Product line optimization with multiples sites

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

[a]*Department of Industrial Engineering, Universidad de Chile, Chile*
[b]*Département d'Informatique, 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*
[e]*Universidad de Santiago de Chile (USACH), Industrial Engineering Department, Laboratory for the Development of Sustainable Production Systems (LDSPS), Chile*

Abstract

We consider the problem faced by a retail chain that must select what mutual-substitute items to display in each one of its stores to maximize revenues. The number of items cannot exceed the limit space capacity of each store. Customers purchase the one product that maximizes their utility, which depends on the product price, travel cost to the store, and reservation price, known to the retailer. The retailer can set different price markdowns at different stores and products. The retailer considers the decisions of customers, and solves a mixed-integer bilevel optimization problem, which can be formulated as a single-level optimization problem by using optimality conditions for the lower level. We propose Branch and Cut and Cut and Branch methods and include a family of valid inequalities to solve the problem. We compare the results with those of a Benders decomposition method. Our computational results show that the proposed Cut and Branch method obtains the best performance and improves the current state of the art.

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

1. Introduction

The product line optimization problem consists of finding the best product assortment to display in a store, in such a way so as to fit both the consumers' preferences and the limited available space in that store. Assortment optimization is especially relevant for products such as appliances (e.g., dishwasher machines, washers and dryers, refrigerators, TV sets, among others) because of the space they take in the store. This problem is usually solved for a single store. However, a retailer with a chain with several outlets in different locations in a city, can increase sales by jointly optimizing the assortments of all its stores. This is because some store of the chain that is not the closest to a customer, can have available a product that lacks in her closest stores due to the limited space in each one of them, and the customer could be willing to travel longer to obtain that product. Again, this is particularly valid for goods as appliances, as opposed to inexpensive fast-moving consumer goods, that are usually purchased at the closest store or at a facility that offers good prices for baskets of products.

We address the product line optimization of a retailer who sells appliances in several stores in different locations, and wants to optimize the product assortment across the whole chain, and in addition, to find the best markdowns , i.e., price reductions on the regular price for individual products in different stores. In this, we follow a JCPenney executive who states that "'assortments, allocations, markdown pricing are all linked and optimized together' at his company" [21].

The assortment selection problem has attracted much attention. In [23], 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 [24], 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 [35], a comparison is performed between different heuristics for the Product Line Design (PLD) problem. The authors introduce a ranking-based formulation, where each client's utility is introduced in the formulation as a

constraint. Another ranking-based formulation for the PLD problem is presented in [4]. These previous approaches ([4] and [35]) are compared in [6], which also introduces a strong formulation for the approach in [4]. A Benders decomposition (BD) approach, together with an efficient algorithm, is used to solve the problem, where the master problem defines the available products, and a separable subproblem problem solves each client's purchase decision. The efficiency of the BD is evaluated on synthetic data, obtaining good results.

The problem of selecting a subset of products and allocating them to a limited, integer number of shelves in a store, so that a function of costs and revenues is maximized is addressed in [45]. 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 [16], the authors use conjoint analysis to decide the product assortment, based on each product's 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 [44]. Their problem also considered exogenous demand, supplier selection, shelf space limitations and inventory management considerations. In [21], 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, i.e., their maximum willingness to pay. In [22], 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 [7] using a multinomial logit model for consumers' demand. The paper [36] 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 [27] 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 the allocation of products to shelves and deciding how much space to allocate to each product.

In the last years, the literature has included diverse extensions to the classic assortment problem, that intend to find a better representation of reality. For instance, [31] extends the concept of fully ranked-choice models to models with a partial ranking that additionally allow for indifference among subsets of products, that is, models in which the customer does not have a strict product preference. They prove that partially ranked-choice models are theoretically equivalent to fully ranked-choice models and proposed an embedded column generation procedure to efficiently estimate ranked-choice models partially from the historical transaction and assortment data. The subproblems involved can be efficiently solved by using a growing preference tree representing partially ranked preferences, enabling it to learn preferences and optimize assortments for thousands of products.

In addition, the classic version considers that the firm knows either the customers' willingness to pay or their preferences list (e.g., throughout a conjoint analysis approach), but this is not necessarily true. Instead of this, given the historical data, a firm may know the distribution probability with which purchasers buy a product. In this sense, recently, studies have focused on uncertainty in the choice of purchases. To this end, some proposed a data-driven optimization approach, being the multinomial logit the most used consumer behavior model [39]. For instance, in [14] incorporate latent class using a Nested logit, and they showed that the problem is polynomially solvable when the nest dissimilarity parameters of the choice model are less than one and the customers always purchase within the selected nest. Similarly, [34] presented a novel formulation of the d-level nested logit model as a tree of depth d, and provide an efficient algorithm to find the optimal assortment. For a d-level nested logit model with n products, the algorithm ran in 0(dn log n) time and an iterative algorithm is developed that generates a sequence of prices converging to a stationary point. In addition, [3] proposed a model-based approach that incorporates customers' preferences for compromise alternatives. They apply a utility model that integrates compromise variables into a multinomial logit model. They

formulate the resulting optimization problem as a mixed-integer linear program, in which the endogenous effects of selected products on other alternatives' utilities make it more complicated concerning models without compromising alternatives.

