Product line optimization with multiples sites

Sebastián Dávila, Martine Labbé, Vladimir Marianov, Fernando Ordóñez, Frédéric Semet

Product line optimization with multiples sites

Sebastián Dávila[a], Martine Labbé[b], Vladimir Marianov[c], Fernando Ordónẽz[a], Frédéric Semet[d]

[a] *Department of Industrial Enginnering, Universidad de Chile, Chile*

[b] *Université Libre de Bruxelles, Belgium*

[c] *Department of Electrical Engineering, Pontificia Universidad Católica de Chile and Instituto Sistemas Complejos de Ingeniería (ISCI), Chile*

[d] *Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRIStAL Lille, France*

Abstract

We consider the problem faced by a retailer that selects the set of products to allocate in finite capacity stores to maximize patronage. The purchase decision is made by customers that purchase exactly one product that maximizes her utility that depends on the product price, distance traveled to the store and reservation price, known to the retailer. The retailer's bilevel optimization problem is transformed into an integer optimization formulation. Small size instances are solved optimally, while for large instances, we explore Benders Decomposition, Branch and Cut and Cut and Branch to solve the problem. Our computational results show that the proposed Cut and Branch method obtains the best results, and improves on the current state of the art.

Keywords: Product allocation to multiple stores; Bilevel programming; Location; Cut and Branch; Branch and Cut

1. Introduction

Inventory and product obsolescence costs are very significant in the retail industry. Reducing these costs requires a fast turnaround of the inventory, which depends, in turn, on a good knowledge of customers' preferences. When the products being offered to customers do not match these preferences, there is a slow inventory turnaround and high end-of-season stock levels. To get rid of the stock, and free shelf and storage space in their stores, retailers must significantly mark the prices down.

Furthermore, consumers' preferences and buying power depend on their locations. A retailer with several outlets in different cities, or different neighborhoods of a city must have information on what are the products and brands that fit each store customers' preferences. Matching these preferences to the right products, sizes, colors, and brands is essential to avoid maintaining long-term stocks of products that do not sell. This issue is coupled with the fact that display space in stores and supermarkets is limited, which precludes displaying all possible products, and requires displaying and having stock of only those products that are right for local customers. This strategy can also be accompanied by a controlled markdown strategy that, although reduces unit profit, maximizes sales as it attracts more customers.

We address these issues in the case of a retailer with several stores that desires to select, for each location, a subset of mutual imperfect substitute products, i.e., products that serve the same customers' needs, but differ in color, appearance, flavor, and similar attributes, such that a customer purchases only one of them at a time. The choice of such subset must be optimal to fit the limited space, as well as the preferences of the local customers, and it can include a smart marking down strategy. In this, we follow a JCPenney executive who states that assortments, allocations, markdown pricing are all linked and optimized together at his company (Ghoniem and Maddah, 2015).

The assortment selection problem has attracted much attention. In (Green and Krieger, 1985), the authors present two formulations to help a retailer make decisions on the composition of a set of similar products, introducing heuristic algorithms to solve the problems. A review of the literature on product selection is presented in (Green and Krieger, 1989), discussing issues

such as cannibalization between products, different objectives, and differentiation of buyers by buying power. They focus on the line of products of a manufacturer, although the work is applicable to a retailer. The models require knowledge of the positioning of the products and consumers in an attribute space. Display and storage space is not an issue. In (Belloni et al., 2008), a ranking-based formulation is presented for the Product Line Design (PLD) problem where the utility of each client is introduced in the formulation as a constraint. In (McBride and Zufryden, 1988), a comparison is performed between different heuristics for the PLD problem. The authors present a new ranking-based formulation, and solve it with Lagrangian Relaxation. These previous approaches ((Belloni et al., 2008) and (McBride and Zufryden, 1988)) are compared in (Bertsimas and Mišić, 2019), which also introduces a strong formulation for (Belloni et al., 2008). A Benders Decomposition (BD), together with an efficient algorithm, is used to solve the problem, where the master problem defines the available products and a separable slave problem solves each client's purchase decision. The efficiency of the BD is evaluated on synthetic data, providing good results.

The problem of selecting a subset of products and allocating them to a limited, integer number of shelves in a supermarket, so that a function of costs and revenues is maximized is addressed in (Zufryden, 1986). In that work, the demand for the product depends on shelf space, price, advertising, promotions, and store attraction, among other factors and a dynamic programming solution method is used. In (Dobson and Kalish, 1993), the authors use conjoint analysis to decide the product assortment, based on each products fixed and variable costs, as well as its cannibalization effects on other products, rather than storage and shelf space. Each customer segment purchases the product that provides her with the best utility. The problem is solved using a heuristic. Demand substitution, which means that a consumer prefers to buy a product that is an imperfect substitute of her best choice, rather than not making the purchase at all is addressed in (Yücel et al., 2009). Their problem considered also exogenous demand, supplier selection, shelf space limitations and inventory management considerations. In (Ghoniem and Maddah, 2015), the product assortment is optimized, together with the pricing and inventory decisions, not considering either space limitations or preferences variation across the region of interest. They consider multiple periods and customer segments, and determine optimal prices and inventory levels for the subset of products in each time period. A deterministic utility

function is used depending on reservation and selling prices, and make customers purchase their best choice. They also do the exercise of calibrating consumers reservation price. In (Ghoniem et al., 2016), the optimal product assortments and pricing for multiple product categories are found when these are complementary (nachos, cheese spread and guacamole). Consumers can choose to purchase the primary product (nachos) and make a mix with complementary categories (a brand of cheese spread or guacamole). Customers choose according to a deterministic utility function. The problem of assortment planning and pricing in a competitive setting is addressed in (Besbes and Sauré, 2016) using a multinomial logit model for consumers demand. The paper (Moon et al., 2017) solves the problem of selecting a mutual substitute product line by a retailer who also selects the price among a set of discrete prices. They include limited shelf space and dynamic substitution, the behavior that makes consumers purchase products that are not necessarily their best choice when it is unavailable. They consider one line of products without differentiating by store and solve it using a genetic algorithm. In (Hübner and Schaal, 2017) the authors optimize shelf-space planning in a store, taking into account products that are substitutes to each other (e.g., different brands) and the effects of not listing products or replacing them by other products, and the effect of these actions on the demand for other products. Their main problem is allocation of products to shelves and deciding how much space to allocate to each product.

A frequent practice is optimizing shelf space together with the determination of a set of products to be displayed, that are not substitutes of each other, i.e., items that consumers could purchase together (Chen and Lin, 2007). Other authors that address the same problem with different variants are (Flamand et al., 2017, Hübner and Schaal, 2017, Kök et al., 2008, Hübner and Kuhn, 2012), who offer a review on the subject. A later review is presented by (Kök et al., 2015), which states that no dominant solution has yet emerged for assortment planning, so assortment planning represents a wonderful opportunity for academia to contribute to enhancing retail practice

Reviews of planning of mutual substitute products assortment can be found in (Shin et al., 2015), on planning of product lines and (Mou et al., 2017) on retail store operations, including a good section on product assortment. There exists also commercial software, which solves the problem of product allocation to stores using simple rules of thumb (Hübner and Kuhn, 2012).

In (Chen et al., 2015), markdowns are dealt with. They assume multiple stores owned by different chains, served by a single warehouse over a time horizon during which, products can take different discrete prices from a set. Store owners must follow a set of rules to maintain a fair competition and inventory turnover. There is just one product and demand is stochastic. von Stackelberg pricing was the subject of (Briest and Krysta, 2011), who assume customers that purchase sets of products and are not capable of computing their exact utilities, as they are computationally bounded. Somewhat related is the competitive facility location problem (see Eiselt et al., 2015). However, the players of the von Stackelberg game are the firms locating their stores, as opposed to a firm and customers. For a review of bi-level models for competitive location, (see Aras and Küçükaydın, 2017). Finally, (Bhatnagar and Syam, 2014), address the problem of product allocation among brick-and-mortar stores and online stores belonging to the same chain. None of the reviewed papers takes into account the geographical distribution of customers, the possibility of customers purchasing at different stores, and the fact that different stores belonging to the same retailer can offer different product lines with possibly different mark-downs. This is our contribution.

Note that the structure of the MPLP is similar to that of the Facility Location Problem with Clients' Preferences (FLPCP). Valid inequalities for the FLPCP are developed in (Cánovas et al., 2007) and (Vasilyev and Klimentova, 2010), for their use within a Branch & Cut solution procedure.

We present a bilevel formulation to this problem, which is collapsed to a single level integer optimization formulation. We adapt valid inequalities that have been used for the Facility Location problem with Preferences to solve our problem using Branch and Cut ($B\&C$) and Cut and Branch ($C\&B$) methods. We compare the $B\&C$ and $C\&B$ with existing Benders' decomposition methods for the single-store case.

The paper is organized as follows: Section 2 presents the problem, the bilevel formulation, three different single-level formulations, and some valid inequalities. In section 3, the proposed solution methods are described. Computational testing comparing the different formulations and methods is presented in Section 4. Finally we present our conclusions and lines for future work in section 5.

2. Problem definition and formulations

This Section presents a description of the Multi-Product Location Problem (MPLP) considered in this work. We provide three different formulations for the MPLP and introduce valid inequalities that are used later.

2.1. Problem description

A firm owns a set $\mathcal{J}$ of stores, geographically distributed over a region. A set $\mathcal{K}$ denotes all products in a category, e.g., TV sets of a certain screen size. All products in this set are imperfect substitutes of each other that differ in secondary characteristics and price. Let $\pi_{j,k}$ be the unit price of product $k \in \mathcal{K}$ at store $j \in \mathcal{J}$. There are capacity constraints, indicating that each store $j \in \mathcal{J}$ can display (and offer) up to p_j products of the set $\mathcal{K}$.

