Assortment Optimization under the Multinomial Logit Model with Product Synergies

Venus Lo[a,1,*], Huseyin Topaloglu[b]

[a]School of Operations Research and Information Engineering, Cornell University, Ithaca, New York, United States 14853
[b]School of Operations Research and Information Engineering, Cornell Tech, New York, New York, United States 10044

Abstract

In synergistic assortment optimization, a product's attractiveness changes as a function of which other products are offered. We represent synergy structure graphically. Vertices denote products. An edge denotes synergy between two products, which increases their attractiveness when both are offered. Finding an assortment to maximize retailer's expected profit is NP-hard in general. We present efficient algorithms when the graph is a path, a tree, or has low treewidth. We give a linear program to recover the optimal assortment for paths.

Keywords: Assortment optimization, synergy, multinomial logit model

1. Introduction

Modeling customers' purchasing behaviours is critical in retail operations because it affects the retailer's decisions on which products to offer in order to maximize his profit. Early work in inventory management assumed that demands are independent of the offered assortment. Subsequent work in revenue management recognized that demand for a product might decrease if customers have more options. However, most choice models do not allow for synergy effects. Synergy can increase the demand for a product when it is seen with some other products.

We study the assortment optimization problem under a synergistic version of the multinomial logit model (MNL). Customers associate a preference weight with each product, and the preference weight can increase via synergy. Synergy occurs between a pair of products when the retailer offers both of them in his assortment, even when customers purchase at most one product. The purchase probabilities of either product may increase or decrease, depending on the increase in preference weight and the weights of the other product. Marketing research shows that retailers can increase demand by offering a less attractive product to highlight the target product. The assortment optimization problem is to select a subset of products to offer, in order to maximize the retailer's expected profit.

Our Contributions: We consider a retailer who has access to n products. Each product has a base preference weight, which describes a customer's preference for the product when it is seen alone. Each pair of products, i and j, also have a pair of synergy weights, which describes how synergy from product i acts on j and vice versa. The purchase probability of a product is proportional to its preference weight in the offered assortment.

We use a graph to depict pairs of products with positive synergy when they are seen together. The assortment optimization problem under synergistic MNL is NP-hard for general synergy graphs. We restrict the product pairs with synergy and study the special cases of a synergy path and a synergy tree. We present algorithms that find the optimal assortment via dynamic programming, with runtime polynomial in the number of products. We extend our dynamic program to consider synergy graphs with low treewidth. In the case of the synergy path, we present a linear program that can recover the optimal assortment.

Literature Review: Our paper is inspired by the welfare-based choice models of [9]. The authors introduced a welfare function, which defined the welfare of an assortment as a function of its products' utilities. Purchase probabilities are given by the gradients of the welfare functions with respect to each product. The welfare function has three properties. Monotonicity ensures that welfare increases when all the utilities increase. Translation invariance states that purchase probabilities remain the same if all utilities increased by the same amount. Convexity ensures that a high-utility product improves welfare more than multiple low-utility products.

The authors defined operations on welfare-based choice models which create new welfare-based choice models [9]. One resulting model is the basis for our synergistic MNL. In their model, if two products create synergy, then the sum of their synergy weights is equal to a weighted geometric mean of their base preference weights. We allow synergy weights to be any non-negative values. They fo-

*Corresponding author
Email addresses: venus.hl.lo@cityu.edu.hk (Venus Lo), topaloglu@orie.cornell.edu (Huseyin Topaloglu)
[1]Present address - Department of Management Sciences, City University of Hong Kong, Kowloon Tong, Hong Kong SAR

cused on defining a new class of choice models, and did not consider assortment optimization. We focus on the latter problem.

Evidence of synergy exists in marketing literature. Product sales can increase when an inferior product is introduced [23]. In one study, a retailer selling a bread maker introduced a second, over-priced bread maker to his store. The demand for the original bread maker increased even though the assortment became larger. Some products can trigger a change in preference; [11] found that cookies sales increased in a cafeteria if applesauce was offered but not if green beans were offered.

A traditional approach to choice modeling is via utility-maximization. In the class of random utility models, a customer's utility for a product is the sum of its mean utility plus a random noise. MNL is the most famous model in this class, and the random noise for each product follows an independent standard Gumbel distribution [15, 18]. A product's preference weight is the exponential of its mean utility, and its purchase probability is proportional to its preference weight in the assortment.

MNL has the independence of irrelevant alternative (IIA) property: the ratio of two products' purchase probabilities is unchanged regardless of what other products are offered. Our model follows MNL such that purchase probabilities are proportional to preference weights, but it is not subject to IIA because synergy can increase the preference weights of products. We describe extensions to MNL; details of these choice models can be found in [26].

In the nested logit model, products are partitioned into nests by similarities. A customer chooses a nest, and then chooses a product within her chosen nest. The retailer can increase the probability that a customer chooses a nest by adding products to the nest, but that might not increase the purchase probability of individual products. In fact, incorporating synergy into the nested logit model might sacrifice its utility-maximizing property. The nested logit model has been extended to the d-level nested logit model with $d > 2$ levels [14], and to the generalized extreme value model where products belong to multiple nests [26].

In the mixed logit model (MMNL), different customer types have different sets of preference weights [17, 5, 21]. MMNL can approximate any random utility choice model [17], and [6] gave a fully-polynomial time approximation scheme (FPTAS) to compute an assortment that guarantees $(1 - \epsilon)$-fraction of the optimal expected profit.

The random noises of utilities do not have to follow the Gumbel distribution. In the probit model, the noises follow a multivariate normal distribution and the model does not exhibit the IIA property [25]. Choice models outside the class of random utility models include the Markov Chain choice model [1, 8] and the non-parametric choice model [16, 12]. [7] studied a special case of the non-parametric choice model, where customers consider at most k products for some small, fixed k.