Nevertheless, in most of the above cases, expected profit is considered, being this not necessarily a good indicator because although there could be some certainty about the type of model structure, this could be not so about the parameters' values, or about the right model structure to use to describe the customer population. Given that, [5] proposed a new optimization approach for product line design under uncertainty. They optimize the worst-case expected revenue concerning an uncertainty set of models and propose different types of uncertainty sets that account for parametric and structural uncertainty. In addition, [15] study the robust assortment optimization under the Markov chain choice model. In this formulation, the model's parameters are assumed to be unknown, and the goal is to maximize the worst-case expected revenue over all parameter values in an uncertainty set. Under this class of models, which includes the Markov chain model, the choice probabilities are given as solutions to a system of linear equations. Under certain reasonable assumptions, they show a min-max duality result for the robust assortment optimization for this class of choice models.

The cases above consider the assortment decision in a single period, but the behavior of consumers may be different when considering multiple periods. In this sense, [38] considered a Markov chain choice model with a single transition. In this model, customers aim at each product with a certain probability. If the aimed product is unavailable, then the seller can recommend a subset of available products to the customer, and the customer will purchase one of the recommended products or can choose not to purchase with certain transition probabilities. They show that this problem is generally NP-Hard even if each product could only transit to at most two other products. They provided polynomial-time algorithms for several special cases, such as when the transition probabilities are homogeneous concerning the starting point or when each product can only transit to one other product. In addition, they propose a compact mixed-integer program formulation that can solve larger instances of this problem. Another work that considers multiperiod assortment is [19], which studied the strategy of a rotating products within a season and concealing part of the stock to consumers. By this action, the purchaser incorporates uncertainty to the relative valuation of consumer, and makes them purchase more products during the season. The profit obtained for this is called the value of concealment. They proved that if the customer is myopic this is a good strategy.

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 [13]. Other authors that address the same problem with different variants are [20, 27, 32, 28], who offer a review on the subject. A later review is presented by [33], 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 [40], on planning of product lines and [37] on retail store operations, including a good section on product assortment. There also exists commercial software, which solves the problem of product allocation to stores using simple rules of thumb [28].

In [12], 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. Stackelberg pricing with customers that purchase sets of products was the subject of [10]. That work analyzes the complexity and approximability of the resulting combinatorial problem.

Somewhat related is the competitive facility location problem [see 17]. However, the players of the 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 1]. Finally, [8], address the problem of product allocation among brick-and-mortar stores and online stores belonging to the same chain.

None of the above 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 different markdowns.

1.1. Our contribution

We extend for the first time the Product Line Design to the case with multiple stores, in which each customer or customers' segment can choose any store of the chain to make a purchase, as long as it is convenient for her. The problem, which we call the Product Line Optimization in Multiple Stores, PLOMS, includes an optimal price markdown

strategy. We present a bilevel formulation for this problem, which is collapsed to a single-level integer optimization formulation. Similarly to [41], the purpose of this paper it is to argue that this model of buyer behavior is better than others. Our main goal is to provide an algorithmic approach to solve this problem efficiently. With this end, we use and compare Branch and Cut ($B\&C$) and Cut and Branch ($C\&B$) methods. We adapt valid inequalities that have been used for the Facility Location problem with Preferences used in [11] and [43] to solve our formulation. Computational experiments show an improved performance with the $B\&C$ and $C\&B$ versus existing Benders decomposition methods for the single-store case.

1.2. Outline of the paper

The paper is organized as follows: Section 2 presents the problem, the bi-level 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 Product Line Optimization in Multiple Stores (PLOMS) considered in this work. We provide three different formulations for the PLOMS 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 π^l_{jk} be the unit price of product $k \in \mathcal{K}$ at store $j \in \mathcal{J}$, with markdown $l \in \mathcal{L}_{k,j}$, the set of possible markdowns for product k in store j. To simplify notation, we assume each pair (k, l) to be a new product k, and we simply use π_{jk}. The cost of assigning a product to a store is zero. There are capacity constraints, indicating that each store $j \in \mathcal{J}$ can display up to p_j products of the set $\mathcal{K}$.

As in [16], there is also a set $\mathcal{I}$ of consumer segments, to which for brevity, we refer to as "consumers". For modeling effects, we assume that the segments have equal size; however, if it is not so, it suffices including in the objective a weight for each segment, representing its size. The travel cost between consumer i and store j is κd_{ij}, with $\kappa = 1$ representing units of cost per unit of distance, and d_{ij} is the total round-trip distance traveled by the client to purchase at j. For actual values of travel cost, estimations can be computed as in [42]. Similarly to others works as [26] and without loss of generality, each consumer $i \in \mathcal{I}$ is interested only in those products in a set $\mathcal{K}_i \subseteq \mathcal{K}$ and has a reservation price r_{ik} for product k in $\mathcal{K}_i$. Both the set $\mathcal{K}_i$ and the reservation price are determined by market studies. For more details on how to determine these parameters, see the available marketing literature, e.g., [29, 30, 9], and references therein. Furthermore, a detailed study of the pricing models that use reservation price is presented in [41].