There is also a set $\mathcal{I}$ of consumers or potential clients. The distance between consumer i and store j is d_{ij}. We assume that distances are symmetric, i.e. the distance of the round trip between client i and store j is $2 \cdot d_{ij}$. Each consumer $i \in \mathcal{I}$ is interested only in those products in a set $\mathcal{K}_i \subset \mathcal{K}$ and has a reservation price r_{ik} for product k in $\mathcal{K}_i$. Customers will buy at most one unit of product at one store, provided that the full cost (price plus travel cost) of the purchase does not exceed the reservation price for that product. Each consumer chooses the product and store that maximizes her utility, i.e., the surplus obtained subtracting the full cost of the product from the reservation price. The firm must decide what products in $\mathcal{K}$ to display at each store, to maximize its revenue.

Tables 1 and 2 present the set and parameter notation used in this work.

Table 1: Set notation

$\mathcal{I}$	set of clients
$\mathcal{J}$	set of stores
$\mathcal{K}$	set of products
$\mathcal{K}_i$	set of products client $i \in \mathcal{I}$ is interested in

Table 2: Model parameters

d_{ij}	distance between client $i \in \mathcal{I}$ and store $j \in \mathcal{J}$
p_j	maximal number of products assigned to store $j \in \mathcal{J}$
$\pi_{j,k}$	unit price for product $k \in \mathcal{K}$ in store $j \in \mathcal{J}$
r_{ik}	reservation price of client $i \in \mathcal{I}$ for product $k \in \mathcal{K}$

2.2. Bilevel model formulation

The multi-product location problem can be formulated as a linear bilevel optimization problem. We begin by introducing two sets of binary decision variables: a variable y_{jk} for $j \in \mathcal{J}$, $k \in \mathcal{K}$ that represents the retailer's decision to place product k in store j ($y_{jk} = 1$) or not ($y_{jk} = 0$); and a variable $\tilde{x}_{ijk}$, for $i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}_i$, that encodes the decision of whether client i purchases product k at store j ($\tilde{x}_{ijk} = 1$) or not ($\tilde{x}_{ijk} = 0$).

With these variables we can express the retailer's MPLP of placing product in capacity limited stores in order to maximize revenue as the following bilevel optimization problem:

$$\max_{x,y} \quad \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}_i} \pi_{jk}\tilde{x}_{ijk} \tag{1}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} y_{jk} \leq p_j \qquad j \in \mathcal{J} \tag{2}$$

$$y_{jk} \in \{0,1\} \qquad j \in \mathcal{J}, k \in \mathcal{K} \tag{3}$$

for each $i \in \mathcal{I}$ we have

$$x = \arg\max_{\tilde{x}} \quad \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}_i} (r_{ik} - \pi_{jk} - 2d_{ij})\tilde{x}_{ijk} \tag{4}$$

$$\text{s.t.} \quad \tilde{x}_{ijk} \leq y_{jk} \qquad j \in \mathcal{J}, k \in \mathcal{K}_i \tag{5}$$

$$\sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}_i} \tilde{x}_{ijk} \leq 1 \tag{6}$$

$$\tilde{x}_{ijk} \in \{0,1\} \qquad j \in \mathcal{J}, k \in \mathcal{K}_i \tag{7}$$

Note that customer i would not purchase a product for which the reservation price r_{ik} is smaller than the price π_{jk} plus the distance traveled $2d_{ij}$. Therefore we can define the set of products/locations that are attractive to customer i as: $\mathcal{T}_i = \{(j,k)|r_{ik} - \pi_{jk} - 2d_{ij} \geq 0\}$. With this notation we can set $\tilde{x}_{ijk} = 0$ if $(j,k) \notin \mathcal{T}_i$. We replace variable $\tilde{x}_{ijk}$, with variable x_{ijk} with $(j,k) \in T_i$, $i \in \mathcal{I}$ and we can express (1), (4) and (6) as:

$$\sum_{i \in \mathcal{I}} \sum_{(j,k) \in \mathcal{T}_i} \pi_{jk} x_{ijk}$$

$$\sum_{(j,k) \in \mathcal{T}_i} (r_{ik} - \pi_{jk} - 2d_{ij}) x_{ijk}$$

$$\sum_{(j,k) \in \mathcal{T}_i} x_{ijk} \leq 1 \ .$$

2.3. Single level model formulation

We now present a single level formulation for the MPLP, following the approach in (Hansen et al., 2004), which enforces the second level optimality conditions through constraints. This approach considers the following order relation for all $i \in \mathcal{I}$:

$$(j,k) \preceq (j',k') \text{ if and only if } r_{ik} - \alpha_{jk}\pi_{j,k} - 2d_{ij} \leq r_{ik'} - \alpha_{j'k'}\pi_{j',k'} - 2d_{ij'}$$

that sorts product/location pairs. This order means that (j',k') is greater than (j,k) if it provides a larger reward for customer $i \in \mathcal{I}$. Then we define the set of products/locations that are preferred to $(j,k) \in \mathcal{T}_i$ for client $i \in \mathcal{I}$ by $\mathcal{B}_{ijk} = \{(j',k') \in \mathcal{T}_i | (j,k) \preceq (j',k')\}$. Note that $(j,k) \in \mathcal{B}_{ijk}$. A vector x, feasible for the second level problem (4)-(7), is optimal for this problem if and only if it satisfies the following set of constraints (see Hansen et al., 2004)

$$\sum_{(j',k') \in \mathcal{B}_{ijk}} x_{ij'k'} \geq y_{jk} \qquad i \in \mathcal{I}, (j,k) \in \mathcal{T}_i \ . \tag{8}$$

This gives the following equivalent single level integer programming formu-

lation for the MPLP:

$$(M1) \quad \max \quad \sum_{i\in\mathcal{I}} \sum_{(j,k)\in\mathcal{T}_i} \pi_{j,k} x_{ijk} \tag{9}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} y_{jk} \leq p_j \qquad j \in \mathcal{J} \tag{10}$$

$$\sum_{(j',k')\in\mathcal{B}_{ijk}} x_{ij'k'} \geq y_{jk} \qquad i \in \mathcal{I}, (j,k) \in \mathcal{T}_i \tag{11}$$

$$x_{ijk} \leq y_{jk} \qquad i \in \mathcal{I}, (j,k) \in \mathcal{T}_i \tag{12}$$

$$\sum_{(j,k)\in\mathcal{T}_i} x_{ijk} \leq 1 \qquad i \in \mathcal{I} \tag{13}$$

$$x_{ijk} \in \{0,1\} \qquad i \in \mathcal{I}, (j,k) \in \mathcal{T}_i \tag{14}$$

$$y_{jk} \in \{0,1\} \qquad j \in \mathcal{J},\ k \in \mathcal{K}\ . \tag{15}$$

We now consider alternative constraints to (11). For this define the set of products that are not preferable to $(j,k) \in \mathcal{T}_i$ for client $i \in \mathcal{I}$, that is $\mathcal{W}_{ijk} = \{(j',k') \in \mathcal{T}_i | (j,k) \succ (j',k')\}$. Which satisfies $\mathcal{W}_{ijk} \cap \mathcal{B}_{ijk} = \emptyset$ and $\mathcal{T}_i = \mathcal{W}_{ijk} \cup \mathcal{B}_{ijk}$. Then, for $i \in \mathcal{I}, (j,k) \in \mathcal{T}_i$, constraint (11) is equivalent to

$$\sum_{(j',k')\in\mathcal{B}_{ijk}} x_{ij'k'} + \sum_{(j',k')\in\mathcal{W}_{ijk}} x_{ij'k'} \geq y_{jk} + \sum_{(j',k')\in\mathcal{W}_{ijk}} x_{ij'k'}$$

$$\sum_{(j',k')\in\mathcal{T}_i} x_{ij'k'} \geq y_{jk} + \sum_{(j',k')\in\mathcal{W}_{ijk}} x_{ij'k'} \tag{16}$$

By (13), the left hand side of (16) is less than or equal to 1. With this derivation it is possible to show M1 is equivalent to a problem that substitutes (11) with (17), below:

$$y_{jk} + \sum_{(j',k')\in\mathcal{W}_{ijk}} x_{ij'k'} \leq 1 \qquad i \in \mathcal{I}, (j,k) \in \mathcal{T}_i\ . \tag{17}$$

We define by M2 the optimization problem obtained by replacing (11) in M1 by (17). Note that this change does not increase the size of the formulation as both sets of constraints have $\sum_{i\in\mathcal{I}} |\mathcal{T}_i|$ constraints.

2.4. Valid inequalities

Here we introduce additional valid inequalities for the MPLP that are constructed considering the interaction between more than one client $i \in \mathcal{I}$, to help define tighter equivalent formulations for the problem.

We strengthen constraint (17) by considering the set of product/locations that are worse than (j,k) for a second client i' but not for i. That is the set $\mathcal{W}_{ii'jk} = W_{i'jk} \cap \mathcal{B}_{ijk}$. Using this set, similar to (Cánovas et al., 2007), we obtain the following set of stronger inequalities:

$$\sum_{(j',k') \in \mathcal{W}_{ijk}} x_{ij'k'} + \sum_{(j',k') \in \mathcal{W}_{ii'jk}} x_{i'j'k'} + y_{jk} \leq 1 \qquad i,i' \in \mathcal{I}, (j,k) \in \mathcal{T}_i \ . \quad (18)$$

Note that this inequality is satisfied when $\sum_{(j',k') \in \mathcal{W}_{ijk}} x_{ij'k'} = 1$, because in that case, $y_{\tilde{j},\tilde{k}} = 0$ for any $(\tilde{j},\tilde{k}) \succeq (j,k)$. This means that $x_{i\tilde{j}\tilde{k}} = 0$ for any $i \in \mathcal{I}$, giving $\sum_{(j',k') \in \mathcal{W}_{i'jk} \cap \mathcal{B}_{ijk}} x_{i'j'k'} = 0$.