Our underlying parametric problem requires maximizing a quadratic function subject to binary variables. Unconstrained quadratic binary programming is NP-hard, and can be rewritten as an integer program. This problem can be classified by its underlying graph, where vertex i represents variable x_i and edge ij exists if the coefficient of the quadratic term $x_i x_j$ is non-zero. [19] studied the graph and the related integer program, and identified cases where the convex hull of the feasible region is integral. We use their approach to classify cases of the assortment optimization problem under synergistic MNL that can be solved efficiently. A survey of quadratic binary programming can be found in [13].

Our linear program for finding the optimal assortment under a synergy path is similar to the sales-based linear program (SBLP) described by [10]. The authors studied the network revenue management problem under a generalization of MNL, where the retailer offers assortments over time subject to resource constraints. When there is one sales period, then their SBLP recovers the optimal assortment for their model. We construct our version of the SBLP by taking the dual of the linear program formulation of our dynamic program.

Organization: In Section 2, we describe the synergistic MNL model and its corresponding parametric problem, as well as the synergy graph. In Section 3, we focus on the synergy path and synergy tree, and use dynamic programming to solve the parametric problem. In Section 4, we present a linear program which recovers the optimal assortment when we have a synergy path. In Section 5, we consider other forms of the synergy weights and incorporate synergy into MMNL. We conclude in Section 6.

2. General Model and NP-hardness

We describe the general model and the parametric problem, and show that the assortment optimization problem under general synergy effects is NP-hard.

2.1. The Model

The retailer has access to n products: $\{1, \ldots, n\}$. Product i generates a profit of r_i when it is purchased by a customer. We do not require $r_i > 0$ for all products, only that at least one product has positive profit. Note that the retailer can offer products with negative profit if they increase the demand of highly profitable products. The retailer chooses an assortment from $\{1, \ldots, n\}$ to offer, which we denote by a binary vector $x \in \{0, 1\}^n$ such that $x_i = 1$ if product i is in the assortment and 0 otherwise.

We describe customers' preferences by preference weights, and purchase probabilities are proportional to these weights. The preference weight of product i is a function of the assortment x because synergy from other products in the assortment can increase its preference weight. Product i has a base preference weight of $u_i \geq 0$ when it is the only product in the assortment. When product j is added to the assortment, synergy between products i and j increases the preference weight of product i additively. We denote the synergy effect of product j on i by $v_i^j \geq 0$. Simultaneously, product i increases the preference weight of product j by v_j^i. Given an assortment x, the preference weight of product i when it is offered is $u_i + \sum_{j \neq i} v_i^j x_j$.

We discuss the possibility of negative synergy weights in Section 5, but we point out that it is possible for product j to create synergy for product i without synergy in the reverse direction by setting $v_i^j > 0$ and $v_j^i = 0$. Then the probability of selling product i may increase when product j is added into the assortment, but the probability of selling product j cannot increase when product i is added into the assortment.

Upon seeing assortment x, a customer's probability of purchasing product i is proportional to the preference weight of product i in the assortment, along with the no-purchase option. The no-purchase option is her ability to leave the store without a purchase. We scale the preference weight of the no-purchase option to 1 without loss of generality. In response to assortment x, the probability that the customer purchases a product i is:

$$P_i(x) = \frac{\left(u_i + \sum_{j \neq i} v_i^j x_j\right) x_i}{1 + \sum_{k=1}^n \left(u_k + \sum_{j \neq k} v_k^j x_j\right) x_k}.$$

The customer purchases at most one product, and synergy simply makes products look more attractive when they are offered concurrently. The retailer's problem is to find an assortment x that maximizes his expected profit: $\Pi(x) = \sum_{i=1}^n r_i P_i(x)$.

For cleaner notation, we define $\bar{r}_{i,j}$ as the weighted average of the profits of products i and j, and $v_{i,j}$ as the total increase to the preference weight of the assortment due to synergy when both products i and j are offered, with the convention that $i < j$. When $v_j^i = v_i^j = 0$ so that there is no synergy in either directions, then $\bar{r}_{i,j} = 0$ and $v_{i,j} = 0$. Otherwise, define $\bar{r}_{i,j} = (r_i v_i^j + r_j v_j^i)/(v_i^j + v_j^i)$ and $v_{i,j} = v_i^j + v_j^i$.

Using this notation, our assortment optimization problem $\max_{x \in \{0,1\}^n} \Pi(x)$ expands out to:

$$\max_{x \in \{0,1\}^n} \frac{\sum_{i=1}^n r_i u_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \bar{r}_{i,j} v_{i,j} x_i x_j}{1 + \sum_{i=1}^n u_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n v_{i,j} x_i x_j}. \quad (1)$$

2.2. The Parametric Problem

We apply a standard parametrization technique for fractional combinatorial problems [20]. Suppose there exists an assortment x with expected profit greater or equal to δ. By rearranging $\Pi(x) \geq \delta$, we observe:

$$\sum_{i=1}^n (r_i - \delta) u_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n (\bar{r}_{i,j} - \delta) v_{i,j} x_i x_j \geq \delta.$$

We can maximize the left side over all assortments and the inequality would still hold. The parametric problem

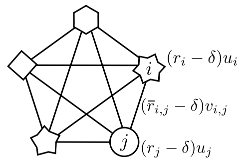

Figure 1: Synergy graph on five products - Vertices i and j represent products and the edge ij represents the synergy between these products. In the parametric problem, the vertices contribute $(r_i - \delta)u_i$ and $(r_j - \delta)u_j$ respectively, and the edge contributes $(\bar{r}_{i,j} - \delta)v_{i,j}$.

corresponding to Problem (1) is:

$$h(\delta) = \max_{x \in \{0,1\}^n} \sum_{i=1}^n (r_i - \delta) u_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n (\bar{r}_{i,j} - \delta) v_{i,j} x_i x_j. \quad (2)$$