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 his or 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. The model can be used to maximize profit without any changes, except that in the objective, the weight on each client's purchase would be, in that case, unit profit for product k in store j.

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}	round-trip distance between client $i \in \mathcal{I}$ and store $j \in \mathcal{J}$
p_j	maximal number of products assigned to store $j \in \mathcal{J}$
π_{jk}	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 Product Line Optimization in Multiple Stores can be formulated as a linear-integer bilevel optimization problem with binary variables. We begin by introducing two sets of 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 $\tilde{x}^{(i)}_{jk}$, 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 ($\tilde{x}^{(i)}_{jk} = 1$) or not ($\tilde{x}^{(i)}_{jk} = 0$). With these variables we can formulate the PLOMS as the following bilevel optimization problem:

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

$$\text{s.t.} \sum_{k\in\mathcal{K}} y_{jk} \leq p_j \qquad j \in \mathcal{J}, \tag{1b}$$

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

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

$$x^{(i)}(y) = \arg\max_{\tilde{x}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}_i} (r_{ik} - \pi_{jk} - d_{ij})\tilde{x}^{(i)}_{jk}, \tag{1d}$$

$$\text{s.t.} \quad \tilde{x}^{(i)}_{jk} \leq y_{jk} \qquad j \in \mathcal{J}, k \in \mathcal{K}_i, \tag{1e}$$

$$\sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}_i} \tilde{x}^{(i)}_{jk} \leq 1, \tag{1f}$$

$$\tilde{x}^{(i)}_{jk} \in \{0,1\} \qquad j \in \mathcal{J}, k \in \mathcal{K}_i. \tag{1g}$$

Equations (1a) to (1g) display the formulation of the optimization model that must be solved by the retailer. The objective function (1a) maximizes the firm's profit obtained from customers' purchases. Equations (1b) upper-bound the number of products offered at each store. Constraints (1d) to (1g) are the optimization model that each customer $i \in \mathcal{I}$ must solve. The customer's objective (1d) maximizes her utility. Constraints (1e) forbid customer i buying product j in a store k that does not carry it, and (1f) represents the fact that the customer purchases only one product, at one store. Recall that we assume products that are expensive and have low turnover, e.g. a refrigerator, a TV, a washing machine, or any similar product. Finally, (1c) and (1g) are the domains of variables $\mathbf{y}$ and $\mathbf{x}$, respectively.

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 travel cost d_{ij}. It is interesting to remark that, as the same product could be included in a store with different price markdowns, this will never happen at the optimal solution, as the buyers would purchase only the least costly option and the costlier option would simply be using display space. Hence, the least price that gives the maximum revenue will always dominate. Therefore we can define the set of products/locations that are attractive to customer i as: $\mathcal{T}_i = \left\{(j,k) \mid r_{ik} - \pi_{jk} - d_{ij} \geq 0\right\}$. With this notation we can set $\tilde{x}^{(i)}_{jk} = 0$ if $(j,k) \notin \mathcal{T}_i$. We replace variable $\tilde{x}^{(i)}_{jk}$, with

variable $x_{jk}^{(i)}$ with $(j,k) \in T_i$, $i \in \mathcal{I}$ and we can express the summed terms in (1a), (1d) and (1f) as:

$$\sum_{i\in\mathcal{I}} \sum_{(j,k)\in\mathcal{T}_i} \pi_{jk} x_{jk}^{(i)},$$
$$\sum_{(j,k)\in\mathcal{T}_i} (r_{ik} - \pi_{jk} - d_{ij}) x_{jk}^{(i)},$$
$$\sum_{(j,k)\in\mathcal{T}_i} x_{jk}^{(i)} \leq 1 \ .$$

2.3. Single level model formulation

We now present a single level formulation for the PLOMS, following the approach in [25], which enforces the second level optimality conditions through constraints. This approach considers the following order relation for all $i \in \mathcal{I}$:

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

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 (1d)-(1g), is optimal for this problem if and only if it satisfies the following set of constraints [see 25]

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

This gives the following equivalent single-level integer programming formulation for the PLOMS:

$$(M1) \quad \max \quad \sum_{i\in\mathcal{I}} \sum_{(j,k)\in\mathcal{T}_i} \pi_{jk} x_{jk}^{(i)} \tag{2a}$$
$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} y_{jk} \leq p_j \qquad j \in \mathcal{J}, \tag{2b}$$
$$\sum_{(j',k')\in\mathcal{B}_{ijk}} x_{j'k'}^{(i)} \geq y_{jk} \qquad i \in \mathcal{I}, (j,k) \in \mathcal{T}_i, \tag{2c}$$
$$x_{jk}^{(i)} \leq y_{jk} \qquad i \in \mathcal{I}, (j,k) \in \mathcal{T}_i, \tag{2d}$$
$$\sum_{(j,k)\in\mathcal{T}_i} x_{jk}^{(i)} \leq 1 \qquad i \in \mathcal{I}, \tag{2e}$$
$$x_{jk}^{(i)} \in \{0,1\} \qquad i \in \mathcal{I}, (j,k) \in \mathcal{T}_i, \tag{2f}$$
$$y_{jk} \in \{0,1\} \qquad j \in \mathcal{J}, \ k \in \mathcal{K}. \tag{2g}$$

