Assortment Optimization under the Multinomial Logit Model with Product Synergies

Venus Lo, Huseyin Topaloglu

School of Operations Research and Information Engineering, Cornell University, Ithaca, New York 14853,
vhl8@cornell.edu, topaloglu@orie.cornell.edu

Abstract

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

Keywords: Assortment optimization, synergy, multinomial logit model, choice modeling

1. Introduction

Modeling customers' purchasing behaviours is critical in retail operations because it affects the retailer's decisions on what to sell in order to maximize his profit. Early work in inventory management assumed that demands are independent of the assortment being offered. Later, choice models in revenue management literature recognized that product demand might decrease if customers have more options. However, most choice models do not allow for synergistic 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 pairs 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 synergistic preference 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 introduced by Feng et al. [7]. The authors introduced a welfare function, which takes as input the utilities of an assortment and outputs its welfare as a real number. Choice 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 do not change if all utilities increased by the same amount. Convexity ensures that a high-utility product improves welfare more than multiple low-utility products.

Feng et al. [7] defined operations on welfare-based choice models which create new welfare-based choice models. One resulting model is the basis for our synergistic MNL model. In their model, the total increase in preference weight from synergy (if any) between two products is equal to a weighted geometric mean of their base preference weights. In our model, we allow synergy weights to be any nonnegative values. They focused on defining a new class of models, and did not discuss 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 into the assortment (Simonson [19]). In one study, a retailer selling a bread maker introduced a second, overpriced bread maker to his store. The demand for the original bread maker increased even though the assortment became larger. Products can also trigger a change in preference; Hanks et al. [9] 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 is the sum of the mean utility plus a random noise. MNL (Luce [12], McFadden et al. [14]) is the most famous model in this class, and the random noise for each product follows an independent standard Gumbel distribution. The preference weight of a product is the exponential of its mean utility, and the purchase probability of a product is proportional to its preference weight in the offered 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 synergistic model follows MNL such that purchase probabilities are proportional to preference weights, but it is not subject to IIA because synergy can increase the weights. We discuss other extensions to MNL, and details can be found in Train [21].

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 effect into the nested logit model might sacrifice its utility-maximizing property. The nested logit model has since been extended into the d-level nested logit model (Li and Huh [11]) and to the generalized extreme value model (Train [21]), where products belong to multiple nests.

In the mixed logit model (MMNL) studied by by Bront et al. [5] and Rusmevichientong et al. [18], different customer types have different sets of preference weights. McFadden and Train [13] showed that MMNL can approximate any random utility choice model. Désir et al. [6] gave a fully-polynomial time approximation scheme to compute a $(1-\epsilon)$-optimal solution.

The random noises of utilities do not have to follow the Gumbel distribution. In the probit model first described by Thurstone [20], the noises follow a multivariate normal distribution and the model does not exhibit the IIA property. Choice models outside the class of random utility models include the Markov Chain choice model (Blanchet et al. [1]) and the non-parametric choice model. Paul et al. [16] studied a special case of the non-parametric choice model, where customers consider at most two products.

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. Padberg [15] studied the graph and the related integer program, and identified cases where the convex hull of the feasible region is integral. We use this graph to specify cases of the synergistic MNL model which can be solved efficiently. A survey of quadratic binary programming can be found in Kochenberger et al. [10].

Our linear program for finding the optimal assortment under a synergy path is similar to the sales-based linear program (SBLP) described by Gallego et al. [8]. They studied a generalization of the network revenue management problem under MNL, where the retailer offers assortments over time subject to resource constraints. We use a similar strategy by taking the dual of the appropriate dynamic program in linear program form.

Organization: In Section 2, we describe the synergistic MNL model and its parametrized form, 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 parametrized problem. In Section 4, we present a linear program which recovers the optimal assortment when we have a synergy path. We conclude in Section 5.

2. General Model and NP-hardness

We describe the general model and the parametrized problem, and show that the general model is NP-hard.

2.1. The Model

The retailer has access to n products, denoted $N = \{1, \dots, n\}$. 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. The retailer can offer negative-profit products if they boost the preferences on high-profit products. The retailer chooses an assortment from N to sell in his store, 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 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 offered in 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 $u_i + \sum_{j \neq i} v_i^j x_j$.

Upon seeing assortment x, a customer makes her purchase decision according to the MNL model. If product i is offered, then her 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, and we scale the preference weight of the no-purchase option to 1 without loss of generality. Hence, if assortment x is offered, the probability that she buys 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 together. 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:

$$\bar{r}_{i,j} = \frac{r_i v_i^j + r_j v_j^i}{v_i^j + v_j^i}, \quad \text{and} \quad v_{i,j} = v_i^j + v_j^i.$$

Using this notation, our assortment optimization problem is $\max_{x \in \{0,1\}^n} \Pi(x)$, which 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 Parametrized Problem

We apply a standard parametrization technique for fractional combinatorial problems (Radzik [17]). 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 parametrized problem, with value $h(\delta)$, 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 value $h(\delta)$ for any $\delta \geq 0$. Since $h(0) > 0$ and $h(\delta)$ is monotone decreasing to $-\infty$, one method to find δ^* from Claim 1 is via Newton's method. By using the techniques that transform a quadratic binary program to an integer program (Padberg [15]), we can show that Newton's method finds δ^* in $O(n^4 \log^2 n)$ iterations of computing $h(\delta)$. Hence, we can solve Problem (1) in polynomial time if we can solve Problem (2) in polynomial time.

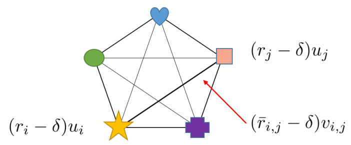

Figure 1: Synergy graph on 5 products, with synergy between each pair of products

Unfortunately, Problem (2) is a special case of quadratic binary programming, which is NP-hard in general (Kochenberger et al. [10]). Problem (1) is also NP-hard, and we prove the next theorem via a reduction from the maximum independent set problem.

Theorem 2. *The assortment optimization problem under the synergistic MNL model 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 (Padberg [15]). 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 $v_{i,j} > 0$. Hence $V = N$ and $E = \{ij : \ v_{i,j} > 0\}$. We call this graph the synergy graph (e.g. Figure 1).

If our synergy graph has at least two components, then Problem (2) can be broken up into smaller sub-problems containing only the products in the component. Without loss of generality, we assume 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 present dynamic programs to solve Problem (2) when the synergy graph is a path, a tree, or has low treewidth.

3.1. A Synergy Path

Suppose the synergy graph is a path. For example, consider ice cream, which improves in taste as the fat content increases. The presence of a compromise ice cream, which balances the fat content and taste, can increase demands for a better-tasting, fattier ice cream and a healthier, less tasty ice cream (Simonson [19]).