Constraint (18) can be generalized to multiple clients, as shown in (Cánovas et al., 2007). Given $i_1, \ldots, i_s \in \mathcal{I}$, and $(j,k) \in \mathcal{T}_{i_1}$, then the following inequalities are valid for MPLP:

$$\sum_{(j',k') \in \mathcal{W}_{i_1jk}} x_{i_1j'k'} + \sum_{t=2}^{s} \sum_{(j',k') \in B_{i_1jk} \cap \left(\bigcap_{q=2}^{t} \mathcal{W}_{i_qjk}\right)} x_{i_tj'k'} + y_{jk} \leq 1 \ . \quad (19)$$

We now present valid inequalities that do not arise from strengthening of constraint (11). The next two sets of valid inequalities are stated in propositions that establish relationships between variables involving two customers.

Proposition 1. *Let x, y be a feasible solution for M1. Then for $i, i' \in \mathcal{I}$, $(j,k) \in \mathcal{T}_i$, we have*

$$x_{ijk} \leq x_{i'jk} \quad \text{if} \quad \mathcal{B}_{i'jk} \subseteq \mathcal{B}_{ijk} \quad (20)$$

Proof: Assume that $\mathcal{B}_{i'jk} \subseteq \mathcal{B}_{ijk}$. If $x_{ijk} = 1$ then, with (13), we have that $x_{ij'k'} = 0$ for all $(j',k') \in \mathcal{B}_{ijk} \setminus \{(j,k)\}$. This implies that $y_{j',k'} = 0$ for all

$(j',k') \in \mathcal{B}_{ijk} \setminus \{(j,k)\}$ by using (11) and that $\mathcal{B}_{ij'k'} \subseteq \mathcal{B}_{ijk} \setminus \{(j,k)\}$ for any such (j',k'). From $x_{ijk} = 1$, (12) and (11) we get $1 = y_{jk} \leq \sum_{(j',k') \in \mathcal{B}_{i'jk}} x_{i'j'k'}$. If $x_{i'jk} = 0$ then there exists some $(j',k') \in \mathcal{B}_{i'jk} \setminus \{(j,k)\}$ such that $x_{i'j'k'} = 1$, which in turn, by (12) implies $y_{j'k'} = 1$. This is a contradiction since $(j',k') \in \mathcal{B}_{ijk} \setminus \{(j,k)\}$ by the hypothesis and we showed above that $y_{j'k'} = 0$. Therefore $x_{i'jk} = 1$ completing the proof. □

Proposition 1 generalizes a result in (Cánovas et al., 2007) to the case when the sets of preferred products are different for each client. That result, when $\mathcal{B}_{ijk} = \mathcal{B}_{i'jk}$, can be obtained as a corollary by repeating Proposition 1.

Corollary 1. *Let x, y be a feasible solution for M1. Then for $i, i' \in \mathcal{I}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$,*

$$x_{ijk} = x_{i'jk} \quad \text{if} \quad \mathcal{B}_{ijk} = \mathcal{B}_{i'jk} \tag{21}$$

The following result, requires the definition of the set $\mathcal{B}_{ii'jk} = \mathcal{B}_{ijk} \cap \mathcal{T}_{i'}$ of product/location pairs that are preferred to (j,k) for i that are profitable for i'. Note that this set $\mathcal{B}_{ii'jk}$ is empty if $\mathcal{T}_i \cap \mathcal{T}_{i'} = \emptyset$.

Proposition 2. *Let x, y be a feasible solution for M2. Then for $i, i' \in \mathcal{I}$, $(j,k) \in \mathcal{T}_i$, we have*

$$\sum_{(j',k') \in \mathcal{W}_{ijk}} x_{ij'k'} + \sum_{(j',k') \in \mathcal{B}_{ii'jk}} x_{i'j'k'} \leq 1 \ . \tag{22}$$

Proof: Note that if $\mathcal{B}_{ii'jk} = \emptyset$, (22) is true as it is implied by (13). Assume therefore that $\mathcal{B}_{ii'jk} \neq \emptyset$. If the first sum is equal to one, then there exists $(j',k') \in \mathcal{W}_{ijk}$ such that $x_{ij'k'} = 1$. This means, considering (17) for $(\hat{j},\hat{k}) \succeq (j,k)$, that $y_{\hat{j}\hat{k}} = 0$ for all $(\hat{j},\hat{k}) \in \mathcal{B}_{ijk}$. In particular, $y_{\hat{j}\hat{k}} = 0$ for all $(\hat{j},\hat{k}) \in \mathcal{B}_{ii'jk}$. This and (12) imply the second sum is zero. Consider now that the second sum is equal to 1. Then there exists $(j',k') \in \mathcal{B}_{ii'jk} \subseteq \mathcal{B}_{ijk}$ such that $x_{i'j'k'} = 1$. Because of (12), $y_{j'k'} = 1$, and therefore $x_{i\hat{j}\hat{k}} = 0$ for all $(\hat{j},\hat{k}) \in \mathcal{W}_{ij'k'}$ due to equation (17). This implies the first sum is zero since $(j',k') \in \mathcal{B}_{ijk}$ means $\mathcal{W}_{ijk} \subseteq \mathcal{W}_{ij'k'}$. □

Using the above valid inequalities we construct two additional equivalent formulations for the MPLP. We note that (20) has at most $\sum_{i\in\mathcal{I}} |\mathcal{T}_i|$ total constraints, while (22) could have up to $|\mathcal{I}| \sum_{i\in\mathcal{I}} |\mathcal{T}_i|$ total constraints.

We denote by M3 the problem that considers valid inequalities (20) and replaces (11) in M1 with (18). This formulation considers constraints that model interactions between pairs of consumers. We also define M4 as the problem that incorporates valid inequalities (20) and replaces (11) in M1 with (19), modeling interactions between sets of customers. These formulations are summarized in Table 3. Since constraints (17), (18), and (19) are

Table 3: Summary of the different formulations

	(9)	(10)	(11)	(12)	(13)	(17)	(18)	(19)	(20)
M1	X	X	X	X	X				
M2	X	X		X	X	X			
M3	X	X		X	X		X		X
M4	X	X		X	X			X	X

increasingly stronger constraints, the corresponding formulations are tighter formulations of the MPLP. Our approach to solve large instances of M3 and M4 will consider subsets of constraints (18) and (19), respectively. In addition, we consider the effect of including inequalities (22) in these formulations. Since there is a large number of these constraints we add them using cutting plane approaches, similar to (Vasilyev and Klimentova, 2010), as we see below.

3. Solution methods

The formulations M2, M3 and M4 of the MPLP problem presented above are binary optimization problems that can be directly handled by a commercial solver. To efficiently solve large instances we investigate different decomposition strategies for the MPLP. In this section we present a Benders decomposition strategy that is applicable to M2, problem reductions for M3 and M4, and a cut generation method for inequalities (22). We finish with some variable and constraint simplifications for MPLP.

3.1. Existing Benders decomposition method

In (Bertsimas and Mišić, 2019), a Benders decomposition approach is presented, using a formulation similar to M2, when there is only one store. The method considers the decision of which products to place in the store as the master problem, letting the customer's purchase decision be the second stage problem, which is separable in $|\mathcal{I}|$ independent sub-problems.

An efficient algorithm to solve these sub-problems is introduced. This solution method first assigns to every client the highest value product that is available to her, then for each client uses a greedy algorithm to solve the dual problem, which generates the Bender's Cuts that are added as lazy constraints. To the best of our knowledge, this is the most efficient solution method available for the one store problem, and we use this solution method as a benchmark for the solution methods proposed here.

To adapt this Benders decomposition method to our multiple store case, we let the master problem solve the product availability problem at all stores, and given the solution of the master problem, the sub-problems remain each customer's purchase decision. In the case of the M2 formulation, these sub-problems remain separable in $|\mathcal{I}|$ sub problems and can still be solved efficiently using the algorithm suggested in (Bertsimas and Mišić, 2019). In this adaptation of the Benders decomposition method, each sub-problem can generate a Benders' cut in every iteration.

The fact that formulations M3 and M4 consider constraints that involve multiple clients breaks the separability of the sub-problems and generates sub-problems that cannot be solved with the method proposed in (Bertsimas and Mišić, 2019). Adapting this solution method for these formulations in the multiple store problem is not straightforward and the topic of future research. We therefore only consider this Benders decomposition method on formulation M2, and refer to it as M2-BD.

3.2. Cut generation methods

To reduce problem size, here we introduce relaxations to problems M3 and M4 by considering only a subset of constraints (18) and (19). We also present

a cut generation approach that gradually incorporates constraints (22). Cut generation approaches can either be used to add cuts only at the root node of the branch and bound tree, known as a Cut and Branch ($C\&B$) approach or used to add cuts throughout the branch and bound tree as needed, a Branch and Cut ($B\&C$) approach.

The other potentially large set of constraints, (20), does not generate a computational difficulty in our computational experiments either due to the fact that they are precedence constraints or because the condition $\mathcal{B}_{i'jk} \subseteq \mathcal{B}_{ijk}$ is difficult to satisfy.