Note that the second-level problem (1d)-(1g) selects for each client i the product j,k with $y_{jk} = 1$ that has the largest positive profit $r_{ik} - \pi_{jk} - d_{ij}$. This is therefore equivalent to the linear programming problem where the binary variables $\tilde{x}_{jk}^{(i)}$ in (1g) are relaxed to $0 \leq \tilde{x}_{jk}^{(i)} \leq 1$. The optimality conditions (or KKT conditions) of this linear optimization problem can be used to characterize the optimal second-level solutions with constraints in a single-level problem. However, this becomes a – possibly challenging – mixed-integer nonlinear model (with quadratic constraints). How to solve this problem is left for future work. Here we investigate efficient solution methods for the integer programming problem (M1), where the second-level optimality is expressed with constraint (2c) constructed by exploiting the problem structure.

We now consider alternative constraints to (2c). For this define the set of products that are not preferable to $(j,k) \in \mathcal{T}_i$ by client $i \in \mathcal{I}$, that is $\mathcal{W}_{ijk} = \mathcal{T}_i \setminus \mathcal{B}_{ijk}$, (i.e., $\mathcal{W}_{ijk} = \{(j',k') \in \mathcal{T}_i | (j,k) \succ (j',k')\}$) hence, $\mathcal{W}_{ijk} \cap \mathcal{B}_{ijk} = \emptyset$ and $\mathcal{T}_i = \mathcal{W}_{ijk} \cup \mathcal{B}_{ijk}$. Then, for $i \in \mathcal{I}, (j,k) \in \mathcal{T}_i$, constraint (2c) is equivalent to

$$y_{jk} + \sum_{(j',k') \in \mathcal{W}_{ijk}} x^{(i)}_{j'k'} \leq 1 \qquad i \in \mathcal{I}, (j,k) \in \mathcal{T}_i \,. \tag{2c-1}$$

We define by M2 the optimization problem obtained by replacing (2c) in M1 by (2c-1). 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 PLOMS, which are constructed considering the interaction of more than than one client $i \in \mathcal{I}$, to help define tighter equivalent formulations for the problem.

We strengthen constraint (2c-1) by considering the set of product/locations that provide less utility than the combination (or "product") (j,k) for a second client i' but not for i. That is the set $\mathcal{W}_{i'jk} \cap \mathcal{B}_{ijk}$. Using this set, similar to [11], we obtain the following set of stronger inequalities:

$$\sum_{(j',k') \in \mathcal{W}_{ijk}} x^{(i)}_{j'k'} + \sum_{(j',k') \in \mathcal{W}_{i'jk} \cap \mathcal{B}_{ijk}} x^{(i')}_{j'k'} + y_{jk} \leq 1 \qquad i, i' \in \mathcal{I}, (j,k) \in \mathcal{T}_i \,. \tag{2c-2}$$

Note that this inequality is satisfied when $\sum_{(j',k') \in \mathcal{W}_{ijk}} x^{(i)}_{j'k'} = 1$, because in that case, $y_{\tilde{j}\tilde{k}} = 0$ for any $(\tilde{j},\tilde{k}) \in \mathcal{B}_{ijk}$, and in particular $\sum_{(j',k') \in \mathcal{W}_{i'jk} \cap \mathcal{B}_{ijk}} x^{(i')}_{j'k'} = 0$.

Constraint (2c-2) can be generalized to multiple clients by induction, as shown in [11]. Given $i_1, \ldots, i_s \in \mathcal{I}$, and $(j,k) \in \mathcal{T}_{i_1}$, then the following inequalities are valid for PLOMS:

$$\sum_{(j',k') \in \mathcal{W}_{i_1 jk}} x^{(i_1)}_{j'k'} + \sum_{t=2}^{s} \sum_{(j',k') \in \mathcal{B}_{i_1 jk} \cap \left(\bigcap_{q=2}^{t} \mathcal{W}_{i_q jk}\right)} x^{(i_t)}_{j'k'} + y_{jk} \leq 1 \qquad i_1 \in \mathcal{I}, (j,k) \in \mathcal{T}_i \,. \tag{2c-3}$$

Similarly to (2c-2), the inequality (2c-3) acts as follows: assume that $\sum_{(j',k') \in \mathcal{W}_{i_1 jk}} x^{(i_1)}_{j'k'} = 1$. Then, $y_{\tilde{j}\tilde{k}} = 0$ for any $(\tilde{j},\tilde{k}) \in \mathcal{B}_{i_1 jk}$, and in particular it must hold that $\sum_{(j',k') \in \mathcal{B}_{i_1 jk} \cap \left(\bigcap_{q=2}^{t} \mathcal{W}_{i_q jk}\right)} x^{(i_t)}_{j'k'} = 0$. Note that the second summation starts at $t = 2$ so that $x^{(i_1)}_{j'k'}$ is not repeated in both summations.