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 $i = 1, \ldots, n-1$ 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 previous product whose state we have to consider. We attribute the synergy effect 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 $h(\delta) = V_1$:

$$\begin{aligned}
V_1 &= \max_{x_1,\ldots,x_n} \sum_{i=1}^{n} (r_i - \delta)\, u_i x_i + \sum_{i=1}^{n-1} (\bar{r}_i - \delta)\, v_i x_i x_{i+1}, \\
V_i(x_{i-1}) &= \max_{x_i,\ldots,x_n} (\bar{r}_{i-1} - \delta) v_{i-1} x_{i-1} x_i + \\
&\qquad \sum_{j=i}^{n} (r_j - \delta) u_j x_j + \sum_{j=i}^{n-1} (\bar{r}_j - \delta) v_j x_j x_{j+1}.
\end{aligned}$$

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 contribution 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:

$$\begin{aligned}
V_1 &= \max_{x_1 \in \{0,1\}} (r_1 - \delta) x_1 + V_2(x_1), &(3)\\
V_i(x_{i-1}) &= \max_{x_i \in \{0,1\}} R_i^{\delta}(x_{i-1}) \cdot x_i + V_{i+1}(x_i), \quad \forall i = 2, \ldots, n, \\
V_{n+1}(x_n) &= 0.
\end{aligned}$$

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 effect 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 effect 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 n products, and $O(1)$ states and $O(1)$ decisions at each state per product. We could run Newton's method to find δ^*. 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 one linear programming computation.

3.2. A Synergy Tree

Suppose the synergy graph is a tree. For example, an inexpensive, generic-brand product creates synergy with the entry-level products of several national brands. In turn, the entry-level product of each national brand creates synergy with the higher-quality products of its brand.

Given a product i, let p_i represent its parent and C_i represent the set of its children. To be consistent with our earlier notation, we index the products so that the indices satisfy $p_i < i < c$ for $c \in C_i$. If product i is a leaf of the synergy tree, then $C_i = \emptyset$. Also, let T_i denote the set of vertices in the subtree rooted at product i. Since a product only creates synergy with either its parent or its children, we can rewrite Problem (2) as:

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

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 vertices 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:

$$\begin{aligned}
V_{\text{root}} &= \max_{x_1,\ldots,x_n} \sum_{i=1}^{n} (r_i - \delta)\, u_i x_i \\
&\qquad + \sum_{i=1}^{n} \sum_{c \in C_i} (\bar{r}_{i,c} - \delta)\, v_{i,c} x_i x_c, \\
V_i(x_{p_i}) &= \max_{x_j : j \in T_i} (\bar{r}_{p_i,i} - \delta) v_{p_i,i} x_{p_i} x_i \\
&\qquad + \sum_{j \in T_i} (r_j - \delta) u_j x_j \\
&\qquad + \sum_{j \in T_i} \sum_{c \in C_j} (\bar{r}_{j,c} - \delta)\, v_{j,c} x_j x_c.
\end{aligned}$$

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 contribution with its parent given the decision x_{p_i} and its base contribution, $R_i^{\delta}(x_{p_i}) := (\bar{r}_{p_i,i} - \delta) v_{p_i,i} x_{p_i} + (r_i - \delta) u_i$. 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

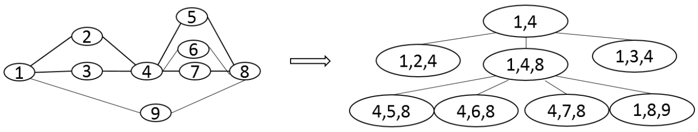

Figure 2: On the left, we have a synergy graph on 9 products. On the right, we have a tree decomposition of the synergy graph. A vertex on the tree represents a subset of vertices on the original graph.

$R_i^\delta(x_{p_i})$ 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. We can find δ^* using Newton's method, or 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.

The dynamic program that we construct 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 graphs 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 (Williamson and Shmoys [23]). Details of this extension are deferred to the appendix, and an example of a tree decomposition is depicted in Figure 2.

Theorem 3. *For a synergy graph G, suppose we are given a tree decomposition with width t and $O(n)$ vertices. Then it takes $O(2^t n)$ 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 $O(2^t n^5 \log^2 n)$.*

For example, if the synergy graph has no K_4 minor so that it is a series-parallel graph, then it has treewidth $t^* = 2$ (Wald and Colbourn [22], Brandstadt et al. [4]). In our theorem, we assume 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 at most $O(n)$ vertices (Bodlaender [2]), with runtime polynomial in n but exponential in t^*. On the other hand, there exists algorithms to find a tree decomposition with width $t = O(t^* \log n)$ in polynomial runtime (Bodlaender et al. [3]), which increases the runtime of the dynamic program by a factor of n 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 immediately reveals the optimal assortment. This allows a practitioner to take advantage of linear program solvers and avoids solving Problem (2) repeatedly.

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 value of this linear program is $h(\delta)$.

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

Let $\mathcal{V}$ denote the set of constraints above. Claim 1 tells us that the linear program has optimal 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:

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

Lemma 4. *The optimal 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 can 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 value, strong duality guarantees that its dual is feasible and has optimal value δ^*. By taking the dual of Problem (6) and making the appropriate transformation of variables, we obtain:

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

Problem (7) looks similar to the SBLP in Gallego et al. [8], 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 are interested in solutions satisfying the following condition.

Condition 1. *We are interested in solutions (y, w) to Problem (7) of the following form:*

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 $y_i/u_i = y_0$.*

Our goal is to show that extreme points of Problem (7) satisfies condition 1. Moreover, there is a one-to-one correspondence between extreme points and assortments. Lemma 5 maps assortments to solutions satisfying condition 1.

Lemma 5. *Given an assortment x, there exists a solution (y, w) which is feasible to Problem (7) with objective value equal to $\Pi(x)$. Furthermore, (y, w) satisfies condition 1.*

We now consider the reverse direction, and show that solutions (y, w) that satisfy condition 1 map to assortments. Our main theorem proves that every extreme point solution satisfies condition 1.

Lemma 6. *Suppose (y, w) is feasible to Problem (7) and satisfies condition 1. If $x_i = \mathbb{1}[y_i > 0]$, then assortment x has expected profit equal to the objective value of (y, w).*

Theorem 7. *Every extreme point of the polytope in Problem (7) satisfies condition 1. Hence, given an extreme point optimal solution (y^*, w^*) to Problem (7), we can recover an optimal assortment x^* by taking $x_i^* = \mathbb{1}[y_i^* > 0]$.*

In summary, computing the optimal assortment of the synergistic MNL model when we have a synergy path can be achieved by solving a simple linear program. This allows us to avoid running the dynamic program altogether.

5. Conclusion

We presented a synergistic version of MNL for assortment optimization. Synergy is created if the preference weight of at least one product in a pair of products increases when both are present in the assortment. The optimal assortment can be computed efficiently when the synergy graph is a path or a tree. Furthermore, when we have a synergy path, the optimal assortment can be found by solving a simple linear program.

In terms of future research, it may be possible to extend our model to encompass cannibalization across products in addition to synergy. The natural approach would be to set the synergy weights $v_i^j < 0$, but doing so might violate the underlying assumptions that lets us parametrize our assortment optimization problem. A careful analysis would be needed to determine whether our lemmas and theorems are still valid.

Parameter estimation and validity of this model are also interesting future directions. 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 the synergistic MNL model performs well with estimating purchase probabilities and computing optimal assortments.