3.2.1. A subset of constraints (18)

Note that for a particular constraint (18) to be different from the constraints in (17) it is necessary that $\mathcal{W}_{ii'jk} \neq \emptyset$. Given the definition of the set $\mathcal{W}_{ii'jk}$ this set can be large when the attractive products for two clients overlap significantly. Therefore, the subsets of constraints (18) that are selected, correspond to pairs of clients $i, i' \in \mathcal{I}$ with $i \neq i'$ that have a large set of common attractive product/locations, i.e. large $c_{ii'} = |\mathcal{T}_i \cap \mathcal{T}_{i'}|$.

To find a set of pairs of clients that have a large number of common attractive product/locations we consider the following optimization problem.

$$\begin{aligned} \max \quad & \sum_{i \in \mathcal{I}} \sum_{\substack{i' \in \mathcal{I} \\ i \neq i'}} c_{i,i'} z_{i,i'} & \\ \text{s.t.} \quad & \sum_{\substack{i' \in \mathcal{I} \\ i \neq i'}} (z_{i,i'} + z_{i',i}) = 1 & i \in \mathcal{I} \\ & z_{i,i'} \in \{0,1\} & (i,i') \in \mathcal{I}, i \neq i' \end{aligned}$$

The optimal solution to this optimization problem indicates which pair of clients to use to build the subset of constraints (18). We then include one such constraint for each pair i, i' such that $z_{i,i'} = 1$. Note that the number of constraints generated are equal to the number of constraints in (17), since we generated the constrains for $(i,i'), (j,k) \in \mathcal{T}_i$ and $(i,i'), (j,k) \in \mathcal{T}_{i'}$

3.2.2. A subset of constraints (19)

There is a set of constraints (19) for every possible group of clients $\{i_1, \ldots, i_r\}$. To identify which groups of clients we use to generate the subset of (19) we use the following procedure, introduced in (Cánovas et al., 2007).

1. Let $(j,k) \in \mathcal{J} \times \mathcal{K}$ and let $C = \{i_1 \in \mathcal{I} : \exists i_2 \in \mathcal{I} \mid \mathcal{W}_{i_1jk} \cap \mathcal{W}_{i_2jk} = \emptyset,\ \mathcal{W}_{i_1jk} \cap \mathcal{B}_{i_2jk} \neq \emptyset,\ \mathcal{W}_{i_2jk} \cap \mathcal{B}_{i_1jk} \neq \emptyset\}$.
2. We consider the graph obtained from associating a node to each element of C and an edge to each pair $(i_1, i_2) \in C \times C$ such that $\mathcal{W}_{i_1jk} \cap \mathcal{W}_{i_2jk} = \emptyset,\ \mathcal{W}_{i_1jk} \cap \mathcal{B}_{i_2jk} \neq \emptyset,\ \mathcal{W}_{i_2jk} \cap \mathcal{B}_{i_1jk} \neq \emptyset$.
3. We search for a clique $\{i_1, \ldots, i_r\}$ in this graph and replace the inequalities $\sum_{(j',k') \in \mathcal{W}_{i_tjk}} x_{i_tj'k'} + y_{jk} \leq 1$ for $t = 1, \ldots, r$, with the tighter inequality $\sum_{t=1}^{r} \sum_{(j'k') \in \mathcal{W}_{i_tjk}} x_{i_tj'k'} + y_{jk} \leq 1$.
4. Nodes $i_1, \ldots, i_r$ are removed form the graph and the process is repeated with the remaining nodes until a graph with no edges is obtained.

This procedure, modifies constraints (17) in step 3 by replacing them with constraints of the form (19). Since at every iteration we are introducing tighter constraints, the resulting subset of (19) implies constraints (17).

3.2.3. Cut generation

For the cut generation strategy we solve problem M4 adding a subset of constraints (22). Given a solution to this problem, we check whether any of the remaining (22) constraints is violated. For each client, We generate at most one of the violated constraints, including it in the formulation. Then, we re-optimize and repeat this procedure until the optimal solution satisfies all constraints (22). A similar cut generation strategy for constraints (19) was not competitive.

A critical part in constructing an effective cut generation strategy is to be able to quickly check if there are violated constraints. For the case of constraint

(22) we begin by noting that it is not necessary to check all inequalities indexed in $(j,k) \in \mathcal{T}_i$ for a given pair $i, i' \in \mathcal{I}$. For this define $\sigma_i(j,k)$ as the position of pair (j,k) in the set of preferences $\mathcal{T}_i$ (in increasing order with respect to the utility of client i). Define also $(j,k)^i_{\min} = \arg\min_{(j,k)\in S}\{\sigma_i(j,k)\}$ and $(j,k)^{i'}_{\max} = \arg\max_{(j,k)\in S\cap\mathcal{T}_{i'}}\{\sigma_i(j,k)\}$, where $S = \{(j,k)|x_{ijk} > 0\ (j,k) \in \mathcal{T}_i\}$. We now show that for any $(j,k) \notin [(j,k)^i_{\min}, (j,k)^i_{\max}]$ variable x satisfies constraint (22).

By definition we have that $x_{ij'k'} = 0$ for all $(j',k') \in \mathcal{W}_{i(j,k)^i_{\min}}$, which implies that $\sum_{(j'k')\in\mathcal{W}_{i(j,k)^i_{\min}}} x_{ij'k'} = 0$ hence inequality (22) is satisfied. Likewise, $x_{i'jk} = 0$ for all $(j,k) \in \mathcal{B}_{ii'(j,k)^{i'}_{\max}}$ which implies that $\sum_{(j'k')\in\mathcal{B}_{ii'(j,k)^{i'}_{max}}} x_{i'j'k'} = 0$ hence inequality (22) is satisfied.

The process to generate cuts by only verifying the range $[(j,k)^i_{\min}, (j,k)^i_{\max}]$ for each $i, i' \in \mathcal{I}$ is described in Algorithm 1 below. Note that we add at most one valid inequality for each client in each iteration. These inequalities are added to the problem as lazy constraints.

3.3. Problem preprocessing

To speed up the solution for these models, we remove or simplify constraints that are easy to check, reducing the problem size. In particular we conduct the following simplifications for model M2:

- If $|\mathcal{W}_{ijk}| = 0$ ($|\mathcal{B}_{ijk}| = \mathcal{T}_i$) then the pair (j,k) is the worst for client i. Constraint (17) becomes $y_{jk} \leq 1$ and can be removed.
- If $|\mathcal{W}_{ijk}| = |\mathcal{T}_i| - 1$ ($|\mathcal{B}_{ijk}| = 1$) then the pair (j,k) is the best option for client i. Constraint (17) becomes $y_{jk} \leq x_{ijk}$, but by (12) $y_{jk} \geq x_{ijk}$. Both constraints are removed and replaced by $y_{jk} = x_{ijk}$.

Similarly, we conduct the following simplification for model M3

Algorithm 1 Cut generation

$stop = True$
for $i \in \mathcal{I}$ **and** $stop = True$ **do**
  $S = \{(j,k) | x_{ijk} > 0 \ (j,k) \in \mathcal{T}_i\}$
  $jk^i_{\min} = \arg\min_{(j,k)\in S}\{\sigma_i(j,k)\}$
  $jk^i_{\max} = \arg\max_{(j,k)\in S}\{\sigma_i(j,k)\}$
  if $y_{(j,k)^i_{\max}} < 1$ **then**
    for $i' \in \mathcal{I} \setminus \{i\}$ **do**
      $(j,k)^{i'}_{\max} = \arg\max_{(j,k)\in S\cap\mathcal{T}_{i'}} \{\sigma_i(j,k)\}$
      for $(\hat{j},\hat{k}) \in \mathcal{T}_i \cap [(j,k)^i_{\min}, (j,k)^{i'}_{\max}]$ **do**
        if $\sum_{(j',k')\in\mathcal{W}_{i\hat{j}\hat{k}}} x_{ij'k'} = y_{(j,k)^i_{max}}$ **then**
          break
        if $\sum_{(j',k')\in\mathcal{W}_{i\hat{j}\hat{k}}} x_{ij'k'} + \sum_{(j',k')\in\mathcal{B}_{ii'\hat{j}\hat{k}}} x_{i'j'k'} \leq 1$ **then**
          add cut$(i, i', \hat{j}, \hat{k})$
          $stop = False$
          break

- If $|\mathcal{W}_{ii'jk}| = 0$ and $|\mathcal{W}_{ijk}| = 0$ then the pair (j,k) is the worst for client i. Constraint (18) becomes $y_{jk} \leq 1$, and it can be removed.
- If $|\mathcal{W}_{ii'jk}| = 0$ and $|\mathcal{W}_{ijk}| \neq 0$ then constraint (18) becomes equal to constraint (17) and can be removed.
- If $|\mathcal{W}_{ii'jk}| \neq 0$ and $|\mathcal{W}_{ijk}| = 0$ then constraint (18) becomes equal to constraint (17) and can be removed.
- If $|\mathcal{W}_{ijk}| = |\mathcal{T}_i| - 1$ and $|\mathcal{W}_{ii'jk}| = 0$ then the pair (j,k) is the best option for client i. Constraint (18) becomes $y_{jk} \leq x_{ijk}$, but by (12) $y_{jk} \geq x_{ijk}$, so both are removed and replaced by $y_{jk} = x_{ijk}$.

4. Computational experiments

In this section we present the computational tests performed. All procedures and algorithms have been written using the language Python and, for MIP

problems, we used CPLEX version 12.9. We begin describing the synthetic data that was used in our experiments. Our computational results explore the strength of the different formulations and the use of existing Benders decomposition strategies (Subsection 4.2), the effectiveness of the constraints (22) in Subsection 4.3, and the comparison of the proposed decomposition methods and a benchmark method on the multiple location problem (Subsection 4.4).