Below, we present valid inequalities that do not arise from strengthening constraint (2c). The next two sets of valid inequalities are stated in propositions that establish relationships between two customers' variables.

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*

$$x^{(i)}_{jk} \leq x^{(i')}_{jk} \quad \text{if} \;\; \mathcal{B}_{i'jk} \subseteq \mathcal{B}_{ijk} \tag{3}$$

Proof: Assume that $\mathcal{B}_{i'jk} \subseteq \mathcal{B}_{ijk}$. If $x^{(i)}_{jk} = 1$ then, with (2e), we have that $x^{(i)}_{j'k'} = 0$ 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 (2c) and that $\mathcal{B}_{ij'k'} \subseteq \mathcal{B}_{ijk} \setminus \{(j,k)\}$ for any such (j',k'). From $x^{(i)}_{jk} = 1$, (2d) and (2c) we get $1 = y_{jk} \leq \sum_{(j',k') \in \mathcal{B}_{i'jk}} x^{(i)}_{j'k'}$. This is a contradiction, since $(j',k') \in \mathcal{B}_{ijk} \setminus \{(j,k)\}$ by the hypothesis and, as concluded above, $y_{j'k'} = 0$. Therefore $x^{(i')}_{jk} = 1$ completing the proof. □

Proposition 1 generalizes a result in [11] to the case in which the sets of preferred products are different for each client. When $\mathcal{B}_{ijk} = \mathcal{B}_{i'jk}$, this result can be obtained as a corollary when repeating Proposition 1.

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

$$x^{(i)}_{jk} = x^{(i')}_{jk} \quad \text{if} \;\; \mathcal{B}_{ijk} = \mathcal{B}_{i'jk}$$

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*

$$\sum_{(j',k') \in \mathcal{W}_{ijk}} x^{(i)}_{j'k'} + \sum_{(j',k') \in \mathcal{B}_{ii'jk}} x^{(i')}_{j'k'} \leq 1 \,. \tag{4}$$

Proof: Note that if $\mathcal{B}_{ii'jk} = \emptyset$, (4) is true, since it is implied by (2e). Let us 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 $x^{(i)}_{j'k'} = 1$. This means, considering (2c-1) 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 (2d) imply that the second sum is zero. Consider now that the second sum is equal to one. Then, there exists $(j', k') \in \mathcal{B}_{ii'jk} \subseteq \mathcal{B}_{ijk}$ such that $x^{(i')}_{j'k'} = 1$. Because of (2d), $y_{j'k'} = 1$, and therefore $x_{i\hat{j}\hat{k}} = 0$ for all $(\hat{j}, \hat{k}) \in \mathcal{W}_{ij'k'}$ due to equation (2c-1). This implies that 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 PLOMS. We note that (3) has at most $\sum_{i\in\mathcal{I}} |\mathcal{T}_i|$ total constraints, while (4) 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 (3) and replaces (2c) in M1 with (2c-2). This formulation includes constraints that model interactions between pairs of consumers. We also define M4 as the problem that incorporates valid inequalities (3) and replaces (2c) in M1 with (2c-3), modeling interactions between sets of customers. These formulations are summarized in Table 3. Since constraints (2c-1), (2c-2), and (2c-3) are increasingly

Table 3: Summary of the different formulations

	(2a)	(2b)	(2c)	(2d)	(2e)	(2c-1)	(2c-2)	(2c-3)	(3)
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

stronger constraints, the corresponding formulations are tighter formulations of the PLOMS. Our approach to solve large instances of M3 and M4 will consider subsets of constraints (2c-2) and (2c-3), respectively. In addition, we consider the effect of including inequalities (4) in these formulations. Since there is a large number of these constraints, we add them using cutting plane approaches, similar to [43], as we see below.

3. Solution methods

The formulations M2, M3 and M4 of the PLOMS problem introduced in the previous table, represent mixed-integer problems that can be directly handled by a commercial solver. To efficiently solve large instances, we investigate different decomposition strategies for the PLOMS. This section presents a Benders decomposition strategy that is applicable to M2, problem preprocessing to remove constraints in M3 and M4, and a cut generation method for inequalities (4). We finalize with some variable and constraint simplifications for PLOMS.

3.1. Existing Benders decomposition method

In [6] a Benders decomposition approach is presented, using a formulation which is similar to M2, but applies when there is only one store. The master problem in the method prescribes which products are to be placed in the store, letting the customer's purchase decision as the second stage problem, which is separable in $|\mathcal{I}|$ independent sub-problems.

The solution method in [6] first assigns to each client the available product with the highest price, using for each customer an exact greedy algorithm to solve the dual problem both to the integer and fractional master solution, which generates the Benders 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.

In order to adapt this Benders decomposition method to the multiple-store case, we let the master problem solve the product availability problem in all stores and pass the solution to the master problem; the sub-problems depend on 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 [6]. In this adaptation of the Benders decomposition method, each sub-problem can generate a Benders optimality cut in every iteration.