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

References

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

[2] Bodlaender, H. L. [1996], 'A linear-time algorithm for finding tree-decompositions of small treewidth', *SIAM Journal on computing* **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', *Operations Research* **57**(3), 769–784.

[6] Désir, A., Goyal, V. and Zhang, J. [2014], 'Near-optimal algorithms for capacity constrained assortment optimization'.

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

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

[9] Hanks, A. S., Just, D. R. and Wansink, B. [2012], 'Trigger foods: the influence of "irrelevant" alternatives in school lunchrooms'.

[10] 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', *Journal of Combinatorial Optimization* **28**(1), 58–81.

[11] Li, H. and Huh, W. T. [2011], 'Pricing multiple products with the multinomial logit and nested logit models: Concavity and implications', *Manufacturing & Service Operations Management* **13**(4), 549–563.

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

[13] McFadden, D. and Train, K. [2000], 'Mixed MNL models for discrete response', *Journal of applied Econometrics* **15**(5), 447–470.

[14] McFadden, D. et al. [1973], 'Conditional logit analysis of qualitative choice behavior'.

[15] Padberg, M. [1989], 'The boolean quadric polytope: some characteristics, facets and relatives', *Mathematical programming* **45**(1), 139–172.

[16] Paul, A., Topaloglu, H. and Feldman, J. [2016], 'Assortment optimization for choosy customers'.

[17] Radzik, T. [1998], Fractional combinatorial optimization, *in* 'Handbook of combinatorial optimization', Springer, pp. 429–478.

[18] Rusmevichientong, P., Shmoys, D., Tong, C. and Topaloglu, H. [2014], 'Assortment optimization under the multinomial logit model with random choice parameters', *Production and Operations Management* **23**(11), 2023–2039.

[19] Simonson, I. [1999], 'The effect of product assortment on buyer preferences', *Journal of Retailing* **75**(3), 347–370.

[20] Thurstone, L. L. [1927], 'A law of comparative judgment.', *Psychological review* **34**(4), 273.

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

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

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

Appendix A. Graphs with Low Treewidth

We present a dynamic program to solve the parametrized problem, Problem (2), when we are given a tree decomposition of the synergy graph with width t. For a synergy graph $G = (V, E)$, let (X, T) denote a tree decomposition of G, where $T = (V', E')$ is a tree. Furthermore, $X = \{X_1, \ldots, X_K\}$ is a collection of subsets of vertices in V, such that each vertex $k \in V'$ is associated with $X_k \subseteq V$, and X satisfies three properties:

1. Every vertex of G is in some X_k: $\cup_{k \in V'} X_k = V$.
2. For every $ij \in E$, there exists $k \in V'$ such that $i, j \in X_k$.
3. Suppose $m \in V'$ is on the unique k-ℓ path in T. Then $X_k \cap X_\ell \subseteq X_m$.

The width of (X, T) is defined as $t = \max_{k \in V'} |X_k| - 1$, and the treewidth t^* of G is the minimum width of all its tree decompositions.

Select a vertex of V' to be the root of T. Since we work with subsets of vertices, we use set notation rather than vector notation in this appendix. We use p_k to denote the parent of k on T and C'_k to denote the set of children of k. Let T_k denote the set of products which are in some X_ℓ, where ℓ is a vertex in the subtree rooted at k. Recursively, $T_k = X_k$ if k is a leaf and $T_k = (\cup_{c \in C'_k} T_c) \cup X_k$ if k is not a leaf. Notice that $C'_k \subseteq V'$ and $X_k \subseteq T_k \subseteq V$, so they are subsets of vertices on different graphs.

Define $R^\delta(S)$ as the contribution to Problem (2) from the vertices and edges in the subgraph of G induced by products in S. In other words,

$$R^\delta(S) = \sum_{i=1}^{n} (r_i - \delta) u_i \cdot \mathbb{1}[i \in S] + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (\bar{r}_{i,j} - \delta) v_{i,j} \cdot \mathbb{1}[i, j \in S].$$

Our value function is defined on the tree rather than the original graph. At a vertex $k \in V'$, our value function considers the maximum contribution to Problem (2) that we can achieve by offering products within $S \subseteq T_k$, given that we include $S_{p_k} \subseteq X_{p_k}$ in the assortment. A tree decomposition allows product i to be in X_{p_k} and T_k, so we must ensure that such a product is included in S if and only if it is included in S_{p_k}. If so, we say that $S \subseteq T_k$ is feasible with respect to S_{p_k}. More precisely, given $S_{p_k} \subseteq X_{p_k}$, the feasible subsets of T_k are:

$$F_k^T(S_{p_k}) = \{S \subseteq T_k \ : \ S \cap X_{p_k} = S_{p_k} \cap T_k\}.$$

The value function at $k \in V'$ is:

$$V_k(S_{p_k}) = \max_{S \in F_k^T(S_{p_k})} R^\delta(S).$$

To formulate a dynamic program, we want to restrict our current decision to $S \subseteq X_k$ rather than $S \subseteq T_k$. Similar to the definition of $F_k^T(S_{p_k})$, we want to define the feasible subsets that we may offer from X_k when we are given S_{p_k}. Let $F_k^X(S_{p_k})$ be the collection of feasible subsets that we may offer:

$$F_k^X(S_{p_k}) = \{S \subseteq X_k : \ S \cap X_{p_k} = S_{p_k} \cap X_k\}.$$

The next lemma proves we can compute our value function via dynamic programming using the definition of $F_k^X(S_{p_k})$ and $R^\delta(S)$.

Lemma 8. *The value function $V_k(S_{p_k})$ can be written as:*

$$V_k(S_{p_k}) = \max_{S_k \in F_k^X(S_{p_k})} R^\delta(S_k) + \sum_{c \in C'_k} V_c(S_k) - R^\delta(S_k \cap X_c).$$

The intuition behind the dynamic program is that the decision made at vertex ℓ, where ℓ is in the subtree rooted at k, is feasible with respect to S_{p_k}. This is because a product $i \in X_{p_k} \cap X_\ell$ must also be in X_k, as well as all X_m where m is on the k-ℓ path in the tree. Hence we would make the same decision regarding product i at each tree vertex from k to ℓ. To prove the correctness of the dynamic program, we check that we do not double-count a product or the synergy between products since we may observe a graph vertex or edge in multiple X_ℓ. Furthermore, if $ij \in E$ and $i, j \in T_k$, we check that both i, j are contained in some X_ℓ to ensure that their synergy contribution is recorded. We first use Lemma 8 to prove the runtime analysis in Theorem 3, and then revisit Lemma 8 to prove the correctness of the dynamic program.

Proof of Theorem 3. For each vertex on the tree, the state space is the collection of subsets of X_{p_k} and has size $O(2^t)$ since $|X_{p_k}| \leq t - 1$. We consider including $S \in F_k^X(S_{p_k})$ in the assortment, so we consider at most $O(2^t)$ decisions. Our dynamic program is computed from the leaves of T up to the root, and the tree has $O(n)$ vertices. Hence Problem (2) can be solved in $O(2^t n)$ operations.

