Data-driven assortment optimization
Dimitris Bertsimas, Velibor Misic
Data-driven assortment optimization

1 Motivation

A ubiquitous element of business is that of making an assortment decision, which can be described most generally as follows. A firm offers a set of products (an assortment) to a group of individuals. The individuals possess preferences over the different products, which vary from individual to individual, and proceed to select the product that they most prefer; the firm then garners some revenue from their choices. The problem of assortment optimization is to decide what set of products, from a larger set of possible products, should be offered so as to maximize the firm’s expected revenue when customers exercise their preferences. Assortment optimization arises in many contexts, such as retail, online advertising (deciding which ads to display on a webpage) and social security (deciding which funds to allow citizens to invest in for retirement) and is of critical importance in many different domains. In practice, assortment decisions are difficult to make for a number of reasons:

**Choice modeling.** There exists a wide variety of discrete choice models that may be used to model customer choice behavior, leaving the firm with the question of which model to use. If the model is too simple, it may underfit the available data, and if it is too complex, it may overfit the available data, leading to inaccurate predictions.

**Tractability.** Many choice models lead to assortment optimization problems that are theoretically intractable. There is also often an undesirable tradeoff between predictive accuracy and tractability: simple but inaccurate choice models lead to easy optimization problems, while complex choice models lead to intractable problems.

**Constraints.** Firms may have various business rules that limit the possible assortments. For example, a retailer may require that some products be offered together, that only some number of products within a certain subclass of products is offered or that at least some number of products within a given subset of products be offered.

**Auxiliary decisions.** Firms are often not just interested in deciding which products to offer, but may also wish to make auxiliary decisions related to the assortment. For example, a retail firm may be interested in also setting prices for the products that it offers.

2 Our approach

In this ongoing research, we propose a new approach for making assortment decisions that directly addresses the above challenges. The approach is based on modeling the choice behavior of the market through a generic, non-parametric model of choice, and then using this non-parametric choice model to formulate and solve a practically tractable mixed integer optimization (MIO) problem that directly yields the optimal assortment decision.

We assume that there are $n$ products, indicated by $i = 1, 2, \ldots, n$, that the firm can select from, and we use 0 to represent the no-purchase option. We assume that the market is represented by a set of $K$ rankings $\sigma^1, \ldots, \sigma^K$ of the products – for ranking $k$, $\sigma^k(i) < \sigma^k(j)$ if and only if $i$ is preferred to $j$ under ranking $k$ – and that a customer chooses according to ranking $k$ with probability $\lambda^k$. This model of choice is similar to the non-parametric choice
model of Farias et al. [2013], but there is a critical difference in how we proceed from this representation. Farias et al. [2013] use this model to make worst case revenue predictions: for a given assortment, their approach is used to compute the worst case expected revenue, where the worst case is taken over all distributions $\lambda$ supported on the $(n + 1)!$ possible rankings that are consistent with some available transaction data. In contrast, we do not take a worst case approach, but rather we fix a set of rankings and a single probability distribution over this set. It turns out that the corresponding optimization problem can be modeled efficiently as an MIO problem, as we now describe.

In our MIO formulation, the decision variables are $x$ and $y$, where $x_i = 1$ if product $i$ is included in the assortment and 0 otherwise, and $y^k_i = 1$ if under ranking $k$ a customer selects option $i$ (for the current assortment specified by $x$) and 0 otherwise. The formulation we propose has a number of appealing features. First, it turns out that only the $x$ variables need to be binary, because the constraints that connect $x$ and $y$ via the rankings $\sigma_1, \ldots, \sigma^K$ ensure that $y$ is binary whenever $x$ is binary. Thus, although the formulation is an MIO problem and is theoretically intractable, from a practical perspective it is less daunting as there are only $n$ binary variables. Second, the problem lends itself to efficient optimization by branch-and-bound: due to the constraints, branching up on a given $x_i$ variable forces many $y^k_i$ variables to zero. This ensures that branching substantially changes the objective and potentially allows large portions of the branch-and-bound tree to be pruned.

Last but not least, the formulation readily accommodates many kinds of constraints and auxiliary decisions via standard MIO modeling techniques. For example:

**Precedence constraints:** “If $i$ is offered, we must offer $j$” can be written as $x_i \leq x_j$.

**Subset constraints:** “We must include at least $L_S$ and at most $U_S$ products from the subset $S \subseteq \{1, \ldots, n\}$” can be represented as $L_S \leq \sum_{i \in S} x_i \leq U_S$.

**Pricing:** Suppose that we can introduce product $i$ at $P_i$ different prices, indexed from 1 to $P_i$. To model this, we model each product-price combination $(i, p)$ for $p \in \{1, \ldots, P_i\}$ as a new product and instead of $x_i$, we use $x_{i,p}$ to model whether product $i$ at price $p$ is in the assortment or not. We then add the constraint $\sum_{p=1}^{P_i} x_{i,p} \leq 1$ that ensures that product $i$ is offered at a single price, if it is offered at all.

## 3 Preliminary results

We present here some preliminary results that demonstrate the value of our approach.

**Tractability.** To test the tractability of the formulation, we consider the following experiment. We fix the number of products $n$ to 30 and the number of rankings $K$ to 1000. We then randomly generate 100 instances, where we uniformly at random generate the set of rankings $\sigma_1, \ldots, \sigma^K$ from the set of all possible permutations, the revenue $r_i$ of each product $i$ from the set $\{1, \ldots, 100\}$, and the probability distribution $\lambda$ from the $(K-1)$-dimensional unit simplex. We solve both the linear optimization (LO) relaxation and the actual MIO problem itself using Gurobi 5.60. We do not impose any constraints or incorporate any auxiliary decisions. In 33 of the 100 cases, the LO solution was integral. Figure 1a plots a histogram of the optimality gap of the LO relaxation and the integer optimal solution; we can see that even when the solution is not integral, the optimality gap is small. Figure 1b
plots a histogram of the time required to solve the MIO problem. Here, we can verify that although the problem is an MIO, it is amenable to modern MIO solution technology and can be solved to provable optimality within an operationally feasible timeframe (in the worst case, no more than 13 seconds).

**Constraints.** We again fix $n$ to 30 and $K$ to 1000, and randomly generate 100 instances in the same way as in the tractability experiment. For each instance, we generate a random collection of precedence constraints. We compare our MIO formulation to an adapted version of the ADXOpt local search algorithm of Jagabathula [2014], where the set of candidate moves is restricted to those that do not violate feasibility. We find that in 38 of the 100 instances, the ADXOpt solution is suboptimal, and among this subset of instances, the MIO expected revenues were on average 2% higher. Although the solution times are higher (6.4 seconds on average for MIO compared to 0.4 seconds for ADXOpt), the MIO problem can still be solved quickly to provable optimality in the presence of constraints.

**References**
