le \frac{1+\tilde{\varepsilon}}{1-\varepsilon(\mathcal{M}^*)}\hat{u}_C.$$

Now since

$$\delta_C = \max_{\mathcal{M}:\,|\mathcal{M}|\le C} \frac{\varepsilon(\mathcal{M})}{1-\varepsilon(\mathcal{M})} w(\mathcal{M}),$$

by letting $\varepsilon_{\max} = \max_{\mathcal{M}:\,|\mathcal{M}|\le C} \varepsilon(\mathcal{M})$ and $W_C^{\max} = \max_{\mathcal{M}:\,|\mathcal{M}|\le C} w(\mathcal{M})$, we have

$$\delta_C \le \frac{\varepsilon_{\max}}{1-\varepsilon_{\max}} W_C^{\max}.$$

Thus,

$$\tilde{\varepsilon} = \frac{2C}{w(M_C^{\mathrm{OPT}})}\delta_C \le \frac{2C}{w(M_C^{\mathrm{OPT}})}\frac{\varepsilon_{\max}}{1-\varepsilon_{\max}} W_C^{\max} \stackrel{\mathrm{def}}{=} f(w,\varepsilon_{\max})/2.$$

With these definitions, it is easy to see that $\varepsilon(\mathcal{M}^*) \le \varepsilon_{\max} \le f(w,\varepsilon_{\max})/2$. It now follows that

$$\frac{u_C - \hat{u}_C}{u_C} \le 1 - \frac{1-\varepsilon(\mathcal{M}^*)}{1+\tilde{\varepsilon}} \le \frac{\tilde{\varepsilon}+\varepsilon(\mathcal{M}^*)}{1+\tilde{\varepsilon}} \le \varepsilon(\mathcal{M}^*) + \tilde{\varepsilon} \le f(w,\varepsilon_{\max}).$$

This establishes the result of the theorem.

4.2. Proof of Lemma 1

Suppose the while loop in the subroutine terminates at the end of iteration T. Then, it follows from the description of the subroutine that at least one of the following conditions holds at the end of iteration T:

1. The set of products $\mathcal{N}_{T+1} \setminus \mathcal{M}_{T+1}$ available for additions or exchanges is empty.
2. No further additions or exchanges can increase the revenues.

Our goal is to prove the existence of an iteration $t^* \le T$ such that

$$H_{B(u^*)}(u^*) - H_{\mathcal{M}^*}(u^*) \le 2\tilde{C}_{u^*}\delta_C u^*,$$

where $\mathcal{M}^*$ denotes the assortment $\mathcal{M}_{t^*+1}$ and u^* denotes $R(\mathcal{M}^*)$. We prove this by considering two cases corresponding to each of the two ways in which the subroutine terminates. Note that in order to simplify the notation, we have dropped the subscript from the notation of $B_{S+1}(\cdot)$.

Case 1: Subroutine terminates with $\mathcal{N}_{T+1} = \mathcal{M}_{T+1}$. We first consider the case when the subroutine terminates when the set of products $\mathcal{N}_{T+1} \setminus \mathcal{M}_{T+1}$ becomes empty. In this case, we prove the existence of an iteration $t^* \le T$ that satisfies the condition stated in the hypothesis of the lemma. In fact, we prove something stronger; we shall show that the iteration $t^* \le T^*$, where $T^* \le T$ is the first iteration such that $\mathcal{N}_{T^*} \subset \mathcal{N}$ (recall that $\mathcal{N}_1 = \mathcal{N}$). We prove this result by contradiction. In particular, suppose that after every iteration $t \le T^*$ of the while loop, we have

$$H_{B(u)}(u) - H_{\mathcal{M}_{t+1}}(u) > 2\tilde{C}_u\delta_C u, \tag{10}$$

where u denotes the revenue $R(\mathcal{M}_{t+1})$ and $\tilde{C}_u$ denotes the constant $1 + |B(u) \setminus \mathcal{M}_{t+1}|$. Note that a product i would be removed from the universe $\mathcal{N}_t$ at the end of some iteration t only if it has been exchanged-out b times. Since $b \ge \bar{C}(\delta_C)$, it is easy to see that we arrive at a contradiction if we show that as long (10) is satisfied at the end of each iteration, each product i can be exchanged-out at most $\bar{C}(\delta_C) + 2$ times.