If we use Newton's algorithm to find the fixed point δ^*, then we need to solve Problem (2) at most $O(n^4 \log^2 n)$ times. Hence the total number of operations to find the optimal assortment is $O(2^t n^5 \log^2 n)$. □

In order to prove Lemma 8, we use a series of claims to show that the decisions for products in different subtrees T_c are independent when we are given the decision to include $S_k \subseteq X_k$, where c is a child of k. We first show that a product in T_k must either be in X_k or in exactly one T_c so that we can split the decision over vertex k and its subtrees. Similarly, a pair of products with positive synergy must either both be in X_k or both be in exactly one T_c.

Claim 9. *Consider vertex $k \in V'$ and two of its children $c, c' \in C'_k$. Then no products are in both T_c and $T_{c'}$ unless they are also in X_k: $(T_c \backslash X_k) \cap (T_{c'} \backslash X_k) = \emptyset$.*

Proof. Suppose by contradiction that there exists some product i such that $i \in (T_c \backslash X_k) \cap (T_{c'} \backslash X_k)$. Then there exists some $\ell \in V'$ in the subtree rooted at c, such that

$i \in X_\ell$. Similarly, there exists $\ell' \in V'$ in the subtree rooted at c' such that $i \in X_{\ell'}$. The path from ℓ to ℓ' contains k, so $X_\ell \cap X_{\ell'} \subseteq X_k$. This implies $i \in X_k$ and contradicts $i \in (T_c \backslash X_k) \cap (T_{c'} \backslash X_k)$. □

Claim 10. *Suppose there exists an edge $ij \in E$ such that $i, j \in T_k$. Then either $i, j \in X_k$, or at least one of products i, j is not in X_k and there exists an unique $c \in C'_k$ such that $i, j \in T_c$.*

Proof. By definition of tree decomposition, there exists some $\ell \in V'$ such that $i, j \in X_\ell$. If $\ell = k$, then the first condition is satisfied. If $\ell \neq k$ but is a vertex in the subtree rooted at k, then ℓ must be in the subtree rooted at a vertex $c \in C'_k$, and $i, j \in T_c$. Furthermore, the uniqueness of c follows from observing that at least one of i, j is not in X_k and applying Claim 9. This satisfies the second condition. We show that such a vertex ℓ must exist in the subtree rooted at k.

Suppose by contradiction that ℓ is not a vertex in the subtree rooted at k. There must exists vertices ℓ', ℓ'' in the subtree rooted at k, $\ell' \neq \ell''$, such that $i \in X_{\ell'} \backslash X_{\ell''}$ and $j \in X_{\ell''} \backslash X_{\ell'}$ since $i, j \in T_k$. The unique path from ℓ to ℓ' on T contains k, and similarly the path from ℓ to ℓ'' contains k. This implies that $X_\ell \cap X_{\ell'} \subseteq X_k$, so $i \in X_k$. Similarly, $X_\ell \cap X_{\ell''} \subseteq X_k$, so $j \in X_k$. This contradicts our assumption that no vertex ℓ exists in the subtree rooted at k such that $i, j \in X_\ell$. □

The next claim breaks up $R^\delta(S)$ over X_k and T_c for $c \in C'_k$, so that we can count the contributions from X_k and T_c separately.

Lemma 11. *At vertex $k \in V'$, suppose we have $S \subseteq T_k$. Then*

$$R^\delta(S) = R^\delta(S \cap X_k) + \sum_{c \in C'_k} R^\delta(S \cap T_c) - R^\delta(S \cap T_c \cap X_k).$$

Proof. By definition of $R^\delta(S)$, we have

$$\begin{aligned} R^\delta(S) = &\sum_{i=1}^{n} (r_i - \delta) u_i \cdot \mathbb{1}[i \in S] \\ &+ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (\bar{r}_{i,j} - \delta) v_{i,j} \cdot \mathbb{1}[i, j \in S]. \end{aligned}$$

First, we consider the sum over the vertices of G. For a fixed product $i \in T_k$, we have,

$$\begin{aligned} \mathbb{1}[i \in S] = &\ \mathbb{1}[i \in S \cap X_k] \\ &+ \sum_{c \in C'_k} \mathbb{1}[i \in S \cap T_c] - \mathbb{1}[i \in S \cap T_c \cap X_k]. \end{aligned} \tag{A.1}$$

If $i \in X_k$, then $\mathbb{1}[i \in S \cap T_c] = \mathbb{1}[i \in S \cap T_c \cap X_k]$, so we count the contribution of product i when it appears in X_k. If $i \notin X_k$, then Claim 9 tells us that there exists an unique $c \in C'_k$ such that $i \in T_c$.

Next, we consider the sum over the edges of G. Suppose $ij \in E$ so that $v_{i,j} \neq 0$, then the following is true:

$$\begin{aligned} \mathbb{1}[i, j \in S] = &\ \mathbb{1}[i, j \in S \cap X_k] \\ &+ \sum_{c \in C'_k} \mathbb{1}[i, j \in S \cap T_c] - \mathbb{1}[i, j \in S \cap T_c \cap X_k]. \end{aligned} \tag{A.2}$$

If $i, j \in X_k$, then $\mathbb{1}[i, j \in S \cap T_c] = \mathbb{1}[i, j \in S \cap T_c \cap X_k]$ and we count the synergy between the two products in X_k. Otherwise, Claim 10 implies that there is an unique c such that $i, j \in T_c$.

We can expand $R^\delta(S)$ and apply equalities (A.1) and (A.2). Furthermore, recall that $v_{i,j} = 0$ if $ij \notin E$, so equality (A.2) holds true for all pairs i, j if we multiply both sides by $v_{i,j}$. Grouping the terms by their indicator functions, we see that $R^\delta(S) = R^\delta(S \cap X_k) + \sum_{c \in C'_k} R^\delta(S \cap T_c) - R(S \cap T_c \cap X_k)$. □

Finally, we prove that every subset in $F^T_k(S_{p_k})$ can be written as the union of $S_k \in F^X_k(S_{p_k})$ and $S_c \in F^T_c(S_k)$ for $c \in C'_k$, and vice versa.