Claim 1. *Given $h(\delta)$ as defined above, let δ^* be the optimal expected profit of our assortment optimization problem. Then the following are true: i) $h(\delta) > \delta$ if $\delta < \delta^*$, ii) $h(\delta) < \delta$ if $\delta > \delta^*$, and iii) $h(\delta) = \delta$ if $\delta = \delta^*$.*

Suppose we can solve Problem (2) with corresponding optimal objective value $h(\delta)$ for any $\delta \geq 0$. Since $h(0) > 0$ and $h(\delta)$ is monotone decreasing to 0, one method to find δ^* is Newton's method. Let m denote the number of pairs i, j such that $v_{i,j} \neq 0$. We can rewrite the quadratic functions in the numerator and denominator of $\Pi(x)$ with linear functions by introducing $O(m)$ new binary variables and constraints [19]. Newton's method finds δ^* in $O(m^2 \log^2 m)$ iterations of computing $h(\delta)$ when the feasible region is a subset of $\{0,1\}^{O(m)}$. Hence, for a general synergy graph, we can solve Problem (1) in polynomial time if we can solve Problem (2) in polynomial time.

Unfortunately, Problem (2) is a special case of quadratic binary programming, which is NP-hard in general [13]. We prove that Problem (1) is NP-hard in the next theorem via a reduction from the maximum independent set problem. All proofs are in the online supplement.

Theorem 2. *The assortment optimization problem under synergistic MNL is NP-hard.*

Instead, one way to classify quadratic binary programs and identify cases which are solvable in polynomial-time is to consider the underlying graph [19]. Construct a graph $G = (V, E)$ such that vertex i represents variable x_i and an edge goes from vertex i to j if the coefficient of the quadratic term $x_i x_j$ is non-zero. In Problem (2), the coefficient $(\bar{r}_{i,j} - \delta) v_{i,j}$ changes as we vary δ and we want to consider the quadratic binary program for all possible values of δ, so we add an edge ij whenever $v_{i,j} \neq 0$. Then $V = \{1, \ldots, n\}$ and $E = \{ij : v_{i,j} \neq 0\}$, with $n = |V|$ and $m = |E|$. We call this graph the synergy graph (e.g. Figure 1). We write Problems (1) and (2) using graph notation from hereon, unless G is a path so that there is a clear ordering of the products.

If our synergy graph has at least two components, then Problem (2) can be broken up into smaller sub-problems

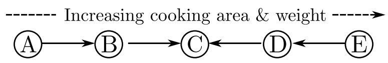

Figure 2: Synergy graph for portable grills which are vertically differentiated - The arrows point in the direction that synergy is created.

containing only the products in the component. Without loss of generality, we assume that our synergy graph is connected. In practice, synergy does not exist between arbitrary pairs of products. We focus on the cases where the synergy graph is a path or a tree and extend our results to consider synergy graphs with low treewidth.

3. Optimal Assortments on Synergy Paths and Trees

We give dynamic programs to solve Problem (2) when the synergy graph is a path, a tree, or has low treewidth.

3.1. A Synergy Path

The synergy path models products that are vertically differentiated. Consider the portable grills in Figure 2, which increase in weight as their cooking areas become larger [22, 24]. A mid-weight, mid-size grill becomes more attractive when it is seen with an extreme alternative, as it offers a compromise between the two features. In the presence of A, B may feel bigger because it is not the smallest option [24]. In the presence of E, D may feel lighter because it is not the heaviest grill. In this example, the presence of A exerts positive synergy on B, and the presence of E exerts positive synergy on D.

Since the synergy graph is a path, we can order the products so that product i creates synergy with products $i-1$ and $i+1$ only. Let $\bar{r}_i = \bar{r}_{i,i+1}$ and $v_i = v_{i,i+1}$ for $i = 1, \dots, n-1$ when we consider the synergy path. Problem (2) simplifies to:

$$h(\delta) = \max_{x \in \{0,1\}^n} \sum_{i=1}^{n} (r_i - \delta)\, u_i x_i + \sum_{i=1}^{n-1} (\bar{r}_i - \delta)\, v_i x_i x_{i+1}.$$

We use a dynamic program to compute the value of $h(\delta)$. For $i \geq 2$, define the value function $V_i(x_{i-1})$ to be the maximum value that products i to n can contribute to the objective function of Problem (2), given the state x_{i-1} for product $i-1$. At product 1, there is no preceding product whose state we have to consider. We attribute the synergy between products $i-1$ and i to product i, given the decision x_{i-1}. Hence, our value function can be written as the following, with $h(\delta) = V_1$:

$$V_1 = \max_{\substack{x_i \in \{0,1\}: \\ i=1,\dots,n}} \sum_{i=1}^{n} (r_i - \delta)\, u_i x_i + \sum_{i=1}^{n-1} (\bar{r}_i - \delta)\, v_i x_i x_{i+1},$$

$$V_i(x_{i-1}) = \max_{\substack{x_j \in \{0,1\}: \\ j=i,\dots,n}} (\bar{r}_{i-1} - \delta) v_{i-1} x_{i-1} x_i + \sum_{j=i}^{n} (r_j - \delta) u_j x_j + \sum_{j=i}^{n-1} (\bar{r}_j - \delta) v_j x_j x_{j+1}.$$

The presence or absence of product $i-1$ does not affect products $i+1$ to n once we have decided whether or not to offer product i. If product i is offered, then its contribution is $R_i^\delta(x_{i-1}) := (\bar{r}_{i-1} - \delta) v_{i-1} x_{i-1} + (r_i - \delta) u_i$. The first term is the synergy between products $i-1$ and i, given the decision of offering product $i-1$. The second term is the base contribution of product i. We can rewrite our value functions using the definition of $R_i^\delta(x_{i-1})$ and $V_{i+1}(x_i)$ to get our dynamic program:

$$V_1 = \max_{x_1 \in \{0,1\}} (r_1 - \delta) x_1 + V_2(x_1), \tag{3}$$

$$V_i(x_{i-1}) = \max_{x_i \in \{0,1\}} R_i^\delta(x_{i-1}) \cdot x_i + V_{i+1}(x_i), \quad \forall i = 2, \dots, n,$$

$$V_{n+1}(x_n) = 0.$$

Our base cases are $V_{n+1}(0) = V_{n+1}(1) = 0$, and we compute the dynamic program backwards from product n to 1. If we decide $x_i = 0$, then $V_i(x_{i-1}) = V_{i+1}(0)$, and we immediately lose the synergy with both products $i-1$ and $i+1$, as well as the base value $(r_i - \delta) u_i$ from product i. If we decide $x_i = 1$, then $V_i(x_{i-1}) = R^\delta(x_{i-1}) + V_{i+1}(1)$. We get the base value from product i, and we may get the synergy with product $i-1$, depending on the value of x_{i-1}. Furthermore, we have the opportunity to create synergy with product $i+1$ at the next value function.