4.1. Instances: description

The data sets were adapted from Beasley's OR-Library (Beasley, 1990). Values d_{ij}, r_k and r_i^k were build based on data files *pmed10*, *pmed25* for the uncapacitated warehouse location problem. $|\mathcal{I}|$ nodes were selected randomly as clients. To select $|\mathcal{J}|$ nodes as stores we solved an uncapacitated p-median problem. The distances between each client i and each store j are immediate to obtain. The remaining parameters were determined as follows: $\pi_k \in [1.5\overline{d}, 5\overline{d}]$ and $r_i^k \in [2\underline{d_i} + \overline{\pi}, 2\overline{d_i} + \overline{\pi}]$, where $\overline{d}$ is the average of all distances, $\overline{\pi}$ is the average of all products prices; $\underline{d_i}$ and $\overline{d_i}$ are the distances between the client i and the closest and farthest store respectively. Finally, we fixed a capacity p_j arbitrarily for each store. Without loss of generality, we fixed all stores with equal capacity, i.e $p = p_j$.

For the number of customers, stores, products, and capacities, we used the values $|\mathcal{I}| = 100, 160, 250, 400$, $|\mathcal{J}| = 4, 8, 12$, $|\mathcal{K}| = 20, 30, 40, 50, 80$ and $p_j = p = 5, 10\ \forall j \in \mathcal{J}$.

Each instance so defined, was run using five different scenarios. Table 4.1 shows statistics of the generated instances, obtained over the five scenarios. The Table is divided in three blocks displaying the same statistics for 4, 8 and 12 stores. Each block is divided in seven columns. The first three show minimum, average and maximum number of clients (i) per pair (j, k), i.e. (product, store), while columns four, five and six show the converse. Column seven shows the average number of clients with the same set $\mathcal{T}_i$.

We performed several experiments to compare our formulations and methods with each other, and to compare these with the Benders Decomposition approach.

Table 4: Statistics of generated instances

		4 stores							stores 8							stores 12						
		clients x (j,k)			items x clients			same clients	clients x items			items x clients			same clients	clients x items			items x clients			same clients
		min	avg	max	min	avg	max		min	avg	max	min	avg	max		min	avg	max	min	avg	max	
100	20	12,2	20,8	28,4	12,2	16,6	28,0	0,6	12,0	20,5	29,2	12,0	32,8	84,0	0,2	11,2	20,5	28,8	11,2	32,8	56,0	0,2
	30	11,4	20,6	29,8	11,4	24,7	40,0	0,0	11,0	20,7	29,6	11,0	49,8	120,0	0,0	11,0	20,7	30,4	11,0	49,6	80,0	0,2
	40	10,8	20,7	31,4	10,8	33,2	55,6	0,0	9,2	20,3	30,2	9,2	64,9	164,8	0,0	8,6	20,3	30,8	8,6	64,9	109,6	0,0
	50	12,2	21,5	31,6	12,2	43,0	72,0	0,0	8,4	20,8	33,8	8,4	83,3	209,8	0,0	9,8	20,8	32,6	9,8	83,3	143,6	0,0
	80	10,2	21,1	34,6	10,2	67,4	110,4	0,0	7,6	20,3	31,6	7,6	130,1	327,4	0,0	8,8	20,3	32,2	8,8	130,1	216,4	0,0
160	20	22,8	34,1	45,4	22,8	17,1	28,0	1,6	20,2	33,6	45,4	20,2	33,6	84,0	1,0	19,6	33,6	45,8	19,6	33,6	56,0	1,4
	30	19,6	32,1	43,0	19,6	24,1	40,0	0,2	17,8	30,9	43,0	17,8	46,3	120,0	0,0	17,8	30,9	42,6	17,8	46,3	79,8	0,0
	40	21,4	33,7	46,0	21,4	33,7	56,0	0,0	15,2	32,5	45,4	15,2	65,1	167,4	0,0	18,6	32,5	46,8	18,6	65,1	110,6	0,0
	50	22,2	34,3	46,4	22,2	42,9	72,0	0,0	16,6	33,5	45,8	16,6	83,7	213,0	0,0	19,2	33,5	46,8	19,2	83,7	143,6	0,0
	80	21,6	34,2	49,4	21,6	68,4	111,4	0,0	15,4	33,5	48,0	15,4	133,8	326,0	0,0	17,8	33,5	46,4	17,8	133,8	217,6	0,0
250	20	30,8	52,3	67,2	30,8	16,7	28,0	3,4	33,4	51,3	66,6	33,4	32,8	84,0	3,6	31,4	51,3	67,4	31,4	32,8	56,0	2,6
	30	30,4	49,6	66,2	30,4	23,8	40,0	0,0	28,2	49,4	67,0	28,2	47,4	120,0	0,0	30,0	49,4	66,8	30,0	47,4	80,0	0,0
	40	33,2	50,8	71,4	33,2	32,5	56,0	0,0	28,8	50,2	69,8	28,8	64,2	168,0	0,0	31,6	50,2	69,0	31,6	64,2	112,0	0,0
	50	32,6	52,5	73,2	32,6	42,0	72,0	0,0	27,0	52,6	71,4	27,0	84,2	214,2	0,0	28,0	52,6	71,2	28,0	84,2	143,0	0,0
	80	28,6	51,3	72,0	28,6	65,7	109,8	0,0	27,0	51,7	71,2	27,0	132,4	327,4	0,0	29,6	51,7	71,0	29,6	132,4	220,4	0,0
400	20	56,6	81,0	103,4	56,6	16,2	28,0	10,0	51,2	79,6	103,8	51,2	31,9	84,0	8,0	53,0	79,6	103,4	53,0	31,9	56,0	10,8
	30	51,2	78,5	100,6	51,2	23,6	40,0	0,4	50,4	78,7	104,2	50,4	47,2	120,0	0,2	49,2	78,7	103,4	49,2	47,2	80,0	0,4
	40	52,8	80,7	105,2	52,8	32,3	56,0	0,0	47,2	80,2	104,4	47,2	64,1	168,0	0,0	52,0	80,2	108,2	52,0	64,1	112,0	0,0
	50	54,2	83,7	108,0	54,2	41,8	71,8	0,0	52,8	83,6	107,6	52,8	83,6	216,0	0,0	52,4	83,6	106,6	52,4	83,6	143,8	0,0
	80	52,8	82,3	107,4	52,8	65,8	111,6	0,0	49,8	82,0	108,6	49,8	131,2	333,6	0,0	49,6	82,0	107,8	49,6	131,2	222,2	0,0

In the results below we denote each instances with its problem size, as either "$|\mathcal{I}|\_|\mathcal{J}|\_|\mathcal{K}|$" or "$|\mathcal{I}|\_|\mathcal{K}|$" depending on what is being compared with. For each problem we consider 5 different random instances, all results presented are the average over these 5 instances.

The headers of the tables presenting the computational results use the following nomenclature:

- $|\mathcal{I}|\_|\mathcal{J}|\_|\mathcal{K}|$ or $|\mathcal{I}|\_|\mathcal{K}|$: Name of instance.
- GAP (%) : average gap $(\frac{UB-LB}{LB}) \times 100\%$, where UB(LB) is the best upper(lower) bound.
- GAP_{LP} (%) : average integrality gap $(\frac{LP-LB}{LB}) \times 100\%$, with LP = linear relaxation value
- $TIME$: average CPU time in seconds (Total time)
- $TIME_{LP}$: average CPU time in seconds of the linear relaxation
- Ni : number of instances solved to full optimality

4.2. Efficiency of problem formulation and benders decomposition on multi store instances

Table 5 shows the comparison between M2, M3 and M4 in terms of GAP, GAP_{LP} and $TIME$, using the default CPLEX's Branch & Cut. The formulations M3 and M4 use the subset of constraints described in Section 3. The results show that M4 dominates M3, which in turn, dominates M2 in terms of GAP. Except for three instances, M4 dominates M3 in GAP_{LP}, and M3 always dominates M2, suggesting that M4 has the tightest LP relaxation. In terms of $TIME$, there is no clear winner.

Table 5: Comparison of GAP, GAP_{LP} and $TIME$ between the formulations M2, M3 and M4

	M2			M3			M4		
$\lvert\mathcal{I}\rvert$-$\lvert\mathcal{J}\rvert$-$\lvert\mathcal{K}\rvert$	GAP	GAP_{LP}	$TIME$	GAP	GAP_{LP}	$TIME$	GAP	GAP_{LP}	$TIME$
100-4-20	0,00	4,18	25	0,00	2,31	23	0,00	2,55	23
100-4-30	0,00	3,06	62	0,00	2,22	62	0,00	2,13	41
100-4-40	0,00	3,07	244	0,00	2,67	224	0,00	2,38	110
100-4-50	0,00	1,82	423	0,00	1,60	113	0,00	1,49	86
100-4-80	0,00	1,41	752	0,00	1,31	346	0,00	1,16	335
160-4-20	0,00	5,17	33	0,00	2,61	56	0,00	2,59	54
160-4-30	0,00	3,89	278	0,00	2,92	348	0,00	2,72	174
160-4-40	0,00	3,29	1220	0,00	2,71	675	0,00	2,47	611
160-4-50	0,29	2,48	2124	0,09	2,23	1513	0,00	2,06	1049
160-4-80	0,67	1,94	3094	0,25	1,76	2961	0,12	1,60	2455
250-4-20	0,00	7,48	99	0,00	3,10	48	0,00	3,11	78
250-4-30	0,00	5,64	615	0,00	3,66	417	0,00	3,25	510
250-4-40	0,00	4,40	1389	0,00	3,62	1365	0,00	3,24	1073
250-4-50	2,06	5,11	3600	1,78	4,27	3600	0,78	3,75	3050
250-4-80	2,78	4,20	3600	2,50	3,59	3600	2,23	3,62	3600
400-4-20	0,00	8,17	269	0,00	3,30	286	0,00	3,17	556
400-4-30	0,49	6,33	1882	0,00	3,65	2013	0,00	3,40	2306
400-4-40	2,25	5,77	3600	1,83	4,28	3600	1,11	3,61	3600
400-4-50	3,13	5,16	3600	2,72	4,23	3600	1,74	3,57	3600
400-4-80	3,75	4,72	3600	3,22	3,87	3600	2,26	3,02	3600

In Table 6 we compare the results obtained by different existing solution methods on formulation M2. Column M2 indicates default CPLEX as before, column M2-CPLEX-BD is CPLEX with its available Benders decomposition strategy, and M2-BD uses the algorithm proposed in (Bertsimas and Mišić, 2019). In the largest instances, the best results were those of M2-BD while

for medium and small instances, the best results were obtained with M2-CPLEX-BD.