Claim 12. *Suppose we are given the decision $S_{p_k} \subseteq X_{p_k}$ and consider a subset of products $S \subseteq F^T_k(S_{p_k})$. Define $S_k = S \cap X_k$ and $S_c = S \cap T_c$ for $c \in C'_k$. Then $S_k \in F^X_k(S_{p_k})$ and $S_c \in F^T_c(S_k)$.*

Proof. Since $S \in F^T_k(S_{p_k})$, we know that

$$S \cap X_{p_k} = S_{p_k} \cap T_k.$$

We intersect both sides with X_k to obtain:

$$S \cap X_{p_k} \cap X_k = S_{p_k} \cap T_k \cap X_k.$$

Since $S_k = S \cap X_k$ and $X_k \subseteq T_k$, we have $S_k \cap X_{p_k} = S_{p_k} \cap X_k$. Hence $S_k \in F^X_k(S_{p_k})$. Finally, to show that $S_c \in F^T_c(S_k)$, observe:

$$S_c \cap X_k = S \cap T_c \cap X_k = S_k \cap T_c.$$

□

Claim 13. *Given $S_{p_k} \subseteq X_{p_k}$, suppose we have $S_k \in F^X_k(S_{p_k})$ and $S_c \in F^T_c(S_k)$ for all $c \in C'_k$. Let S be the union of these sets such that $S = (\cup_{c \in C'_k} S_c) \cup S_k$, then $S \in F^T_k(S_{p_k})$.*

Proof. We consider:

$$\begin{aligned} S \cap X_{p_k} = &\Big(S_k \cap X_{p_k}\Big) \\ &\cup \Big(\cup_{c \in C'_k} \big(S_c \cap X_{p_k}\big)\Big). \end{aligned} \tag{A.3}$$

First we show that $X_{p_k} \cap T_k = X_{p_k} \cap X_k$. The "$\supseteq$" direction is due to $T_k \supseteq X_k$. For the "$\subseteq$" direction, observe that for every $\ell \in V'$ in the subtree rooted at k, including $\ell = k$, we have $X_{p_k} \cap X_\ell \subseteq X_k$ by the third property of (X, T) being a tree decomposition. Since T_k is the union of all

X_ℓ, we have $X_{p_k} \cap T_k \subseteq X_k$. Hence $X_{p_k} \cap T_k = X_{p_k} \cap X_k$, which also implies that $S_{p_k} \cap T_k = S_{p_k} \cap X_k$.

For the first term in the right side of equality (A.3), we use $S_k \in F_k^X(S_{p_k})$ to obtain:

$$\begin{aligned} S_k \cap X_{p_k} &= S_{p_k} \cap X_k \\ &= S_{p_k} \cap T_k. \end{aligned}$$

For the second term, consider some $c \in C'_k$. We have:

$$\begin{aligned} S_c \cap X_{p_k} &= S_c \cap X_{p_k} \cap X_k \\ &= S_k \cap T_c \cap X_{p_k} \\ &= S_p \cap X_k \cap T_c. \end{aligned}$$

The first line is true since $S_c \cap X_{p_k} \subseteq T_k \cap X_{p_k} \subseteq X_k$. The second line uses $S_c \in F_c^T(S_k)$, and the third line uses $S_k \in F_k^X(S_{p_k})$. Finally, this second term is a subset of the first term, so:

$$S \cap X_{p_k} = S_{p_k} \cap T_k,$$

and hence $S \in F_k^T(S_{p_k})$ as required. □

We are now ready to prove Lemma 8.

Proof of Lemma 8. We prove the correctness of our dynamic program via induction. In the base case of a leaf, the dynamic program is correct because a leaf vertex does not have children.

At a non-leaf vertex $k \in V'$, assume that the dynamic program holds for all descendents in the subtree rooted at k on T. Apply Lemma 11:

$$\begin{aligned} V_k(S_{p_k}) &= \max_{S \in F_k^T(S_{p_k})} R^\delta(S) \\ &= \max_{S \in F_k^T(S_{p_k})} R^\delta(S \cap X_k) \\ &\quad + \sum_{c \in C'_k} R^\delta(S \cap T_c) - R^\delta(S \cap T_c \cap X_k) \end{aligned}$$

Claims 12 and 13 state that we can choose S_k from $F_k^X(S_{p_k})$ and then S_c from $F_c^T(S_k)$ such that $S = (\cup_{c \in C'_k} S_c) \cup S_k$, rather than S from $F_k^T(S_{p_k})$. This gives us:

$$\begin{aligned} V_k(S_{p_k}) &= \max_{S_k \in F_k^X(S_{p_k})} \Big(R^\delta(S_k) \\ &\quad + \sum_{c \in C'_k} \max_{S_c \in F_c^T(S_k)} \big(R^\delta(S_c) - R^\delta(S_c \cap X_k) \big) \Big) \\ &= \max_{S_k \in F_k^X(S_{p_k})} \Big(R^\delta(S_k) \\ &\quad + \sum_{c \in C'_k} \max_{S_c \in F_c^T(S_k)} R^\delta(S_c) - R^\delta(S_k \cap X_c) \Big) \\ &= \max_{S_k \in F_k^X(S_{p_k})} R^\delta(S_k) + \sum_{c \in C'_k} V_c(S_k) - R^\delta(S_k \cap X_c). \end{aligned}$$

The second line is true because $S_c \in F_c^T(S_k)$. The last line is true by observing that only the second term is dependent on the inner maximization, and applying the induction hypothesis to get $V_c(S_k) = \max_{S_c \in F_c^T(S_k)} R^\delta(S_c)$. □

Appendix B. Constructing the Sales-based Linear Program

In this section, we construct the SBLP in Problem (7), and show that the extreme points of its feasible region satisfy condition 1. As a result, we can compute the optimal assortment simply by solving a linear program, rather than going through multiple iterations of a dynamic program.

Problem (6) expands out to the following linear program when we implicitly set $\delta = V_1$:

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

By Lemma 4, the above linear program has optimal value δ^*. Substitute in the value of $V_{n+1}^0 = V_{n+1}^1 = 0$ and associate the following dual variables with the constraints relating products $i-1$ and i:

- z_i^{00} when we do not offer $i-1$ and i,
- z_i^{01} when we do not offer $i-1$ but offer i,
- z_i^{10} when we offer $i-1$ but do not offer i, and
- z_i^{11} when we offer both $i-1$ and i.

We define the above for $i = 1, \ldots, n$ with $z_1^{10} = z_1^{11} = 0$, because there is no profit nor synergy associated with the no-purchase option. The dual is:

$$\max \sum_{i=1}^{n} r_i u_i \left(z_i^{01} + z_i^{11} \right) + \sum_{i=1}^{n-1} \bar{r}_i v_i z_{i+1}^{11}$$

s.t.

$$\begin{aligned} & z_1^{00} + z_1^{01} + \sum_{i=1}^{n} u_i (z_i^{01} + z_i^{11}) + \sum_{i=1}^{n-1} v_i z_{i+1}^{11} = 1 \\ & z_i^{00} - z_{i-1}^{00} + z_i^{01} - z_{i-1}^{10} = 0 && \forall i = 2, \ldots, n \\ & z_i^{10} - z_{i-1}^{01} + z_i^{11} - z_{i-1}^{11} = 0 && \forall i = 2, \ldots, n \\ & z_i^{00}, z_i^{01}, z_i^{10}, z_i^{11} \geq 0 && \forall i = 1, \ldots, n \\ & z_1^{10} = z_1^{11} = 0. \end{aligned}$$

Lemma 14. *The dual to Problem (6) is equivalent to Problem (7).*

Proof. Suppose we have a solution z to the dual of Problem (6), then we can construct our solution (y, w) as:

$$\begin{aligned} y_0 &= z_1^{00} + z_1^{01}, \\ y_i &= u_i (z_i^{01} + z_i^{11}), && \forall i = 1, \ldots, n, \\ w_i &= v_i z_{i+1}^{11}, && \forall i = 1, \ldots, n-1. \end{aligned}$$

It is simple to verify that (y, w) is feasible to Problem (7) and has the same objective value as z.