We conclude with the runtime analysis. For a fixed δ, we can compute $h(\delta)$ in $O(n)$ operations because there are n products, and $O(1)$ states and $O(1)$ decisions at each state per product. Since the synergy graph is a path, we have $m = n - 1$, so Newton's method finds δ^* in $O(n^2 \log^2 n)$ iterations of computing $h(\delta)$. The optimal assortment can be computed in $O(n^3 \log^2 n)$ operations. Alternatively, we can find δ^* by solving a linear program in Section 4, which has $O(n)$ variables and constraints. This allows us to compute the optimal assortment with $O(n)$ operations plus the solution of one linear program.

3.2. A Synergy Tree

Suppose the synergy graph is a tree. For example, a generic-brand product creates synergy with the entry-level products of national brands. In turn, the entry-level product of each national brand creates synergy with the higher-end products of its brand [23].

Given a product i, let p_i denote its parent and C_i denote the set of its children. We index the products so that $p_i < i < c$ for $c \in C_i$. If product i is a leaf of the synergy tree, then $C_i = \emptyset$. Let T_i denote the set of products in the subtree rooted at product i. Since each non-root product only has one parent, we let $\bar{r}_i = \bar{r}_{p_i,i}$ and $v_i = v_{p_i,i}$ for cleaner notation. We can rewrite Problem (2) as:

$$h(\delta) = \max_{x \in \{0,1\}^n} \sum_{i \in V} (r_i - \delta)\, u_i x_i + \sum_{i \in V} \sum_{c \in C_i} (\bar{r}_c - \delta)\, v_c x_i x_c.$$

Suppose we are considering whether or not to offer product i, and we already know whether its parent is offered (i.e. x_{p_i}). Then the decision of whether or not to offer product i is independent of the decisions for the ancestors of p_i, as well as the other children of p_i. We can focus on the subtree rooted at i. When product i is not the root, we define our value function $V_i(x_{p_i})$ as the maximum contribution from products in T_i to the objective function of Problem (2), given our decision on offering product p_i. We attribute the synergy between products p_i and i to product i and the value function $V_i(x_{p_i})$. The value function at the root, V_{root}, has no synergy from a parent and is by definition equal to $h(\delta)$. Our value functions are:

$$V_{\text{root}} = \max_{\substack{x_i \in \{0,1\}:\\ i \in V}} \sum_{i \in V} (r_i - \delta)\, u_i x_i + \sum_{i \in V} \sum_{c \in C_i} (\bar{r}_c - \delta)\, v_c x_i x_c,$$

$$V_i(x_{p_i}) = \max_{\substack{x_j \in \{0,1\}:\\ j \in T_i}} (\bar{r}_i - \delta) v_i x_{p_i} x_i + \sum_{j \in T_i} (r_j - \delta) u_j x_j + \sum_{j \in T_i} \sum_{c \in C_j} (\bar{r}_c - \delta)\, v_c x_j x_c.$$

We apply the strategy from the case of the synergy path. Suppose product i is offered in our assortment and it is not the root. Its contribution to the objective function of Problem (2) is the synergy with its parent given the decision x_{p_i} and its base contribution: $R_i^\delta(x_{p_i}) := (\bar{r}_i - \delta) v_i x_{p_i} + (r_i - \delta) u_i$. If product i is the root, then its contribution is $(r_{\text{root}} - \delta) u_{\text{root}}$. We can rewrite the value functions as a dynamic program, using the definition of $R_i^\delta(x_{p_i})$ and $V_c(x_i)$ for $c \in C_i$:

$$V_{\text{root}} = \max_{x_{\text{root}} \in \{0,1\}} (r_{\text{root}} - \delta) u_{\text{root}} x_{\text{root}} + \sum_{c \in C_{\text{root}}} V_c(x_{\text{root}}),$$

$$V_i(x_{p_i}) = \max_{x_i \in \{0,1\}} R_i^\delta(x_{p_i}) \cdot x_i + \sum_{c \in C_i} V_c(x_i), \quad \forall i \neq \text{root}. \tag{4}$$

We solve the dynamic program starting from the leaves until we reach the root. If product i is a leaf, then $C_i = \emptyset$ and $V_i(x_{p_i}) = \max_{x_i \in \{0,1\}} R_i^\delta(x_{p_i}) \cdot x_i$, so the base case looks similar to the synergy path's base case. If product i is not a leaf, then $x_i = 0$ implies that $V_i(x_{p_i}) = \sum_{c \in C_i} V_c(0)$. This means we lose the base contribution from product i and the synergy with its parent, as well as any chance of synergy with its children. If $x_i = 1$, then we keep the base contribution of product i and possibly synergy with its parent. The decision of whether we create synergy with each child is deferred to the corresponding child.

We need to compute the value function at each vertex of our synergy tree, and there are $O(1)$ states and $O(1)$ decisions at each state per product. Hence, we can solve the dynamic problem in Problem (4) and compute $h(\delta) = V_{\text{root}}$ in $O(n)$ operations for any fixed δ. Similar to the case of the synergy path, we have $m = n - 1$, so Newton's method can find δ^* in $O(n^2 \log^2 n)$ iterations of computing $h(\delta)$. Alternatively, we can find δ^* via a linear program similar to the one in Section 4. The number of operations to compute the optimal assortment when we have a synergy tree or a synergy path is on the same order.