Table 6: Computational results of model M2

	M2-CPLEX-BD			M2-BD			M2		
$\lvert\mathcal{I}\rvert\_\lvert\mathcal{J}\rvert\_\lvert\mathcal{K}\rvert$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$
100-4-20	0.00	5	25	0.01	5	18	0.00	5	36
100-4-30	0.00	5	62	0.01	5	67	0.00	5	81
100-4-40	0.00	5	244	0.01	5	671	0.00	5	301
100-4-50	0.00	5	423	0.01	5	441	0.00	5	131
100-4-80	0.00	5	752	0.01	5	429	0.00	5	428
160-4-20	0.00	5	33	0.01	5	32	0.00	5	67
160-4-30	0.00	5	278	0.01	5	595	0.00	5	536
160-4-40	0.00	4	1.220	0.15	4	1.601	0.01	4	1.406
160-4-50	0.29	3	2.124	0.47	2	2.260	0.17	3	2.204
160-4-80	0.67	1	3.094	0.57	1	3.087	0.47	1	2.964
250-4-20	0.00	5	99	0.01	5	163	0.00	5	173
250-4-30	0.00	5	615	0.07	4	1.508	0.05	4	1.152
250-4-40	0.00	5	1.389	0.47	4	2.813	0.29	3	2.239
250-4-50	2.06	-	3.600	2.18	-	3.600	1.86	-	3.600
250-4-80	2.78	-	3.600	2.26	-	3.600	2.70	-	3.600
400-4-20	0.00	5	269	0.01	5	273	0.00	5	487
400-4-30	0.49	4	1.882	0.88	2	3.440	0.25	4	2.492
400-4-40	2.25	-	3.600	2.16	-	3.600	2.19	-	3.600
400-4-50	3.13	-	3.600	2.55	-	3.600	2.89	-	3.600
400-4-80	3.75	-	3.600	2.64	-	3.600	3.13	-	3.600

Comparing Tables 5 and 6, it is clear that the method that dominates in both gap and time (except for instance 400-4-20), is M4. This method even outperforms the Benders decomposition approaches on multiple location instances.

4.3. The effectiveness of constraints (22)

To assess the effect of constraints (22), we added all of them to problem M3 for small instances and compare the results in terms of gap, linear relaxation run time, and total run time with solving problem M3 without these

constraints. We slightly modified M3 by adding all constraints (18), rather than a subset of them. We call this model M3*. We set the product capacity of all stores to two. Table 7 compares model M3* with and without constraints (22). Adding these constraints clearly reduces GAP_{LP} and, although $TIME_{LP}$ increases, the total solution time $TIME$ is reduced in the largest instances. We remark that we did not use model M4 in this comparison, due to the high computational cost of generating all constraints (19).

Table 7: Comparison of GAP_{LP}, TIME and $TIME_{LP}$ between formulations M3 and M3 plus (22). Store capacity p = 5

	M3*			M3* + (22)		
$\|\mathcal{I}\|$-$\|\mathcal{J}\|$-$\|\mathcal{K}\|$	GAP_{LP}	$TIME$	$TIME_{LP}$	GAP_{LP}	$TIME$	$TIME_{LP}$
30-4-5	1.14	0.03	0.01	0.00	0.05	0.01
35-4-15	0.56	0.71	0.09	0.13	1.26	0.28
40-4-25	1.41	3.52	0.27	0.11	2.11	0.98
50-4-30	0.77	14.07	0.82	0.19	17.73	3.71
50-4-40	0.92	27.02	1.24	0.35	29.64	4.55
50-8-30	1.25	150.81	4.48	0.25	92.25	13.07
50-8-35	0.89	659.93	12.67	0.04	169.12	35.71
50-8-40	0.84	1127.90	15.61	0.14	365.75	46.05
50-8-45	0.92	1010.72	15.90	0.27	518.06	41.39
50-8-50	0.39	665.13	19.29	0.02	298.21	61.65

In Table 8 we explore the efficiency of valid inequalities (22) on large instances. In this case we consider model M4 incorporating inequalities (22) using $C\&B$ (with CPLEX default cuts turned off) and compare its gap, number of solved instances, and solution time with that of solving model M4 and model M4^o (with CPLEX default cuts off). We observe that model M4^o obtains a similar efficiency to model M4, showing that CPLEX default inequalities do not significantly influence the solution of model M4 and that $C\&B$, adding cuts (22) at the root node, dominates both M4 and M4^o.

We finish the evaluating the impact of constraint (22) solving the single

[1]Two instances reached the $TIME$ limit before solving the root node; thus, there is no GAP in these instances. Therefore, the GAP average reported in the Table is the average of the three remaining instances.

Table 8: Comparison of valid inequality (22) with default CPLEX cuts and $C\&B$. Store capacity $p = 5$

	$\|\mathcal{J}\| = 4$						$\|\mathcal{J}\| = 8$						$\|\mathcal{J}\| = 12$					
	C&B		M4º		M4		*C&B*		M4º		M4		*C&B*		M4º		M4	
$\|\mathcal{I}\|$_$\|\mathcal{K}\|$	*GAP*	*TIME*	*GAP*	*TIME*	*GAP*	*TIME*	*GAP*	*TIME*	*GAP*	*TIME*	*GAP*	*TIME*	*GAP*	*TIME*	*GAP*	*TIME*	*GAP*	*TIME*
100-20	0,00	2	0,00	20	0,00	23	0,00	3	0,08	1078	0,00	753	0,00	7	0,46	1934	0,41	1783
100-30	0,00	3	0,00	34	0,00	41	0,00	10	0,10	1650	0,00	1398	0,00	13	0,53	2913	0,59	3357
100-40	0,00	8	0,00	134	0,00	110	0,00	15	0,37	1900	0,38	2132	0,00	31	1,49	2649	1,69	3025
100-50	0,00	10	0,00	91	0,00	86	0,00	21	0,04	1606	0,03	1445	0,00	17	0,14	2364	0,14	2715
100-80	0,00	21	0,00	215	0,00	335	0,00	46	0,35	3125	0,36	3172	0,00	50	0,47	3600	0,54	3600
160-20	0,00	5	0,00	43	0,00	54	0,00	10	0,16	2304	0,28	2285	0,00	16	1,21	3367	1,29	3496
160-30	0,00	13	0,00	188	0,00	174	0,00	21	0,65	3460	0,72	3318	0,00	37	1,99	3600	2,01	3600
160-40	0,00	55	0,00	391	0,00	611	0,00	52	2,21	3600	2,16	3600	0,00	172	3,15	3600	3,21	3600
160-50	0,00	33	0,00	1148	0,00	1049	0,00	71	1,06	3308	0,87	3301	0,00	239	2,67	3600	2,98	3600
160-80	0,00	247	0,18	2435	0,12	2455	0,00	474	1,51	3600	1,73	3600	0,00	733	2,18	3600	2,81	3600
250-20	0,00	14	0,00	68	0,00	78	0,00	33	0,29	1223	0,14	1676	0,00	51	2,16	3600	2,38	3600
250-30	0,00	37	0,00	364	0,00	510	0,00	331	2,17	3550	2,35	3600	0,00	297	3,50	3600	3,83	3600
250-40	0,00	45	0,00	921	0,00	1073	0,00	661	2,35	3600	2,07	3600	0,00	318	3,89	3600	4,49	3600
250-50	0,00	718	0,72	3229	0,78	3050	0,00	491	3,33	3600	3,41	3600	0,00	1166	5,32	3600	5,01	3601
250-80	0,00	944	1,68	3600	2,23	3600	0,03	1703	3,40	3600	3,83	3601	0,12	2398	4,10	3601	11,90	3600
400-20	0,00	50	0,00	386	0,00	556	0,00	257	3,17	3029	2,85	3279	0,00	260	4,77	3600	5,00	3600
400-30	0,00	123	0,00	2189	0,00	2306	0,00	607	3,89	3600	3,64	3600	0,00	1044	5,73	3600	4,94	3601
400-40	0,00	308	0,99	3600	1,11	3600	0,00	636	4,10	3600	3,76	3600	0,03	1224	4,48	3600	4,24	3601
400-50	0,00	609	1,83	3600	1,74	3600	0,03	2360	3,68	3600	3,67	3601	2,71	2480	4,28	3601	4,10	3600
400-80	0,15	2745	2,55	3600	2,26	3600	0,48	3600	18,71	3601	25,45	3600	14,11 [1]	3600	16,21	3600	17,66	3600

store instances in (Bertsimas and Mišić, 2019) with B&C and C&B and comparing it to the Benders' method proposed in that paper. As Table 9 shows, all methods arrived at the optimal solution. For the small and medium instances, the C&B obtained the best results. When the number of clients exceeds 500 and the number of products is 50 or more, the Benders' method solves most of the instances faster.