In the reverse direction, if we have a solution (y, w) to Problem (7), we can construct a solution z as follows:

$$
\begin{aligned}
z_1^{00} &= x_0 - \frac{y_1}{u_1}, \\
z_i^{00} &= x_0 - \frac{y_i}{u_i} - \frac{y_{i-1}}{u_{i-1}} + \frac{w_{i-1}}{v_{i-1}}, && \forall i = 2, \ldots, n, \\
z_1^{01} &= \frac{y_1}{u_1}, \\
z_i^{01} &= \frac{y_i}{u_i} - \frac{w_{i-1}}{v_{i-1}}, && \forall i = 2, \ldots, n, \\
z_i^{10} &= \frac{y_{i-1}}{u_{i-1}} - \frac{w_{i-1}}{v_{i-1}}, && \forall i = 2, \ldots, n, \\
z_i^{11} &= \frac{w_{i-1}}{v_{i-1}}, && \forall i = 2, \ldots, n, \\
z_1^{10} &= z_1^{11} = 0.
\end{aligned}
$$

This solution z is feasible to the dual and has the same objective value as (y, w) in Problem (7). □

The key to proving Theorem 7 is to show that every feasible point with some y_i such that $0 < y_i/u_i < y_0$ can be written as a convex combination of two other feasible points, and hence it is not an extreme point.

Lemma 15. *Every extreme point to the feasible region in Problem (7) satisfies $y_i = 0$ or $y_i/u_i = y_0$ for all $i = 1, \ldots, n$.*

Proof. Suppose we have a solution (y, w) with at least one variable y_i such that $0 < y_i/u_i < y_0$. Define the set of non-tight variables as $J = \{y_i : 0 < y_i/u_i < y_0\} \cup \{w_i : 0 < w_i/v_i < y_0\}$.

Order the variables as $y_1, w_1, y_2, \ldots, y_{n-1}, w_{n-1}, y_n$. Define the set of indices $B = \{b_1, \ldots, b_{L+1}\}$ based on the tightness of the third constraint of Problem (7). Specifically, $b_1 = 0$, $b_{L+1} = n$, $b_1 < b_2 < \cdots < b_{L+1}$, and $i \in B$ if and only if $y_0 = y_i/u_i + y_{i+1}/u_{i+1} - w_i/v_i$.

We partition the variables into $L+1$ sets, $A_0, A_1, \ldots, A_L$. For A_ℓ such that $1 \le \ell \le L$, we define:

$$
\begin{aligned}
A_\ell =& \{y_i : \ b_\ell < i \le b_{\ell+1}\} \cup \{w_i : \ b_\ell < i < b_{\ell+1}\} \\
&\cup \{w_i : \ i = b_{\ell+1} \neq n, \ w_i/v_i = y_i/u_i < y_{i+1}/u_{i+1}\} \\
&\cup \{w_i : \ i = b_\ell \neq 0, \ w_i/v_i = y_{i+1}/u_{i+1} < y_i/u_i\}.
\end{aligned}
$$

Set A_0 contains the remaining indices of w:

$$
\begin{aligned}
A_0 =& \{w_i : \ i \in B, \ w_i/v_i = y_i/u_i = y_{i+1}/u_{i+1}\} \\
&\cup \{w_i : \ i \in B, \ w_i/v_i < \min\{y_i/u_i, y_{i+1}/u_{i+1}\}\}
\end{aligned}
$$

To create a new solution, we first define a scaling factor β as:

$$
\beta = \sum_{\substack{i:\, y_i \in J \cap A_\ell \\ \ell \text{ odd}}} u_i + \sum_{\substack{i:\, w_i \in J \cap A_\ell \\ \ell \text{ odd}}} v_i - \sum_{\substack{i:\, y_i \in J \cap A_\ell \\ \ell \neq 0 \text{ even}}} u_i - \sum_{\substack{i:\, w_i \in J \cap A_\ell \\ \ell \neq 0 \text{ even}}} v_i.
$$

The value of β can any real number. For some small ϵ satisfying $0 < \epsilon < 1/\beta$, the new solution $(\bar{y}, \bar{w})$ is:

$$
\bar{y}_i = \begin{cases} \frac{1}{1+\epsilon\beta} y_i & \text{if } y_i \notin J \\ \frac{1}{1+\epsilon\beta}(y_i + \epsilon u_i) & \text{if } y_i \in J \cap A_\ell \text{ for odd } \ell \\ \frac{1}{1+\epsilon\beta}(y_i - \epsilon u_i) & \text{if } y_i \in J \cap A_\ell \text{ for even } \ell \neq 0 \end{cases},
$$

$$
\bar{w}_i = \begin{cases} \frac{1}{1+\epsilon\beta} w_i & \text{if } w_i \notin J \text{ or } w_i \in A_0 \\ \frac{1}{1+\epsilon\beta}(w_i + \epsilon v_i) & \text{if } w_i \in J \cap A_\ell \text{ for odd } \ell \\ \frac{1}{1+\epsilon\beta}(w_i - \epsilon v_i) & \text{if } w_i \in J \cap A_\ell \text{ for even } \ell \neq 0 \end{cases},
$$

$$
\bar{y}_0 = y_0/(1 + \epsilon\beta).
$$

It is clear that $\bar{y}_0 + \sum_{i=1}^n \bar{y}_i + \sum_{i=1}^{n-1} \bar{w}_i = 1$, so we need to check the other constraints.

Case 1 - $\bar{y}_i, \bar{w}_i, \bar{y}_{i+1} \in A_\ell$ for odd ℓ: If $w_i \notin J$, then $\bar{w}_i = w_i/(1 + \epsilon\beta)$ and $\bar{y}_i \ge y_i/(1 + \epsilon\beta)$, so we have $\bar{y}_i/u_i \ge \bar{w}_i/v_i$. If $w_i \in J$ and $y_i \in J$, then

$$
\begin{aligned}
\bar{y}_i/u_i &= (y_i/u_i + \epsilon)/(1 + \epsilon\beta) \\
&\ge (w_i/v_i + \epsilon)/(1 + \epsilon\beta) = \bar{w}_i/v_i.
\end{aligned}
$$

If $w_i \in J$ and $y_i \notin J$, then $y_i/u_i = y_0$, which implies $y_i/u_i > w_i/v_i$. There exists small ϵ such that:

$$
\begin{aligned}
\bar{y}_i/u_i &= y_i/u_i/(1 + \epsilon\beta) \\
&\ge (w_i/v_i + \epsilon)/(1 + \epsilon\beta) = \bar{w}_i/v_i.
\end{aligned}
$$

Using the same argument, we can show that $\bar{y}_{i+1}/\bar{u}_{i+1} \ge \bar{w}_i/\bar{v}_i$. Finally, observe:

$$
\begin{aligned}
&\frac{\bar{y}_i}{u_i} + \frac{\bar{y}_{i+1}}{u_{i+1}} - \frac{\bar{w}_i}{v_i} \\
\le& \frac{1}{1 + \epsilon\beta} \left(\frac{y_i + \epsilon u_i}{u_i} + \frac{y_{i+1} + \epsilon u_{i+1}}{u_{i+1}} - \frac{\bar{w}_i}{v_i} \right) \\
\le& \frac{y_0}{1 + \epsilon\beta} = \bar{y}_0,
\end{aligned}
$$

where the last inequality is true for small enough ϵ because the third constraint of Problem (7) is not tight, by definition of B.