In order to bound the number of times a product can be exchanged-out, we establish a special property that should be satisfied whenever an exchange happens. Specifically, suppose an exchange happens during iteration t i.e., $\mathcal{M}_{t+1} = (\mathcal{M}_t \setminus \{i^*\}) \cup \{j^*\}$. In addition, let u denote the revenue $R(\mathcal{M}_{t+1})$, and let product $k^* \in \mathcal{N}_t \setminus \mathcal{M}_t$ denote the product such that $h_{k^*}(u) \ge h_k(u)$ for all products $k \in \mathcal{N}_t \setminus \mathcal{M}_t$. Then, we claim that

$$h_{j^*}(u) \ge h_{k^*}(u) - \delta_C u \tag{11a}$$

$$h_{i^*}(u) \le h_{k^*}(u) - \delta_C u. \tag{11b}$$

We prove this claim as follows. Since $k^* \in \mathcal{N}_t \setminus \mathcal{M}_t$, (11a) follows directly from Proposition 2. We now argue that $h_{i^*}(u) \leq h_{k^*}(u) - \delta_C u$. For that, we first note that

$$h_{i^*}(u) - h_{j^*}(u) \leq 2\delta_C u. \tag{12}$$

To see why, note that since an exchange has happened, it must be that $\tilde{R}(\mathcal{M}_t) \leq \tilde{R}(\mathcal{M}_{t+1})$. This implies by Proposition 1 that $H_{\mathcal{M}_t}(u) \leq (1+\delta(\mathcal{M}_t))H_{\mathcal{M}_{t+1}}(u)$. Since $\delta(\mathcal{M}_1) \leq \delta_C$ and $H_{\mathcal{M}_{t+1}}(u) = u$ by definition, we can write

$$\begin{aligned} H_{\mathcal{M}_t}(u) \leq (1+\delta(\mathcal{M}_t))H_{\mathcal{M}_{t+1}}(u) &\implies H_{\mathcal{M}_t}(u) - H_{\mathcal{M}_{t+1}}(u) \leq \delta_C u \\ &\implies h_{i^*}(u) - h_{j^*}(u) \leq \delta_C u < 2\delta_C u. \end{aligned}$$

Now, consider

$$\begin{aligned} H_{B(u)}(u) - H_{\mathcal{M}_{t+1}}(u) &= H_{B(u)}(u) - H_{\mathcal{M}_t}(u) + H_{\mathcal{M}_t}(u) - H_{\mathcal{M}_{t+1}}(u) \\ &= \sum_{j \in B(u)\setminus \mathcal{M}_t} h_j(u) - \sum_{i \in \mathcal{M}_t \setminus B(u)} h_i(u) + (h_{i^*}(u) - h_{j^*}(u)). \end{aligned}$$

We now collect terms in the above expression as follows. Let $\mathcal{M}_1$ denote the set $\mathcal{M}_t \setminus B(u)$. Further, partition the set $B(u) \setminus \mathcal{M}_t$ into $Mscr_2 \cup \mathcal{M}_3$ such that $\mathcal{M}_2 \cap \mathcal{M}_3 = \emptyset$ and $|\mathcal{M}_2| = |\mathcal{M}_1|$; note that such a partitioning is possible because $|B(u)| = S+1$ (which follows from (6) and the hypothesis that $|M^{\text{OPT}}_{S+1} = S+1|$) and $|\mathcal{M}_t| \leq S+1$. Also note that $\mathcal{M}_3 \neq \emptyset$ if and only if $|\mathcal{M}_t| < S+1$. With this partitioning, we can now write

$$H_{B(u)}(u) - H_{\mathcal{M}_{t+1}}(u) = \sum_{i \in \mathcal{M}_1, j \in \mathcal{M}_2} (h_j(u) - h_i(u)) + \sum_{j \in \mathcal{M}_3} h_j(u) + (h_{i^*}(u) - h_{j^*}(u)).$$