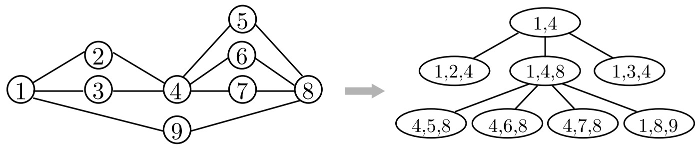

Figure 3: Tree decomposition - On the left, we have a synergy graph on nine products. On the right, we have a tree decomposition of the synergy graph. A vertex on the tree represents a subset of vertices on the original graph. The treewidth is 2.

The dynamic program for the synergy tree suggests that we can use tree decompositions to consider synergy graphs with low treewidth. Tree decompositions are a method to represent a graph with a tree such that each tree vertex represents a subset of vertices on the original graph, and each graph vertex is associated with a subtree on the tree. A graph can have many tree decompositions, and each decomposition is associated with a measure called width. The treewidth of a graph is the minimum width over all of its tree decompositions. Tree-based dynamic programs can be modified into efficient algorithms when the graph has low treewidth [28]. Details of this extension are deferred to Appendix A, and an example of a tree decomposition is depicted in Figure 3.

Theorem 3. *For a synergy graph G, suppose we are given a tree decomposition with width t and $O(n)$ vertices. Then it takes $O(2^{2t} n)$ operations to solve Problem (2). If we use Newton's algorithm to find δ^*, then the total operations needed to compute the optimal assortment is $O(2^{2t} n^5 \log^2 n)$.*

For example, if the synergy graph has no K_4 minor so that it is a series-parallel graph, then it has treewidth $t^* = 2$ [27, 4]. Theorem 3 assumes that a tree decomposition of low width is given. If a graph has treewidth t^*, algorithms exist to find a tree decomposition with width $t = t^*$ and at most $O(n)$ vertices, with runtime polynomial in n but exponential in t^* [2]. On the other hand, there exists algorithms to find a tree decomposition with width $t = O(t^* \log n)$ in polynomial time [3]. This increases the runtime of the dynamic program by a factor of n^2 compared to using a tree decomposition with minimal width.

4. Sales-Based Linear Program for Synergy Path

We revisit our synergy path, and present a linear program that reveals the optimal assortment. This allows a practitioner to take advantage of linear program solvers and avoids solving Problem (2) repeatedly. Proofs and details for this section are deferred to Appendix B.

Suppose the value of δ is fixed. The values of V_1 and $V_i(x_{i-1})$ are not constrained to be integers. If we simply want to compute the value of $h(\delta) = V_1$, then we can transform our dynamic program into a linear program by replacing the decision process at each state with two constraints: the value of $V_i(x_{i-1})$ is greater or equal to the outcomes of both decisions $x_i = 0$ and $x_i = 1$. We transform our dynamic program in Problem (3) into the following linear program with variables V_1 and V_i^x, where $x \in \{0,1\}$ represents the state x_{i-1} for $i \geq 2$. By construction, the optimal objective value of this linear program is $h(\delta)$.

$$\begin{aligned}
\min\ & V_1 && && (5)\\
\text{s.t. } & V_1 \geq V_2^0 \\
& V_1 \geq (r_1 - \delta)u_1 + V_2^1 \\
& V_i^0 \geq V_{i+1}^0 && \forall i = 2, \dots, n \\
& V_i^0 \geq (r_i - \delta)u_i + V_{i+1}^1 && \forall i = 2, \dots, n \\
& V_i^1 \geq V_{i+1}^0 && \forall i = 2, \dots, n \\
& V_i^1 \geq (\bar{r}_{i-1} - \delta)v_{i-1} + (r_i - \delta)u_i + V_{i+1}^1 && \forall i = 2, \dots, n \\
& V_{n+1}^0 = V_{n+1}^1 = 0.
\end{aligned}$$

Let $\mathcal{V}$ denote the set of constraints above. Claim 1 tells us that the linear program has optimal objective value δ^* if and only if we had created this linear program with $\delta = \delta^*$. Furthermore, the constraints in $\mathcal{V}$ are linear in δ, so we can treat δ as a variable. Specifically, we add an extra constraint to the linear program, $\delta = V_1$, to obtain:

$$\min_{V \in \mathcal{V}, \delta \in \mathbb{R}} \{V_1 : \delta = V_1\}. \tag{6}$$

Lemma 4. *The optimal objective value of Problem (6) is equal to the optimal expected profit of Problem (1), δ^*.*

Lemma 4 lets us find δ^* by solving a linear program with $O(n)$ constraints and variables, and immediately proceed to computing $h(\delta^*)$. This avoids iterating over different values of δ in Newton's method and repeatedly solving the dynamic program. These techniques also work for finding δ^* for the synergy tree, and the linear program for a synergy tree also has $O(n)$ constraints and variables.

We take one more step to find the optimal assortment directly via the solution of a linear program when we have a synergy path. Since Problem (6) is a feasible linear program with finite optimal objective value, strong duality states that its dual is feasible and has optimal objective value δ^*. By taking the dual of Problem (6) and making the appropriate transformation of variables, we obtain:

$$\begin{aligned}
\max_{y,w \geq 0}\ & \sum_{i=1}^{n} r_i y_i + \sum_{i=1}^{n-1} \bar{r}_i w_i && && (7)\\
\text{s.t. } & y_i/u_i \geq w_i/v_i && \forall i = 1, \dots, n-1 \\
& y_{i+1}/u_{i+1} \geq w_i/v_i && \forall i = 1, \dots, n-1 \\
& y_0 \geq y_i/u_i + y_{i+1}/u_{i+1} - w_i/v_i && \forall i = 1, \dots, n-1 \\
& y_0 + \sum_{i=1}^{n} y_i + \sum_{i=1}^{n-1} w_i = 1.
\end{aligned}$$

Problem (7) looks similar to the SBLP in [10], except for the terms related to synergy: w_i. If we can construct assortments from solutions to Problem (7), then we can interpret y_i as the probability that a customer purchases product i due to its base preference weight, and w_i as the additional probability that a customer purchases either products i or $i+1$ due to the synergy created when both are present. We would like to show that there is a one-to-one correspondence between extreme points of the feasible region in Problem (7) and assortments.