Case 2 - $\bar{y}_i, \bar{w}_i, \bar{y}_{i+1} \in A_\ell$ for even $\ell \neq 0$: If $w_i \in J$, then

$$
\begin{aligned}
\bar{y}_i/u_i &\ge (y_i/u_i - \epsilon)/(1 + \epsilon\beta) \\
&\ge (w_i/v_i - \epsilon)/(1 + \epsilon\beta) = \bar{w}_i/v_i.
\end{aligned}
$$

If $w_i \notin J$ and $y_i \notin J$, then $\bar{w}_i = w_i/(1 + \epsilon\beta)$ and $\bar{y}_i = y_i/(1 + \epsilon\beta)$, so we have $\bar{y}_i/u_i \ge \bar{w}_i/v_i$. If $w_i \notin J$ and $y_i \in J$, we must have $w_i = 0$ and $y_i > 0$, so there exists small ϵ such that

$$
\bar{y}_i/u_i = (y_i/u_i - \epsilon)/(1 + \epsilon\beta) \ge 0 = \bar{w}_i/v_i.
$$

Finally,

$$
\begin{aligned}
&\frac{\bar{y}_i}{u_i} + \frac{\bar{y}_{i+1}}{u_{i+1}} - \frac{\bar{w}_i}{u_i} \\
\le& \frac{1}{1 + \epsilon\beta} \left(\frac{y_i}{u_i} + \frac{y_{i+1}}{u_{i+1}} - \frac{w_i - \epsilon v_i}{v_i} \right) \\
\le& \frac{y_0}{1 + \epsilon\beta} = \bar{y}_0,
\end{aligned}
$$

where the last equality again holds for small enough ϵ because the third constraint of Problem (7) is not tight,by definition of B.

Case 3 - $\bar{y}_i \in A_\ell$ and $\bar{y}_{i+1} \in A_{\ell+1}$ for odd ℓ: First suppose $w_i \in A_0$. If $w_i/v_i = y_i/u_i = y_{i+1}/u_{i+1}$, then they are equal to y_0 because the third constraint of Problem (7) is tight when y_i, y_{i+1} are in different sets. Hence they are all not in J and only scaled, so $\bar{w}_i/v_i = \bar{y}_i/u_i = \bar{y}_{i+1}/u_{i+1}$ and all the constraints still hold with equality. Otherwise, $w_i/v_i < \min\{y_i/u_i, y_{i+1}/u_{i+1}\}$, so there is some small ϵ such that $w_i/v_i \leq \min\{y_i/u_i, y_{i+1}/u_{i+1}-\epsilon\}$, which ensures $\bar{w}_i/v_i \leq \min\{\bar{y}_i/u_i, \bar{y}_{i+1}/u_{i+1}\}$. For the third constraint, notice that $y_0 - y_i/u_i = y_{i+1}/u_{i+1} - w_i/v_i > 0$ implies $y_i \in J$. Similarly, $y_{i+1} \in J$. Hence the following equalities hold:

$$\begin{aligned}
\bar{y}_0 = \frac{y_0}{1+\epsilon\beta} &= \frac{1}{1+\epsilon\beta}\left(\frac{y_i}{u_i} + \frac{y_{i+1}}{u_{i+1}} - \frac{w_i}{u_i}\right) \\
&= \frac{1}{1+\epsilon\beta}\left(\frac{y_i + \epsilon u_i}{u_i} + \frac{y_{i+1} - \epsilon u_{i+1}}{u_{i+1}} - \frac{w_i}{u_i}\right) \\
&= \frac{\bar{y}_i}{u_i} + \frac{\bar{y}_{i+1}}{u_{i+1}} - \frac{\bar{w}_i}{u_i}.
\end{aligned}$$

Next, suppose $w_i \in A_\ell$. Then $y_i/u_i = w_i/v_i < y_{i+1}/u_{i+1}$, so $y_0 = y_{i+1}/u_{i+1}$ by tightness of the third constraint and $y_{i+1} \notin J$. Both w_i and y_i increase in the same direction, so $\bar{w}_i/v_i = \bar{y}_i/u_i$, and there exists small ϵ such that $\bar{w}_i/v_i \leq \bar{y}_{i+1}/u_{i+1}$. For the third constraint, $\bar{y}_i$ and $\bar{w}_i$ negate each other's changes, and $\bar{y}_{i+1}$ is only scaled by $1/(1+\epsilon\beta)$, so equality must still hold.

Finally, suppose $w_i \in A_{\ell+1}$.Then $y_i/u_i > w_i/v_i = y_{i+1}/u_{i+1}$, so $y_0 = y_i/u_i$ by tightness of the third constraint and $y_i \notin J$. Both w_i and y_{i+1} decrease in the same direction, so $\bar{w}_i/v_i = \bar{y}_{i+1}/u_{i+1}$ and are smaller than $\bar{y}_i/u_i$. For the third constraint, $\bar{y}_{i+1}$ and $\bar{w}_i$ negate each other's changes, and $\bar{y}_i$ is only scaled by $1/(1+\epsilon\beta)$, so equality must still hold.

Case 4 - $\bar{y}_i \in A_\ell$ and $\bar{y}_{i+1} \in A_{\ell+1}$ for even $\ell \neq 0$: A similar analysis as case 3 holds.

Hence $(\bar{y}, \bar{w})$ is feasible to Problem (7). We can define a symmetric solution (y', w') as:

$$y'_i = \begin{cases} \frac{1}{1-\epsilon\beta} y_i & \text{if } y_i \notin J \\ \frac{1}{1-\epsilon\beta}(y_i - \epsilon u_i) & \text{if } y_i \in J \cap A_\ell \text{ for odd } \ell \\ \frac{1}{1-\epsilon\beta}(y_i + \epsilon u_i) & \text{if } y_i \in J \cap A_\ell \text{ for even } \ell \neq 0 \end{cases},$$

$$w'_i = \begin{cases} \frac{1}{1-\epsilon\beta} w_i & \text{if } w_i \notin J \text{ or } w_i \in A_0 \\ \frac{1}{1-\epsilon\beta}(w_i - \epsilon v_i) & \text{if } w_i \in J \cap A_\ell \text{ for odd } \ell \\ \frac{1}{1-\epsilon\beta}(w_i + \epsilon v_i) & \text{if } w_i \in J \cap A_\ell \text{ for even } \ell \neq 0 \end{cases},$$

$$\bar{y}_0 = y_0/(1-\epsilon\beta).$$

The same case analysis shows that (y', w') is feasible to Problem (7).

Finally, we have $(y, w) = \lambda(\bar{y}, \bar{w}) + (1-\lambda)(y', w')$ by taking $\lambda = (1+\epsilon\beta)/2$. Hence, (y, w) is not an extreme point, and any extreme point cannot have y_i such that $0 < y_i/u_i < y_0$. □

Using Lemma 15, we can now prove Theorem 7.