We now claim that at least on of the following must be true: either (1) there exists a pair of products $i \in \mathcal{M}_1$ and $j \in \mathcal{M}_2$ such that $h_j(u) - h_i(u) > 2\delta_C u$, or (2) if $\mathcal{M}_3 \neq \emptyset$, then there exists a product $k \in \mathcal{M}_3$ such that $h_3(u) > 2\delta_C u$. Otherwise, it is easy to see from (12) that $H_{B(u)}(u) - H_{\mathcal{M}_{t+1}}(u) \leq 2\tilde{C}_u \delta_C u$, where $\tilde{C}_u = |B(u) \setminus \mathcal{M}_{t+1}| + 1$, contradicting (10). We now consider each of the cases in turn.

First suppose that $h_j(u) - h_i(u) > 2\delta_C u$ for some $i \in \mathcal{M}_1$ and $j \in \mathcal{M}_2$. It follows from Proposition 2 that $h_{i^*}(u) \leq h_i(u) + \delta_C u$. Thus, we can write

$$h_{i^*}(u) \leq h_i(u) + \delta_C u < h_j(u) - 2\delta_C u + \delta_C u \leq h_{k^*}(u) - \delta_C u,$$

where the last inequality follows from the definition of k^* and the fact that $j \in \mathcal{M}_2 \subset \mathcal{N} \setminus \mathcal{M}_t$. Thus, for this case, we have established (11b).

Now suppose that $\mathcal{M}_3 \neq \emptyset$ and $h_k(u) > 2\delta_C u$ for some $k \in \mathcal{M}_3$. As noted above, in this case, we should have $|\mathcal{M}_{t+1}| < S+1$. This means that an exchange has happened instead of addition, which in turn implies that $\tilde{R}(\tilde{\mathcal{M}}) \leq \tilde{R}(\mathcal{M}_{t+1})$, where $\tilde{\mathcal{M}}$ denotes the set $\mathcal{M}_t \cup \{k\}$. Thus, by Proposition 1, we should have

$$\begin{aligned} & & H_{\tilde{\mathcal{M}}}(u) &\leq (1+\delta(\tilde{\mathcal{M}}))H_{\mathcal{M}_{t+1}}(u) & & \\ &\implies & H_{\tilde{\mathcal{M}}}(u) - H_{\mathcal{M}_{t+1}}(u) &\leq \delta(\tilde{\mathcal{M}})H_{\mathcal{M}_{t+1}}(u) & & \\ &\implies & h_k(u) + h_{i^*}(u) - h_{j^*}(u) &\leq \delta_C u & &\text{as } H_{\mathcal{M}_{t+1}}(u) = u, \delta(\tilde{\mathcal{M}}) \leq \delta_C \\ &\implies & h_{i^*}(u) &\leq h_{j^*}(u) - h_k(u) + \delta_C u & & \\ &\implies & h_{i^*}(u) &\leq h_{j^*}(u) - 2\delta_C u + \delta_C u & &\text{since } h_k(u) > 2\delta_C u \\ &\implies & h_{i^*}(u) &\leq h_{k^*}(u) - \delta_C u & &\text{since } h_{j^*}(u) \leq h_{k^*}(u). \end{aligned}$$

We have thus established that $h_{i^*}(u) \leq h_{k^*}(u) - \delta_C u$ for both the cases.