4.4. Evaluation of decomposition methods on multi store instances

Considering that M2 with the Benders Decomposition approach of (Bertsimas and Mišić, 2019) is the best strategy for the one-store problem, we compare its results with the $B\&C$ and $C\&B$ approaches applied to model M4 for multiple store problems. Table 10 and Table 11 show the average GAP and $TIME$ over 5 instances for each $|\mathcal{I}|\_|\mathcal{J}|\_|\mathcal{K}|$ combination. In the three methods, we deactivate any cuts generated by default in CPLEX, and used as a cut the valid inequality (22). The best results were obtained by $C\&B$ in terms of GAP and $TIME$. In fact, this approach obtained the optimal solution in most instances.

[2]Two instances reached the $TIME$ limit before solving the root node; thus, there is no GAP in these instances. Therefore, the GAP average reported in the Table is the average of the three remaining instances.

Table 9: Comparison of run TIME of instances (Bertsimas and Mišić, 2019). Stores number $|\mathcal{J}| = 1$

	p = 5			p = 10			p = ∞		
$\|\mathcal{I}\|$_$\|\mathcal{K}\|$	$M2-BD$	$B\&C$	$C\&B$	$M2-BD$	$B\&C$	$C\&B$	$M2-BD$	$B\&C$	$C\&B$
100-20	0.67	0.17	0.10	0.62	0.17	0.08	0.64	0.17	0.07
200-20	1.25	0.51	0.29	1.19	0.55	0.33	1.13	0.51	0.19
500-20	3.00	8.97	1.27	2.90	3.94	0.71	2.69	4.22	0.60
1000-20	5.57	21.53	2.75	5.43	12.05	1.98	4.85	11.92	1.69
100-50	1.81	1.69	0.67	2.21	0.47	0.40	2.19	0.43	0.31
200-50	9.39	7.20	2.49	4.53	1.94	1.59	3.98	1.52	1.18
500-50	9.39	71.62	34.36	10.75	24.03	11.14	9.04	20.26	7.90
1000-50	20.14	323.08	328.47	23.20	128.84	112.09	19.17	71.83	101.68
100-100	5.48	6.27	3.39	7.73	2.03	2.02	8.23	1.65	1.35
200-100	14.27	30.66	20.15	45.81	20.97	11.7	13.25	9.68	6.58
100-200	19.08	21.52	18.29	24.50	18.98	15.33	30.91	10.28	7.97
200-200	33.22	94.98	77.72	50.83	65.40	54.24	55.11	35.07	23.92
500-200	73.76	1295.61	1154.98	194.86	598.24	577.19	128.45	315.86	248.01
1000-200	205.63	3600.67	7550.79	482.51	3029.24	6379.81	407.56	1571.28	2229.27
100-500	90.56	132.34	107.86	123.72	98.67	77.63	216.71	56.34	38.14
200-500	189.44	593.00	722.58	314.71	381.79	395.58	527.32	225.23	241.53
500-500	448.54	3523.83	3492.77	1106.05	3024.71	2748.98	1114.97	1634.18	954.68
500-100	25.16	344.96	301.00	32.65	104.55	122.22	30.20	57.94	73.91
1000-100	60.73	1733.49	2339.39	54.41	367.52	334.23	51.57	283.97	223.27

Table 10: Comparison of GAP and run $TIME$s. Store capacity $p = 5$

	$\|\mathcal{J}\|$= 4									$\|\mathcal{J}\|$ = 8									$\|\mathcal{J}\|$ = 12								
	C&B			M2-BD			B&C			C&B			M2-BD			B&C			C&B			M2-BD			B&C		
$\|\mathcal{I}\|$_$\|\mathcal{K}\|$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$
100-20	0,00	5	2	0,01	5	17	0,00	5	17	0,00	5	3	0,01	5	1214	0,14	4	1135	0,00	5	7	1,13	1	3224	0,61	3	2571
100-30	0,00	5	3	0,01	5	91	0,00	5	36	0,00	5	10	1,17	1	3026	0,07	3	1979	0,00	5	13	1,48	0	3600	0,33	3	2968
100-40	0,00	5	8	0,05	4	1009	0,00	5	127	0,00	5	15	1,70	2	2937	0,37	3	1789	0,00	5	31	2,58	0	3600	1,60	2	2609
100-50	0,00	5	10	0,07	4	794	0,00	5	65	0,00	5	21	0,92	1	3212	0,03	4	1536	0,00	5	17	2,46	1	3440	0,13	3	2457
100-80	0,00	5	21	0,01	5	1274	0,01	5	298	0,00	5	46	0,77	1	2931	0,39	2	3099	0,00	5	50	0,87	0	3600	0,42	0	3600
160-20	0,00	5	5	0,01	5	33	0,00	5	37	0,00	5	10	2,41	0	3600	0,23	3	2282	0,00	5	16	3,57	0	3600	1,38	1	3368
160-30	0,00	5	13	0,01	5	710	0,00	5	185	0,00	5	21	3,78	0	3600	0,75	1	3407	0,00	5	37	3,86	0	3600	2,10	0	3600
160-40	0,00	5	55	0,25	3	1861	0,00	5	443	0,00	5	52	2,14	0	3600	2,25	0	3600	0,00	5	172	1,30	0	3600	2,80	0	3600
160-50	0,00	5	33	0,28	2	2273	0,00	5	1069	0,00	5	71	1,19	0	3600	1,02	1	3373	0,00	5	239	3,21	0	3600	2,50	0	3600
160-80	0,00	5	247	0,61	1	3165	0,11	3	2156	0,00	5	474	1,93	0	3600	1,43	0	3600	0,00	5	733	2,77	0	3600	2,47	0	3600
250-20	0,00	5	14	0,01	5	124	0,00	5	61	0,00	5	33	1,09	2	2974	0,21	4	1325	0,00	5	51	4,30	0	3600	1,87	0	3600
250-30	0,00	5	37	0,23	4	1486	0,00	5	301	0,00	5	331	4,49	0	3600	2,18	0	3600	0,00	5	297	5,08	0	3600	3,59	0	3600
250-40	0,00	5	45	0,70	2	3021	0,00	5	1043	0,00	5	661	4,84	1	2963	2,49	0	3600	0,00	5	318	4,85	0	3600	4,07	0	3600
250-50	0,00	5	718	2,34	0	3600	0,72	2	3449	0,00	5	491	4,05	0	3600	3,37	0	3600	0,00	5	1166	6,68	0	3600	4,92	0	3600
250-80	0,00	5	944	2,31	0	3600	1,74	0	3600	0,03	4	1703	5,78	0	3600	3,02	0	3600	0,12	3	2398	7,11	0	3600	4,32	0	3600
400-20	0,00	5	50	0,01	5	377	0,00	5	264	0,00	5	257	5,49	0	3600	3,12	1	2988	0,00	5	260	4,42	0	3600	4,82	1	3404
400-30	0,00	5	123	1,05	0	3600	0,00	5	1715	0,00	5	607	4,27	0	3600	3,91	0	3600	0,00	4	1044	5,50	0	3600	5,72	0	3600
400-40	0,00	5	308	2,48	0	3600	0,80	0	3600	0,00	5	636	4,88	0	3600	4,19	0	3600	0,03	4	1224	5,77	0	3600	4,48	0	3600
400-50	0,00	5	609	2,70	0	3600	1,66	0	3600	0,03	3	2360	3,27	0	3600	3,71	0	3600	2,71	3	2480	4,26	0	3600	4,61	0	3600
400-80	0,15	2	2745	2,56	0	3600	2,56	0	3600	0,48	0	3600	5,97	0	3600	9,23	0	3600	14,11[2]	0	3600	7,38	0	3600	17,74	0	3600

As Table 10 and Table 11 show, as the store capacity increases, the time required to solve the problem tends to decrease. The problem is solved to full optimality within the one-hour time limit in almost all cases using the B&C and C&B approaches, except in the few cases of the largest instances, where these two methods are able to solve one more instance than the M2-BD.