Proof of Theorem 7. The third part of condition 1 is satisfied by Lemma 15, so we focus on the first and second parts. If $y_i = 0$ or $y_{i+1} = 0$, then it is clear that $w_i = 0$. Otherwise, $y_i/u_i = y_{i+1}/u_{i+1} = y_0$. Feasibility to the first three constraints of Problem (7) ensures that $w_i/v_i = y_i/u_i = y_{i+1}/v_{i+1}$. □

Appendix C. Other Proofs

Proof of Claim 1. We only prove the first case because the other two cases are similar. Let x^* be an optimal assortment with expected revenue strictly greater than δ:

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

We can multiply both sides by the denominator because it is strictly positive, and rearrange terms to obtain:

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

The left side of the above inequality is upper-bounded by $h(\delta)$. □

Proof of Theorem 2. Assortment optimization under the synergistic MNL model is in NP: given an assortment x and some K, it is easy to check whether $\Pi(x) \geq K$. We prove that the problem is NP-hard via a reduction from the maximum independent set problem.

Given a graph $G = (V, E)$, we need to determine if there exists a subset of vertices $S \subseteq V$ of size K, such that $ij \notin E$ if $i, j \in S$. Index the vertices by $\{1, \ldots, n\}$. We can create a synergistic MNL instance where the expected profit is greater or equal to $K/(1+K)$ if and only if there exists an independent set of size K.

Create a product i for each vertex $i \in V$. We slightly abuse notation and use the same name for a product and its corresponding vertex. Product i has profit $r_i = 0$ and base preference weight $u_i = 0$. For the synergy terms, let $v_j^i = v_i^j = n/2$ if $ij \in E$ and $v_j^i = v_i^j = 0$ otherwise. Then $\bar{r}_{i,j} = 0$, $v_{i,j} = n$ if $ij \in E$, and $\bar{r}_{i,j} = 0$, $v_{i,j} = 0$ if $ij \notin E$.

Introduce an auxiliary product $n+1$, which has profit $r_{n+1} = 1$ and base preference weight $u_{n+1} = 0$. Product $n+1$ creates synergy with all the other products. Specifically, $v_{n+1}^i = 1$ and $v_i^{n+1} = 0$ for all $i = 1, \ldots, n$, so that the preference weight of product $n+1$ increases by 1 for every additional product offered. Then $\bar{r}_{i,n+1} = 1$ and $v_{i,n+1} = 1$. Since this is the only product with non-zero profit, it must be included in an optimal assortment.

We may disregard $S = \emptyset$ as a single vertex is always an independent set. For a set $S \subseteq V$, create an assortment

x where $x_i = \mathbb{1}[i \in S]$ for $i = 1 \le 1 \le n$ and $x_{n+1} = 1$. Then the expected profit is:

$$\Pi(x) = \frac{\sum_{i=1}^n \bar{r}_{i,n+1} x_i}{1 + \sum_{ij\in E} v_{i,j} x_i x_j + \sum_{i=1}^n v_{i,n+1} x_i}$$
$$= \frac{\sum_{i=1}^n x_i}{1 + n\sum_{ij\in E} x_i x_j + \sum_{i=1}^n x_i}.$$

First we show that any optimal assortment must correspond to an independent set in G. If $S \neq \emptyset$ is an independent set, then

$$\Pi(x) = \frac{|S|}{1+|S|} \ge \frac{1}{2}.$$

If S is not an independent set, then there is at least one edge ij where $x_i = x_j = 1$

$$\Pi(x) \le \frac{|S|}{1+n+|S|} < \frac{1}{2}.$$

The optimal assortment must be an independent set S along with product $n+1$, with expected profit $|S|/(|S|+1)$. As the expected profit is increasing in the size of the independent set, there exists an independent set of size K if and only if the optimal expected profit is at least $K/(K+1)$. □

Proof of Lemma 4. Suppose we know the value of δ^* and solved Problem (5) with $\delta = \delta^*$. By Claim 1, Problem (5) has optimal solution V^* with $V_1^* = h(\delta^*) = \delta^*$. Hence V^* is feasible to Problem (6) with objective value δ^*.

Now suppose by contradiction that Problem (6) has optimal solution $\bar{V}$ with optimal value $\bar{V}_1 = \bar{\delta} < \delta^*$. Then $\bar{V}$ is a feasible solution to Problem (5) when $\delta = \bar{\delta}$, with objective value equal to $\bar{\delta}$. But the optimal value of Problem (5) at $\delta = \bar{\delta}$ is $h(\bar{\delta})$, which implies that $h(\bar{\delta}) \le \bar{\delta} < \delta^*$. This contradicts part (iii) of Claim 1. □

Proof of Lemma 5. Construct a solution (y, w) from assortment x as follows:

$$y_0 = \frac{1}{1 + \sum_{j=1}^n u_j x_j + \sum_{j=1}^{n-1} v_j x_j x_{j+1}},$$
$$y_i = \frac{u_i x_i}{1 + \sum_{j=1}^n u_j x_j + \sum_{j=1}^{n-1} v_j x_j x_{j+1}}, \qquad i = 1, \dots, n,$$
$$w_i = \frac{v_i x_i x_{i+1}}{1 + \sum_{j=1}^n u_j x_j + \sum_{j=1}^{n-1} v_j x_j x_{j+1}}, \quad i = 1, \dots, n-1.$$

Then (y, w) is feasible to Problem (7), satisfies condition 1, and has objective value $\Pi(x)$. □

Proof of Lemma 6. Since condition 1 is satisfied, we can rewrite the fourth constraint of Problem (7) as:

$$1 = y_0 + \sum_{i=1}^n y_0 u_i \cdot \mathbb{1}[y_i > 0] + \sum_{i=1}^{n-1} y_0 v_i \cdot \mathbb{1}[w_i > 0].$$

Condition 1 also tells us that

$$\mathbb{1}[w_i > 0] = \mathbb{1}[y_i > 0] \cdot \mathbb{1}[y_{i+1} > 0].$$

Since $x_i = \mathbb{1}[y_i > 0]$, we can solve for y_0 in terms of x:

$$y_0 = \frac{1}{1 + \sum_{i=1}^n u_i x_i + \sum_{i=1}^{n-1} v_i x_i x_{i+1}}.$$

Using the same logic and substituting in the value of y_0, the objective value of (y, w) can be rewritten in terms of x:

$$\sum_{i=1}^n r_i y_i + \sum_{i=1}^{n-1} \bar{r}_i w_i$$
$$= \sum_{i=1}^n r_i y_0 u_i \cdot \mathbb{1}[y_i > 0] + \sum_{i=1}^{n-1} \bar{r}_i y_0 v_i \cdot \mathbb{1}[w_i > 0]$$
$$= \frac{\sum_{i=1}^n r_i u_i x_i + \sum_{i=1}^{n-1} \bar{r}_i v_i x_i x_{i+1}}{1 + \sum_{i=1}^n u_i x_i + \sum_{i=1}^{n-1} v_i x_i x_{i+1}} = \Pi(x).$$

□