Lemma 5. *If (y, w) is an extreme point of the feasible region in Problem (7), then (y, w) satisfies these conditions:*

1. *If $w_i = 0$, then $y_i = 0$ or $y_{i+1} = 0$.*
2. *If $w_i > 0$, then $y_i/u_i = y_{i+1}/u_{i+1} = w_i/v_i$.*
3. *If $y_i > 0$, then $y_i/u_i = y_0$.*

We show that each assortment maps to a feasible solution of Problem (7).

Lemma 6. *Given an assortment x, there exists a solution (y, w) which is feasible to Problem (7) with objective value equal to $\Pi(x)$.*

We now consider the reverse direction, and show that an extreme point solution (y, w) maps to an assortment.

Lemma 7. *Suppose (y, w) is an extreme point solution to Problem (7). If $x_i = \mathbb{1}[y_i > 0]$, then assortment x has expected profit equal to the objective value of (y, w).*

Theorem 8. *Under synergistic MNL with a synergy path, we can recover the optimal assortment x^* by solving the linear program in Problem (7) for an optimal extreme point solution (y^*, w^*) and setting $x_i^* = \mathbb{1}[y_i^* > 0]$.*

In summary, the linear program allows us to compute the optimal assortment without running the dynamic program for multiple iterations.

5. Extensions

We look at three extensions of our problem. First, we remove the assumption that $v_i^j \geq 0$ for all $i, j \in V$. Second, we incorporate synergy into another choice model: MMNL. Third, we use multiplicative synergy weights so that preference weights change by multiplicative factors.

5.1. Negative Synergy Weights

Suppose product j cannibalizes the demand of product i, to the extent that product j looks more attractive but product i looks less attractive when both products are in the assortment. We can represent this scenario with a negative synergy weight on product i: $v_i^j < 0$. The parametrization techniques in Section 2 are valid if $|\sum_{j \neq i} \min\{v_i^j, 0\}| < u_i$, so that a product ultimately has a positive overall preference weight irrespective of what other products are offered. Note that MNL with a negative overall preference weight does not make sense. To ensure that all notations are well-defined and that the analyses in Sections 3 and 4 continue to hold, if $v_j^i, v_i^j \neq 0$, then we also require that $v_j^i + v_i^j \neq 0$, which is the condition for an edge to exist in our synergy graph. In Problem (7), we would replace $w_i \geq 0$ for all $i = 1, \dots n-1$ with $w_i \geq 0$ if $v_i > 0$ and $w_i \leq 0$ if $v_i < 0$, so that $w_i/v_i \geq 0$.

5.2. Incorporating Synergy into MMNL

Suppose there are K customer types. With probability λ_k, the customer is of type k. A customer of type k has base preference weight $u_i^k \geq 0$ for $i \in V$, and synergy weight from product j on product i of $v_i^{j,k} \geq 0$. In this extension, we assume that $r_i \geq 0$ for all $i \in V$. Then $\bar{r}_{i,j}^k \geq 0$ and $v_{i,j}^k \geq 0$ when we follow the definitions from Section 2 for $k = 1, \ldots, K$. We choose an assortment x to offer to any customer, regardless of her type:

$$\max_{x \in \{0,1\}^n} \sum_{k=1}^{K} \lambda_k \frac{\sum_{i \in V} r_i u_i^k x_i + \sum_{ij \in E} \bar{r}_{i,j}^k v_{i,j}^k x_i x_j}{1 + \sum_{i \in V} u_i^k x_i + \sum_{ij \in E} v_{i,j}^k x_i x_j}. \quad (8)$$

We approach Problem (8) by incorporating the expected profit from customer types 2 to K into the constraints and maximizing the expected profit from type 1 customers. Suppose we are given $(N_2, D_2, \ldots, N_K, D_K) \in \mathbb{R}_+^{2(K-1)}$ and we construct the following problem:

$$\begin{aligned} \max_{x \in \{0,1\}^n} \; & \frac{\sum_{i \in V} r_i u_i^1 x_i + \sum_{ij \in E} \bar{r}_{i,j}^1 v_{i,j}^1 x_i x_j}{1 + \sum_{i \in V} u_i^1 x_i + \sum_{ij \in E} v_{i,j}^1 x_i x_j} & (9) \\ \text{s.t.} \; & \sum_{i \in V} r_i u_i^k x_i + \sum_{ij \in E} \bar{r}_{i,j}^k v_{i,j}^k x_i x_j \geq N_k \quad \forall k = 2, \ldots, K \\ & 1 + \sum_{i \in V} u_i^k x_i + \sum_{ij \in E} v_{i,j}^k x_i x_j \leq D_k \quad \forall k = 2, \ldots, K. \end{aligned}$$

Any feasible solution to this integer program would ensure that the expected profit from a type k customer is at least N_k / D_k. An assortment x with objective value δ_1 to Problem (9) would guarantee $\Pi(x) \geq \lambda_1 \delta_1 + \sum_{k=2}^{K} \lambda_k \frac{N_k}{D_k}$.