The fact that formulations M3 and M4 include constraints that involve multiple clients, breaks the sub-problems separability and generates sub-problems that cannot be solved with the method suggested in [6]. Adapting this solution

method to these formulations in the multiple-store problem is not straightforward, and becomes a matter of future research. We therefore, only consider this Benders decomposition method in formulation M2, and also refer to it as M2-BD.

3.2. Cut generation methods

To reduce the problem size, here we introduce relaxations to formulations M3 and M4 by considering only a subset of constraints (2c-2) and (2c-3) that fully contain (2c-1) so that they are still valid reformulations of M1. We also present a cut generation approach that gradually incorporates constraints (4) to formulation M4. 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, (3), does not generate computational difficulties in our experiments, either because they are redundant constraints or because the condition $\mathcal{B}_{i'jk} \subseteq \mathcal{B}_{ijk}$ is difficult to satisfy.

3.2.1. A subset of constraints (2c-2)

Note that for a particular constraint (2c-2) to be different from the constraints in (2c-1), it is necessary that $\mathcal{W}_{ii'jk} \neq \emptyset$. Given the definition of $\mathcal{W}_{ii'jk}$, this set can be large when the attractive products for two customers overlap significantly. Therefore, the subsets of constraints (2c-2) 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:

$$\begin{aligned} \max \quad & \sum_{i,i' \in \mathcal{I} \mid i<i'} c_{i,i'} z_{i,i'} \\ \text{s.t.} \quad & \sum_{i' \in \mathcal{I} \mid i'>i} z_{i,i'} \leq 1 \quad i \in \mathcal{I}\,, \\ & z_{i,i'} \in \{0,1\} \qquad i, i' \in \mathcal{I}, i < i'\,. \end{aligned}$$

The optimal solution for this optimization problem indicates which pair of clients are to be used to build the subset of constraints (2c-2). We include one such constraint for each pair (i, i') such that $z_{ii'} = 1$. Note that the number of constraints generated are equal to the number of constraints in (2c-1), 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 (2c-3)

There is a set of constraints (2c-3) for every possible group of clients $\{i_1, \ldots, i_r\}$. To identify which groups of them to use to generate the subset of (2c-3), we used the following procedure, introduced in [11].

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 (2c-1) in step 3 by replacing them with constraints of the form (2c-3). Since at every iteration we are introducing tighter constraints, the resulting subset of (2c-3) implied constraints (2c-1).

3.2.3. Cut generation

For the cut generation strategy, we solve either problem M3 or M4 adding a subset of constraints (4). Once an LP relaxation solution to one of these problems is obtained, we check whether any of the remaining constraints (4) is violated. For each client, we generate at most one of the violated constraints and include it in the formulation. Then, we re-optimize and repeat this procedure until the optimal solution satisfies all constraints (4). A similar cut generation strategy for constraints (2c-2) and (2c-3) 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 (4) we begin by noticing 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, we 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 $(j,k)^{ii'}_{\max} = \arg\max_{(j,k)\in S\cap\mathcal{T}_{i'}}\{\sigma_i(j,k)\}$, where $S = \{(j,k)|x^{(i)}_{jk} > 0\ (j,k) \in \mathcal{T}_i\}$. We now show that for any $(j,k) \notin [(j,k)^i_{\min}, (j,k)^{ii'}_{\max}]$ variable x satisfies constraint (4).

By definition we have that $x^{(i)}_{j'k'} = 0$ for all $(j',k') \in \mathcal{W}_{i(j,k)^i_{\min}}$, which implies that $\sum_{(j'k')\in\mathcal{W}_{i(j,k)^i_{\min}}} x^{(i)}_{j'k'} = 0$ hence inequality (4) 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 (4) is satisfied.

The process of generating cuts by only verifying the range $[(j,k)^i_{\min}, (j,k)^{ii'}_{\max}]$ 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 were added to the problem as lazy constraints.

Algorithm 1 Cut generation

```
stop = True
for i ∈ I and stop = True do
    S = {(j,k)|x^(i)_jk > 0 (j,k) ∈ T_i}
    (j,k)^i_min = arg min_{(j,k)∈S} {σ_i(j,k)}
    (j,k)^ii_max = arg max_{(j,k)∈S∩T_i} {σ_i(j,k)}
    if y_{(j,k)^i_max} < 1 then
        for i' ∈ I \ {i} do
            (j,k)^ii'_max = arg max_{(j,k)∈S∩T_i'} {σ_i(j,k)}
            for (ĵ,k̂) ∈ T_i ∩ [(j,k)^i_min, (j,k)^ii'_max] do
                if Σ_{(j',k')∈W_iĵk̂} x^(i)_j'k' = y_{(jk)^i_max} then
                    break
                if Σ_{(j',k')∈W_ijk} x^(i)_j'k' + Σ_{(j',k')∈B_ii'ĵk̂} x^(i)_j'k' ≤ 1 then
                    add cut(i, i', ĵ, k̂)
                    stop = False
                    break
```

3.3. Problem preprocessing

To speed up the solution for these models, we remove or simplify the 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 (2c-1) becomes $y_{jk} \leq 1$ and it is 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 (2c-1) becomes $y_{jk} \leq x^{(i)}_{jk}$, but by (2d) $y_{jk} \geq x^{(i)}_{jk}$. Both constraints are removed and replaced by $y_{jk} = x^{(i)}_{jk}$.