We now use (11) to bound the number of exchange-outs that can happen for each product. Specifically, as mentioned above, we arrive at a contradiction by showing that each product can be exchanged-out at most $\bar{C}(\delta_C)+2$ times. For that, for any iteration $t \leq T^*$, let k_t denote the product such that $k_t \in \mathcal{N}_t \setminus \mathcal{M}_t$ and $h_{k_t}(u_{t+1}) \geq h_j(u_{t+1})$ for all products $j \in \mathcal{N}_t \setminus Mscr_t$ and $u_{t+1} = R(\mathcal{M}_{t+1})$. Now define the function

$$g(u) = \begin{cases} h_{k_t}(u) - \delta_C u & \text{for } u_t < u \leq u_{t+1}, t \leq T^*, \\ h_{k_1}(u_1) - \delta_C u_1 & \text{for } u = u_1. \end{cases}$$

Note that for the above definition to be meaningful, for any $t \leq T^*$, we need to show that $u_t \leq u_{t+1}$. This should be true because by (11), it follows that for $u = R(\mathcal{M}_{t+1})$, we have $h_{i^*}(u) \leq h_{j^*}(u)$; this in turn implies that $H_{\mathcal{M}_t}(u) \leq H_{\mathcal{M}_{t+1}}(u)$, which implies by Proposition 1 that $u_t = R(\mathcal{M}_t) \leq R(\mathcal{M}_{t+1}) = u_{t+1}$. It is easy to see that the function $g(\cdot)$ is piecewise linear. However, note that it may not be continuous.

Now fix a product i, and for this product we argue that it can be exchanged at most $\bar{C}(\delta_C)$ times. For that let t_1 be an iteration in which i is exchanged-out and t_2 be the first iteration after t_1 when i is exchanged-in. Let u_1, u_2 denote $R(\mathcal{M}_{t_1+1})$ and $R(\mathcal{M}_{t_2+1})$ respectively. Furthermore, let k_1 and k_2 respectively denote the products k_{t_1} and k_{t_2}. It now follows from (11) that

$$\begin{aligned} h_i(u_1) &\leq h_{k_1}(u_1) - \delta_C u_1 = g(u_1) \\ h_i(u_2) &\geq h_{k_2}(u_2) - \delta_C u_2 = g(u_2). \end{aligned}$$

This implies that the line $h_i(\cdot)$ is below $g(\cdot)$ at u_1 and above $g(\cdot)$ at u_2. We now argue that $h_i(\cdot)$ intersects $g(\cdot)$ at some $u_1 \leq u \leq u_2$ i.e., $h_i(u) = g(u)$. If $g(\cdot)$ were continuous, this assertion would immediately follow from the intermediate value theorem. However, the way we have defined $g(\cdot)$, it may be discontinuous at some u_t with $t_1 < t \leq t_2$. Now the only way $h_i(\cdot)$ and $g(\cdot)$ do not intersect is if for some $t_1 < t \leq t_2$,

$$g(u_t^-) < h_i(u_t) < g(u_t^+) \quad \text{and} \quad h_i(u) > g(u) \text{ for } u_t \leq u \leq u_2.$$

We argue that this cannot happen. For that consider iteration t. By definition $i \notin \mathcal{M}_t$. Since $\mathcal{N}_t = \mathcal{N}$, it follows by our definition that $h_{k_t}(u_{t+1}) \geq h_i(u_{t+1})$, which in turn implies that $g(u_{t+1}) \geq h_i(u_{t+1})$ resulting in a contradiction. Thus, $h_i(\cdot)$ intersects $g(\cdot)$ from below at some u such that $u_1 \leq u \leq u_2$.