Taking the approach of [6], we use a geometric grid with accuracy level ϵ to construct tuples $(N_2, D_2, \ldots, N_K, D_K)$. If we solve Problem (9) for every tuple on our grid, then the assortment with the largest expected profit would guarantee $(1-\epsilon)$-fraction of the optimal expected profit. Instead of solving Problem (9) exactly, we round the coefficients of the constraints in the manner of the knapsack problem to obtain an approximate solution, yielding an FPTAS for a fixed K. We refer to Problem (9) with rounded coefficients as Problem (9'). The details are deferred to Appendix C.

The above discussion for constructing an FPTAS applies for a general synergy graph. The last hurdle is to solve Problem (9') efficiently, and it is unclear how we can achieve this goal even when the synergy graph is a tree. In fact, we can only construct a dynamic program to solve the parametric problem corresponding to Problem (9') when the synergy graph is a path. Hence, we only give a result pertaining to an FPTAS for assortment optimization under synergistic MMNL with a synergy path.

Lemma 9. *Suppose x^* is the optimal assortment under synergistic MMNL, and $\epsilon \in (0,1)$. Let $R = \max_i r_i$ and $U = \max\{1, \max_{k,i} u_i^k, \max_{k,i} v_i^k\}$, and r, u be defined similarly as the minimum parameters. If the synergy graph is a path, then there exists an algorithm that finds an assortment achieving expected profit of $(1-\epsilon)\Pi(x^*)$ in $O\left(\frac{\log^{K-1}(nRU/r) \log^{K-1}((n+1)U/u)\, n^{2K+1} \log^2 n}{\epsilon^{4(K-1)}}\right)$ operations.*

5.3. Multiplicative Synergy Effects

Inspired by [9], we assumed that the synergy weights are additive. If we take the approach that synergy invokes an additive change in the utility of products, then this translates to multiplicative synergy weights. The synergy effect of product j on i is $v_i^j > 0$ and the preference weight of product i is $u_i \cdot \prod_{j \neq i} (v_i^j)^{x_j}$. Note that multiplicative updates easily incorporate a reduction in utility by letting $v_i^j < 1$. To construct our synergy graph, we let $V = N$ and $E = \{ij : v_j^i \neq 1 \text{ or } v_i^j \neq 1\}$. Our problem becomes:

$$\max_{x \in \{0,1\}^n} \frac{\sum_{i \in V} r_i u_i x_i \cdot \prod_{ij \in E} (v_i^j)^{x_j}}{1 + \sum_{i \in V} u_i x_i \cdot \prod_{ij \in E} (v_i^j)^{x_j}}. \quad (10)$$

We cannot consider the marginal increase in the assortment's total preference weight from synergy between products i and j. Hence, we do not combine the terms v_j^i and v_i^j. We focus on synergy paths and trees to show how multiplicative synergy weights affect our algorithms.

Let $h(\delta)$ be the optimal objective value if we parametrize Problem (10) with δ. To understand the difficulty of using multiplicative weights, consider our value function for the synergy path in Subsection 3.1. To simplify notation, let $\overline{v_i} = v_i^{i+1}$ and $\underline{v_i} = v_i^{i-1}$. We cannot consider the marginal effect on product i from $\overline{v_i}$ and $\underline{v_i}$ separately. Instead, we need the values of x_{i-1}, x_i, and x_{i+1} to determine the total synergy weight in the value function. We modify our value function to remember the decisions for two products. Let $V_i(x_{i-1}, x_i)$ be the maximum value that products i to n can contribute to the objective function of the parametric problem, given the decisions of x_{i-1} and x_i. Using the convention that $\underline{v_1} = \overline{v_{n+1}} = 1$ and $x_0 = x_{n+1} = 1$, our value functions are:

$$\begin{aligned} V_1(x_1) &= \max_{\substack{x_i \in \{0,1\}: \\ i=2,\ldots n}} \sum_{i=1}^{n} (r_i - \delta) u_i x_i \cdot (\underline{v_i})^{x_{i-1}} \cdot (\overline{v_i})^{x_{i+1}}, \\ V_i(x_{i-1}, x_i) &= \max_{\substack{x_j \in \{0,1\}: \\ j=i+1,\ldots,n}} \sum_{j=i}^{n} (r_j - \delta) u_j x_j \cdot (\underline{v_j})^{x_{j-1}} \cdot (\overline{v_j})^{x_{j+1}}, \\ V_n(x_{n-1}, x_n) &= (r_n - \delta) u_n x_n \cdot (\underline{v_n})^{x_{n-1}}. \end{aligned} \quad (11)$$

Based on the value functions, we have $h(\delta) = \max\{V_1(0), V_1(1)\}$. In Appendix D, we use dynamic programming to compute $V_1(0)$ and $V_1(1)$ efficiently for any value of $\delta > 0$.

Lemma 10. *Suppose we have a synergy path and multiplicative synergy weights. We can compute an optimal assortment in $O(n^3 \log^2 n)$ operations.*

The issue of having to remember the decisions for more than one product carries over to the synergy tree. If there is synergy going from a vertex to its parent and vice versa, then we need the values of x_{p_i}, x_i, and x_c for all $c \in C_i$ in order to determine the total preference weight of product i. We cannot consider the decision of each child vertex independently to measure their marginal contribution to the preference weight of product i. We present results on two special cases of synergy trees. First, we consider an out-tree, which we define as a synergy tree with $v_i^{p_i} \neq 1$ and $v_{p_i}^i = 1$ for all $i \neq$ root. Returning to our earlier example, a generic-brand product creates synergy towards the entry-level products of national brands, which in turn create synergy for higher-end products in their respective brands. There is no synergy in the reverse direction. Second, we consider a tree that has at most d children per vertex. In this case, each product only creates synergy with a small number of other products.

Lemma 11. *Suppose G is a synergy tree with multiplicative synergy weights. If G is an out-tree, then we can compute an optimal assortment in $O(n^3 \log^2 n)$ operations. If G has at most d children per vertex, then we can compute an optimal assortment in $O(2^{3d} n^3 (d^2 + \log^2 n))$ operations.*

In the above lemma, if $d = O(\log n)$, then our algorithm has runtime polynomial in n, specifically $O(n^6 \log^2 n)$. If $d = O(1)$, then our algorithm computes the optimal assortment in $O(n^3 \log^2 n)$ operations and has the same runtime as synergy trees with additive synergy weights.