Similarly, we conducted the following simplification for model M3

- If $|\mathcal{W}_{ii'jk}| = 0$ and $|\mathcal{W}_{ijk}| = 0$, then the pair (j,k) is the worst for client i. Constraint (2c-2) becomes $y_{jk} \leq 1$ and it is removed.
- If $|\mathcal{W}_{ii'jk}| = 0$ and $|\mathcal{W}_{ijk}| \neq 0$, then constraint (2c-2) become equal to constraint (2c-1) and it is removed.
- If $|\mathcal{W}_{ii'jk}| \neq 0$ and $|\mathcal{W}_{ijk}| = 0$, then constraint (2c-2) becomes equal to constraint (2c-1) and it is 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 (2c-2) becomes $y_{jk} \leq x^{(i)}_{jk}$, but by (2d) $y_{jk} \geq x^{(i)}_{jk}$, so both are removed and replaced by $y_{jk} = x^{(i)}_{jk}$.

4. Computational experiments

We now present the computational tests. All the procedures and algorithms have been written using Python and, for MIP problems, we used IBM ILOG CPLEX version 12.9. The experiments were performed in an Intel Xeon 32 multicore processors, 2.00 GHz speed each and 128 GB of RAM, running the CentOS release 6.7 Linux operating system. We begin by describing the synthetic data that was used in our experiments. Our computational results explore the strength of the different formulations and the existing Benders decomposition strategies (Subsection 4.2), the effectiveness of the constraints (4) 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 [2]. Values d_{ij}, r_k and r^k_i were built, 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 travel cost between each client i and each store j are trivial to obtain. The remaining parameters were determined as follows: $\pi_k \in [1.5\overline{d}, 5\overline{d}]$ and $r^k_i \in [2\underline{d_i} + \overline{\pi}, 2\overline{d_i} + \overline{\pi}]$, where $\overline{d}$ is the average of all travel costs, $\overline{\pi}$ is the average of all products prices; $\underline{d_i}$ and $\overline{d_i}$ the travel cost between the client i and the closest and farthest store, respectively. Finally, we fixed a capacity p_j, which is the number of products to display, arbitrarily for each store. Without loss of generality, we fixed all stores with equal capacity, i.e $p_j = p \; \forall j \in \mathcal{J}$.

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 $p_j = p = 5, 10 \; \forall j \in \mathcal{J}$. We use small values of capacities to be able to evaluate better the performance of the methods, as preliminary experiments showed that large capacities decrease the computational time. Each instance so defined, was run using five different seeds, that determined five different combination of values of the random parameters and hence, five 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.

In the results below we denote each instance 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 last upper (lower) bound obtained in the branch-and-bound process when the time limit is reached or the optimal solution found, whatever happens first..
- GAP_{LP} (%) : average integrality gap $(\frac{LP-LB}{LB}) \times 100\%$, with LP = linear relaxation value
- $TIME$: average CPU processing time in seconds (total time)
- $TIME_{LP}$: average CPU processing time in seconds of the linear relaxation
- Ni : number of instances solved to full optimality

Table 4: Statistics of generated instances

		4 stores							8 stores							12 stores						
		clients x items			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

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

In Table 5 we compare the results obtained by different existing solution methods on the M2 formulation. 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 [6]. 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 5: Computational results of model M2

	M2-CPLEX-BD			M2-BD			M2-default CPLEX		
$\|\mathcal{I}\|$-$\|\mathcal{J}\|$-$\|\mathcal{K}\|$	*GAP*	Ni	*TIME*	*GAP*	Ni	*TIME*	*GAP*	*Ni*	*TIME*
100-4-20	0.00	5	25	0.00	5	**18**	0.00	5	36
100-4-30	0.00	5	**62**	0.00	5	67	0.00	5	81
100-4-40	0.00	5	**244**	0.00	5	671	0.00	5	301
100-4-50	0.00	5	423	0.00	5	441	0.00	5	**131**
100-4-80	0.00	5	752	0.00	5	429	0.00	5	**428**
160-4-20	0.00	5	33	0.00	5	**32**	0.00	5	67
160-4-30	0.00	5	**278**	0.00	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**	0.00	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**	0.00	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

Table 6 compares the best alternative of M2 with 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 6: Comparison of GAP. GAP_{LP} and $TIME$ between the formulations M2. M3 and M4