Hence, we can correspond each exchange-out with an intersection point corresponding to $h_i(\cdot)$ intersecting $g(\cdot)$ from below. This implies that the total number of exchage-outs can be bounded above by one plus the number of times $h_i(\cdot)$ intersects $g(\cdot)$ from below beyond u_i, where u_i is the revenue of the assortment $\mathcal{M}_t$ immediately after i is added to it (either through an exchange-in or addition). Note that $h_i(\cdot)$ intersects $g(\cdot)$ at $u \geq u_i$ if and only if $w_i \leq w_{k(u)}$ and $h_{k(u)}(u_i) \geq h_i(u_i)$, where $k(u)$ is the product such that $k(u) = k_t$, where $u_t < u \leq u_{t+1}$. Thus, the number of intersection points can be bounded above by the number of products k such that $h_k(u_i) \geq h_i(u_i)$. We now argue that $i \in \bar{B}_{S+1}(\delta_C, u_i)$. If this is true, then it implies that there can be at most $|\bar{B}_{S+1}(\delta_C, u_i)| \leq \bar{C}(\delta_C)$ intersection points, which immediately implies that there can be at most $1 + \bar{C}(\delta_C)$ exchange-outs.

The only thing we are left with is to argue that $i \in \bar{B}_{S+1}(\delta_C, u_i)$. To see this, let $\tilde{\mathcal{M}}$ be the assortment obtained after i is added or exhanged-in for the first time. Then, according to our definition, we have that $u_i = R(\tilde{\mathcal{M}})$. Further, since $H_{B(u_i)}(u_i) - H_{\tilde{\mathcal{M}}}(u_i) > 0$, there exists a product $k \in B(u_i) \setminus \tilde{\mathcal{M}}$. It now follows by Proposition 2 that

$$h_i(u_i) \geq h_k(u_i) - \delta_C u_i \geq h_{i_{S+1}(u_i)} - \delta_C u_i,$$

where $i_{S+1}(u_i)$ is as defined above i.e., $i_{S+1}(u_i) \stackrel{\text{def}}{=} \arg\min_{j \in B(u_i)} h_j(u_i)$. It now follows by the definition of $\bar{B}_{S+1}(\delta_C, u_i)$ that $i \in \bar{B}_{S+1}(\delta_C, u_i)$.

Case 2: Subroutine terminates because no further additions or exchanges increase revenue. We now consider the case when subroutine terminates at iteration T because no further additions or exchanges increase the revenue. Now there are two possibilities: either $\mathcal{N}_t = \mathcal{N}$ for all $t \le T$ or not. In the latter case let T^* be the first iteration t when $\mathcal{N}_t \subset \mathcal{N}$. It then follows from our arguments for the above case that there exists an iteration $t^* \le T^*$ that satisfies the properties of the lemma. Thus, we consider the case when $\mathcal{N}_t = \mathcal{N}$ for all $t \le T^*$. Assuming this, we prove the result by contradiction. In particular, suppose at the end of iteration T we have

$$H_{B(u)}(u) - H_{\mathcal{M}_{T+1}}(u) \ge 2\tilde{C}_u \delta_C u, \tag{13}$$

Now consider

$$H_{B(u)}(u) - H_{\mathcal{M}_{T+1}}(u) = \sum_{k \in \mathcal{M}_3} h_j(u) + \sum_{i \in \mathcal{M}_1, j \in \mathcal{M}_2} \left(h_j(u) - h_i(u)\right),$$

where as above, $\mathcal{M}_1$ denotes the assortment $\mathcal{M}_{T+1} \setminus B(u)$ and the set $B(u) \setminus \mathcal{M}_{T+1}$ is partitioned into $\mathcal{M}_2 \cup \mathcal{M}_3$ such that $\mathcal{M}_2 \cap \mathcal{M}_3 = \emptyset$ and $|\mathcal{M}_2| = |\mathcal{M}_1|$; such a partitioning is possible since $|B(u)| = S+1$ (which follows from (6) and the hypothesis that $|M^{\mathrm{OPT}}_{S+1} = S+1|$) and $|\mathcal{M}_{T+1}| \le S+1$. It now follows that one of the following conditions should hold: either (1) there exists a pair of products $i \in \mathcal{M}_1$ and $j \in \mathcal{M}_2$ such that $h_j(u) - h_i(u) > 2\delta_C u$, or (2) if $\mathcal{M}_3 \neq \emptyset$, then there exists a product $k \in \mathcal{M}_3$ such that $h_3(u) > 2\delta_C u$. Otherwise, it is easy to see that $H_{B(u)}(u) - H_{\mathcal{M}_{T+1}}(u) \le 2\tilde{C}_u \delta_C u$, where $\tilde{C}_u = |B(u) \setminus \mathcal{M}_{T+1}| + 1$, contradicting (13). We consider each of the cases in turn.