Table 11: Comparison of GAP and run $TIME$s. Store capacity $p = 10$

	$\|\mathcal{J}\| = 4$									$\|\mathcal{J}\| = 8$									$\|\mathcal{J}\| = 12$								
	C&B			M2-BD			B&C			C&B			M2-BD			B&C			C&B			M2-BD			B&C		
$\|\mathcal{I}\|$-$\|\mathcal{K}\|$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$	GAP	Ni	$TIME$
100-20	0,00	5	1	0,00	5	0	0,00	5	15	0,00	5	3	0,00	5	2	0,10	4	966	0,00	5	6	0,00	5	7	0,55	3	2585
100-30	0,00	5	2	0,00	5	1	0,00	5	26	0,00	5	8	0,01	5	12	0,11	3	1892	0,00	5	11	0,00	5	16	0,51	2	2811
100-40	0,00	5	5	0,00	5	4	0,00	2	94	0,00	5	11	0,00	5	16	0,35	v3	1756	0,00	5	25	0,00	5	134	1,64	2	2616
100-50	0,00	5	4	0,00	5	2	0,00	5	37	0,00	5	13	0,00	5	29	0,02	4	1427	0,00	5	17	0,00	5	8	0,13	3	1952
100-80	0,00	5	9	0,00	5	18	0,00	5	143	0,00	5	29	0,00	5	5	0,32	2	2872	0,00	5	36	0,00	5	37	0,42	0	3600
160-20	0,00	5	3	0,00	5	2	0,00	5	27	0,00	5	9	0,01	5	10	0,14	3	2204	0,00	5	16	0,00	5	53	1,29	1	3428
160-30	0,00	5	5	0,00	5	3	0,00	5	102	0,00	5	15	0,00	5	117	0,73	2	3532	0,00	5	41	0,00	5	193	2,12	0	3600
160-40	0,00	5	18	0,00	5	34	0,00	5	401	0,00	5	47	0,00	5	41	2,06	0	3600	0,00	5	121	0,01	5	161	3,05	0	3600
160-50	0,00	5	13	0,00	5	11	0,00	5	574	0,00	5	37	0,01	5	266	0,95	1	2193	0,00	5	146	0,01	5	430	2,51	0	3600
160-80	0,00	5	46	0,01	5	125	0,09	4	1847	0,00	5	204	0,06	4	1358	1,36	0	3600	0,00	5	331	0,13	2	2840	2,42	0	3600
250-20	0,00	5	9	0,00	5	3	0,00	5	39	0,00	5	29	0,00	5	10	0,15	4	1275	0,00	5	49	0,00	5	27	2,18	0	3600
250-30	0,00	5	25	0,00	5	11	0,00	5	187	0,00	5	339	0,00	5	110	2,18	1	3533	0,00	5	316	0,01	5	324	3,57	0	3600
250-40	0,00	5	23	0,00	5	23	0,00	5	612	0,00	5	257	0,00	5	122	2,23	0	3600	0,00	5	222	0,00	5	676	4,44	0	3600
250-50	0,00	5	213	0,00	5	420	0,56	3	2924	0,00	5	313	0,00	5	382	3,15	0	3600	0,00	5	651	0,00	5	210	5,12	0	3600
250-80	0,00	5	212	0,01	5	532	0,93	0	3600	0,00	5	843	0,79	5	1062	2,52	0	3600	0,03	4	1431	0,01	4	1909	3,34	0	3600
4000-20	0,00	5	48	0,00	5	17	0,00	5	191	0,00	5	256	0,00	5	335	3,05	1	2978	0,00	5	251	0,11	3	1790	4,81	1	3395
4000-30	0,00	5	87	0,00	5	49	0,01	4	1750	0,00	5	536	0,00	5	888	3,95	0	3600	0,00	5	981	0,08	3	2130	5,69	0	3600
4000-40	0,00	5	156	0,00	5	91	0,47	1	3503	0,00	5	333	0,19	2	2423	3,94	0	3600	0,03	4	1115	0,32	1	3257	4,54	0	3600
4000-50	0,00	5	424	0,01	5	394	1,16	1	3556	0,02	4	1359	0,32	2	2539	3,77	0	3600	0,12	3	2320	1,23	1	3197	4,41	0	3600
4000-80	0,00	5	733	0,00	5	932	1,34	0	3600	0,30	2	2715	6,44	0	4012	10,61	0	3600	7,42	1	3576				12,28	0	3600

5. Conclusions

Most results in the literature dealing with product line optimization, also called assortment planning problem, do not consider that a firm can have more than one store with different products, and that clients can patronize different stores, if they are within reasonable distances. Neither they consider the travel cost of the clients. Previous works consider that the firm has only one store, so the geographical dimension is absent. The models proposed in this paper incorporate the choice of store and product and the cost of travel as part of the utility of the clients, when a firm has more that one store. The clients can decide to travel farther away to buy at a lower price, or a product that is unavailable at their closest store. This behavior cannot be represented by previously published models and, in our case, leads to assortments that capture more customers. A bi-level formulation is proposed that considers the interaction between the choice of the client and the firm.

In our bi-level model, the first level is the retailer problem and the second level is the purchaser problem. Due to its structure, we can collapse it into a single level formulation. Three different single level formulations are proposed, which are equivalent but possess different tightness characteristics

We adapted valid inequalities of the FLPCP to our problem to improve the LP relaxation. As the number of valid inequalities is large, we use $B\&C$ and $C\&B$ strategies to solve the problem. The numerical experiments were done using published data, including that in (Bertsimas and Mišić, 2019). These

experiments show that our approach not only solves a previously unsolved problem, but it also obtains better results in synthetic data than the best approach known so far, proposed in (Bertsimas and Mišić, 2019). For one store, using synthetic data in (Bertsimas and Mišić, 2019), the Benders decomposition algorithm solved faster the instances with 500 and 1000 clients.

There are several directions to extend the research in this paper. For simplicity reasons, it was reasonable to consider here that clients buy at most one unit of one product. In general, however, it is more common that clients purchase different products. It is also interesting to consider the case in which clients buy a bundle of products. There are several possibilities: the clients can buy all of the products in the bundle in one store or in different stores and incorporate the travel cost in their decision.

6. Acknowledgements

The authors gratefully acknowledge the support by Grants FONDECYT 1190064, CONICYT PIA AFB180003 and INRIA Associated Team BIPLOS. Also, to CONICYT for a Doctorate fellowship for Sebastián Dávila Folio 21161328/2016 . Martine Labbé has been partially supported by the Fonds de la Recherche Scientifique -FNRS under Grant PDR T0098.18

References

N. Aras and H. Küçükaydın. Bilevel models on the competitive facility location problem. In *Spatial Interaction Models*, pages 1–19. Springer, 2017.

J. E. Beasley. Or-library: distributing test problems by electronic mail. *Journal of the operational research society*, 41(11):1069–1072, 1990.

A. Belloni, R. Freund, M. Selove, and D. Simester. Optimizing product line designs: Efficient methods and comparisons. *Management Science*, 54(9): 1544–1552, 2008.

D. Bertsimas and V. V. Mišić. Exact first-choice product line optimization. *Operations Research*, 2019.

O. Besbes and D. Sauré. Product assortment and price competition under multinomial logit demand. *Production and Operations Management*, 25 (1):114–127, 2016.

A. Bhatnagar and S. S. Syam. Allocating a hybrid retailer's assortment across retail stores: Bricks-and-mortar vs online. *Journal of Business Research*, 67(6):1293–1302, 2014.

P. Briest and P. Krysta. Buying cheap is expensive: Approximability of combinatorial pricing problems. *SIAM Journal on Computing*, 40(6):1554–1586, 2011.

L. Cánovas, S. García, M. Labbé, and A. Marín. A strengthened formulation for the simple plant location problem with order. *Operations Research Letters*, 35(2):141–150, 2007.

M. Chen, Z.-L. Chen, G. Pundoor, S. Acharya, and J. Yi. Markdown optimization at multiple stores. *IIE Transactions*, 47(1):84–108, 2015.

M.-C. Chen and C.-P. Lin. A data mining approach to product assortment and shelf space allocation. *Expert Systems with Applications*, 32(4):976–986, 2007.

G. Dobson and S. Kalish. Heuristics for pricing and positioning a product-line using conjoint and cost data. *Management Science*, 39(2):160–175, 1993.

H. A. Eiselt, V. Marianov, and T. Drezner. Competitive location models. In *Location science*, pages 365–398. Springer, 2015.

T. Flamand, A. Ghoniem, M. Haouari, and B. Maddah. Integrated assortment planning and store-wide shelf space allocation: An optimization-based approach. *Omega*, 2017.

A. Ghoniem and B. Maddah. Integrated retail decisions with multiple selling periods and customer segments: optimization and insights. *Omega*, 55: 38–52, 2015.

A. Ghoniem, B. Maddah, and A. Ibrahim. Optimizing assortment and pricing of multiple retail categories with cross-selling. *Journal of Global Optimization*, 66(2):291–309, 2016.

P. E. Green and A. M. Krieger. Models and heuristics for product line selection. *Marketing Science*, 4(1):1–19, 1985.

P. E. Green and A. M. Krieger. Recent contributions to optimal product positioning and buyer segmentation. *European Journal of Operational Research*, 41(2):127–141, 1989.

P. Hansen, Y. Kochetov, and N. Mladenovi. *Lower bounds for the uncapacitated facility location problem with user preferences*. Groupe d'études et de recherche en analyse des décisions, HEC Montréal, 2004.

A. Hübner and K. Schaal. An integrated assortment and shelf-space optimization model with demand substitution and space-elasticity effects. *European Journal of Operational Research*, 261(1):302–316, 2017.

A. H. Hübner and H. Kuhn. Retail category management: State-of-the-art review of quantitative research and software applications in assortment and shelf space management. *Omega*, 40(2):199–209, 2012.

A. G. Kök, M. L. Fisher, and R. Vaidyanathan. Assortment planning: Review of literature and industry practice. In *Retail supply chain management*, pages 99–153. Springer, 2008.

A. G. Kök, M. L. Fisher, and R. Vaidyanathan. Assortment planning: Review of literature and industry practice. In *Retail supply chain management*, pages 175–236. Springer, 2015.

R. D. McBride and F. S. Zufryden. An integer programming approach to the optimal product line selection problem. *Marketing Science*, 7(2):126–140, 1988.

I. Moon, K. S. Park, J. Hao, and D. Kim. Joint decisions on product line selection, purchasing, and pricing. *European Journal of Operational Research*, 262(1):207–216, 2017.

S. Mou, D. J. Robb, and N. DeHoratius. Retail store operations: Literature review and research directions. *European Journal of Operational Research*, 2017.

H. Shin, S. Park, E. Lee, and W. Benton. A classification of the literature on the planning of substitutable products. *European Journal of Operational Research*, 246(3):686–699, 2015.

I. L. Vasilyev and K. B. Klimentova. The branch and cut method for the facility location problem with clients preferences. *Journal of Applied and Industrial Mathematics*, 4(3):441–454, 2010.

E. Yücel, F. Karaesmen, F. S. Salman, and M. Türkay. Optimizing product assortment under customer-driven demand substitution. *European Journal of Operational Research*, 199(3):759–768, 2009.

F. S. Zufryden. A dynamic programming approach for product selection and supermarket shelf-space allocation. *Journal of the operational research society*, 37(4):413–422, 1986.