6. Conclusion

One future direction is to study parameter estimation and validity of this model. The log-likelihood function of our model is not concave, so parameter estimation would have to rely on local optimal solutions if the maximum likelihood estimator is used. Under this limitation, it would be interesting to obtain real-world data and test whether synergistic MNL performs well in predicting customer purchases and computing near-optimal assortments.

We extended synergy to MMNL, but it would be interesting to see how synergy can be incorporated in other widely-used choice models, such as the nested logit model. Synergy could be created between nests or between products, and each type of synergy could have different interpretations. The incorporation of synergy into other choice models and the construction of algorithms for the underlying assortment optimization problem could lead to choice models that are more useful in applications.

Acknowledgement: We thank Jacob Feldman for his help in developing the proof of Theorem 2.

Funding: This work was supported by NSERC-PGS-D and Cornell University.

References

[1] Blanchet, J., Gallego, G. and Goyal, V. [2016], ‘A markov chain approximation to choice modeling’, *Oper. Res.* **64**(4), 886–905.

[2] Bodlaender, H. L. [1996], ‘A linear-time algorithm for finding tree-decompositions of small treewidth’, *SIAM J. on Comput.* **25**(6), 1305–1317.

[3] Bodlaender, H. L., Gilbert, J. R., Hafsteinsson, H. and Kloks, T. [1995], ‘Approximating treewidth, pathwidth, frontsize, and shortest elimination tree’, *J. Algorithms* **18**(2), 238–255.

[4] Brandstadt, A., Spinrad, J. P. et al. [1999], *Graph classes: a survey*, Vol. 3, SIAM.

[5] Bront, J. J. M., Méndez-Díaz, I. and Vulcano, G. [2009], ‘A column generation algorithm for choice-based network revenue management’, *Oper. Res.* **57**(3), 769–784.

[6] Désir, A., Goyal, V. and Zhang, J. [2014], ‘Near-optimal algorithms for capacity constrained assortment optimization’, *Available at SSRN 2543309*.

[7] Feldman, J., Paul, A. and Topaloglu, H. [2019], ‘Technical note—assortment optimization with small consideration sets’, *Oper. Res.*.

[8] Feldman, J. and Topaloglu, H. [2017], ‘Revenue management under the markov chain choice model’, *Oper. Res.* **65**(5), 1322–1342.

[9] Feng, G., Li, X. and Wang, Z. [2015], ‘Analysis of discrete choice models: A welfare-based framework’, *arXiv preprint arXiv:1503.01854*.

[10] Gallego, G., Ratliff, R. and Shebalov, S. [2014], ‘A general attraction model and sales-based linear program for network revenue management under customer choice’, *Oper. Res.* **63**(1), 212–232.

[11] Hanks, A. S., Just, D. R. and Wansink, B. [2012], ‘Trigger foods: The influence of “irrelevant” alternatives in school lunchrooms’, *Agric. and Resour. Econ. Rev.* **41**(1), 114–123.

[12] Honhon, D., Jonnalagedda, S. and Pan, X. A. [2012], ‘Optimal algorithms for assortment selection under ranking-based consumer choice models’, *Manuf. Serv. Oper. Manag.* **14**(2), 279–289.

[13] Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H. and Wang, Y. [2014], ‘The unconstrained binary quadratic programming problem: a survey’, *J. Comb. Optim.* **28**(1), 58–81.

[14] Li, H. and Huh, W. T. [2011], ‘Pricing multiple products with the multinomial logit and nested logit models: Concavity and implications’, *Manuf. Serv. Oper. Manag.* **13**(4), 549–563.

[15] Luce, R. D. [1959], *Individual Choice Behavior a Theoretical Analysis*, John Wiley and sons.

[16] Mahajan, S. and Van Ryzin, G. [2001], ‘Stocking retail assortments under dynamic consumer substitution’, *Oper. Res.* **49**(3), 334–351.

[17] McFadden, D. and Train, K. [2000], ‘Mixed MNL models for discrete response’, *J. Appl. Économ.* **15**(5), 447–470.

[18] McFadden, D. et al. [1973], ‘Conditional logit analysis of qualitative choice behavior’.

[19] Padberg, M. [1989], ‘The boolean quadric polytope: some characteristics, facets and relatives’, *Math. Program.* **45**(1), 139–172.

[20] Radzik, T. [1998], Fractional combinatorial optimization, *in* ‘Handbook of Combinatorial Optimization’, Springer, pp. 429–478.

[21] Rusmevichientong, P., Shmoys, D. B., Tong, C. and Topaloglu, H. [2014], ‘Assortment optimization under the multinomial logit model with random choice parameters’, *Prod. Oper. Manag.* **23**(11), 2023–2039.

[22] Simonson, I. [1989], ‘Choice based on reasons: The case of attraction and compromise effects’, *J. Consum. Res.* **16**(2), 158–174.

[23] Simonson, I. [1999], ‘The effect of product assortment on buyer preferences’, *J. Retail.* **75**(3), 347–370.

[24] Simonson, I. and Tversky, A. [1992], ‘Choice in context: Tradeoff contrast and extremeness aversion’, *J. Mark. Res.* **29**(3), 281–295.

[25] Thurstone, L. L. [1927], ‘A law of comparative judgment.’, *Psychol. Rev.* **34**(4), 273.

[26] Train, K. [2003], *Discrete choice methods with simulation*, Cambridge university press.

[27] Wald, J. A. and Colbourn, C. J. [1983], ‘Steiner trees, partial 2–trees, and minimum IFI networks’, *Networks* **13**(2), 159–167.

[28] Williamson, D. P. and Shmoys, D. B. [2011], *The design of approximation algorithms*, Cambridge university press.