First, suppose that there exist a pair of products $i \in \mathcal{M}_1$ and $j \in \mathcal{M}_2$ such that $h_j(u) - h_i(u) > 2\delta_C u$. Let $\tilde{\mathcal{M}}$ denote the assortment $(\mathcal{M}_{T+1} \setminus \{i\}) \cup \{j\}$. We can then write

$$H_{\tilde{\mathcal{M}}}(u) - H_{\mathcal{M}_{T+1}}(u) = h_j(u) - h_i(u) > 2\delta_C u.$$

Since $H_{\mathcal{M}_{T+1}}(u) = u$ and $\delta_C \ge \delta(\tilde{\mathcal{M}})$, it follows that by Proposition 1 that $\tilde{R}\tilde{\mathcal{M}} > \tilde{R}\mathcal{M}_{T+1}$. This contradicts the assumption that the subroutine terminates with $\mathcal{M}_{T+1}$ because no further additions or exchanges result in an increase of revenue.

Next, suppose $\mathcal{M}_3 \neq \emptyset$ and $h_k(u) > 2\delta_C u$ for some $k \in \mathcal{M}_3$. Now let $\tilde{\mathcal{M}} = \mathcal{M}_{T+1} \cup \{k\}$; note that since $\mathcal{M}_3 \neq \emptyset$, it must be that $|\mathcal{M}_{T+1}| = S$. We can now write

$$H_{\tilde{\mathcal{M}}}(u) - H_{\mathcal{M}_{T+1}}(u) = h_k(u) > 2\delta_C u.$$

Since $H_{\mathcal{M}_{T+1}}(u) = u$ and $\delta_C \ge \delta(\tilde{\mathcal{M}})$, it follows that by Proposition 1 that $\tilde{R}\tilde{\mathcal{M}} > \tilde{R}\mathcal{M}_{T+1}$. This contradicts the assumption that the subroutine terminates with $\mathcal{M}_{T+1}$ because no further additions or exchanges result in an increase of revenue. This finishes the proof of this case.

The proof of the lemma now follows.

5. Summary and discussion

This paper focused on using choice models to make decisions. Assuming that we have access to a revenue prediction subroutine, we designed an algorithm to find an approximation of the optimal assortment with as few calls to the revenue subroutine as possible.

We designed a general algorithm for the optimization of set-functions to solve the static assortment optimization algorithms. Most existing algorithms (both exact and approximate) heavily exploit the structure of the assumed choice model; consequently, the existing algorithms – even without any guarantees – cannot be used with other choice models like the probit model or the mixture of MNL models with a continuous mixture. Given these issues, we designed an algorithm that is (a) not tailored to specific parametric structures and (b) requires only a subroutine that gives revenue estimates for assortments. Our algorithm is a sophisticated form of greedy algorithm, where the solution is constructed from a smaller assortment through greedy additions and exchanges. The algorithm is proved to find the optimal assortment exactly when the underlying choice model is the MNL model. We also showed that the algorithm is robust to errors in the revenue estimates provided by the revenue subroutine, as long as the underlying choice model is the MNL model.

References

P. Rusmevichientong, Z.J. Max Shen, and D.B. Shmoys. A ptas for capacitated sum-of-ratios optimization. *Operations Research Letters*, 37(4):230–238, 2009.

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

P. Rusmevichientong, D. Shmoys, and H. Topaloglu. Assortment optimization with mixtures of logits. Technical report, Tech. rep., School of IEOR, Cornell University, 2010b.

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