	M2			M3			M4		
$\|\mathcal{I}\|$-$\|\mathcal{J}\|$-$\|\mathcal{K}\|$	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	1,220	0.00	2.71	675	0.00	**2.47**	**611**
160-4-50	0.29	2.48	2,124	0.09	2.23	1,513	**0.00**	**2.06**	**1,049**
160-4-80	0.67	1.94	3,094	0.25	1.76	2,961	**0.12**	**1.60**	**2,455**
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	1,389	0.00	3.62	1,365	0.00	**3.24**	**1,073**
250-4-50	2.06	5.11	3,600	1.78	4.27	3,600	**0.78**	**3.75**	**3,050**
250-4-80	2.78	4.20	3,600	2.50	**3.59**	3,600	**2.23**	3.62	3,600
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	1,882	**0.00**	3.65	2,013	**0.00**	**3.40**	**2,306**
400-4-40	2.25	5.77	3,600	1.83	4.28	3,600	**1.11**	**3.61**	3,600
400-4-50	3.13	5.16	3,600	2.72	4.23	3,600	**1.74**	**3.57**	3,600
400-4-80	3.75	4.72	3,600	3.22	3.87	3,600	**2.26**	**3.02**	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 (4)

To assess the effect of constraints (4), we added all of them to problem M3 for small instances and compared 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 (2c-2), 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 (4). Adding these constraints clearly reduces GAP_{LP} and, although $TIME_{LP}$ increases, the total solution $TIME$ is reduced in the largest instances. We remark that we did not use model M4 in this comparison, due to the high computer cost for generating all the constraints (2c-3).

Table 7: Comparison of GAP_{LP}, TIME and $TIME_{LP}$ between formulations M3 and M3 plus (4). Store capacity p = 5

	M3*			M3* + (4)		
$\|\mathcal{I}\|$-$\|\mathcal{J}\|$-$\|\mathcal{K}\|$	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 Tables 8 and 9 we explore the efficiency of valid inequalities (4) on large instances. In this case, we considered model M4 incorporating inequalities (4) using $C\&B$ (with CPLEX default cuts turned off) and compared its gap, the number of solved instances, solution time with that solving model M4 and model M4^o (with CPLEX default cuts off), the number of instances in which the optimal solution was found and the number of nodes. 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 (4) at the root node, dominates both M4 and M4 in every aspect^o.

Table 8: Comparison of valid inequality (4) with default CPLEX cuts and *C&B*. Store capacity $p = 5$

	$\|\mathcal{J}\| = 4$						$\|\mathcal{J}\| = 8$						$\|\mathcal{J}\| = 12$					
	C&B		M4^o		M4		*C&B*		M4^o		M4		*C&B*		M4^o		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

Table 9: Comparison average number of nodes and number optimal solutions find for valid inequality (4) with default CPLEX cuts and *C&B*. Store capacity $p = 5$

	$\|\mathcal{J}\| = 4$						$\|\mathcal{J}\| = 8$						$\|\mathcal{J}\| = 12$					
	C&B		M4^o		M4		*C&B*		M4^o		M4		*C&B*		M4^o		M4	
$\|\mathcal{I}\|$-$\|\mathcal{K}\|$	*SOL*	*NODE*	*SOL*	*NODE*	*SOL*	*NODE*	*SOL*	*NODE*	*SOL*	*NODE*	*SOL*	*NODE*	*SOL*	*NODE*	*SOL*	*NODE*	*SOL*	*NODE*
100-20	5	**5**	5	3669	5	1661	**5**	**2**	4	158331	5	30941	**5**	**13**	4	119326	4	24029
100-30	5	**1**	5	3170	5	1356	**5**	**76**	4	87937	5	27348	**5**	**2**	2	92235	1	38340
100-40	5	**14**	5	9623	5	2837	**5**	**7**	3	45818	3	25501	**5**	**45**	2	34246	2	17346
100-50	5	**9**	5	2393	5	897	**5**	**6**	4	32502	4	12230	**5**	**1**	3	48469	3	22129
100-80	5	**21**	5	4048	5	2170	**5**	**7**	2	42063	2	22131	**5**	**1**	0	34958	0	23770
160-20	5	**7**	5	2297	5	1508	**5**	**1**	3	108741	2	22474	**5**	**1**	2	63365	1	11367
160-30	5	**40**	5	6687	5	1416	**5**	**31**	1	66044	1	14911	**5**	**33**	0	33766	0	7602
160-40	5	**362**	5	6738	5	3184	**5**	**24**	0	28663	0	8510	**5**	**67**	0	13705	0	6110
160-50	5	**31**	5	15597	5	4184	**5**	**7**	1	16292	1	6678	**5**	**42**	0	8312	0	2380
160-80	5	**217**	3	12652	3	5493	**5**	**173**	0	13655	0	3426	**5**	**105**	0	2824	0	1118
250-20	5	**12**	5	1835	5	1610	**5**	**1**	4	8158	4	5420	**5**	**20**	0	12200	0	2733
250-30	5	**4**	5	4821	5	2554	**5**	**262**	1	10688	0	2719	**5**	**155**	0	7405	0	2047
250-40	5	**4**	5	7469	5	1922	**5**	**210**	0	6134	0	1778	**5**	**44**	0	3237	0	959
250-50	**5**	**419**	3	12787	3	2906	**5**	**26**	0	3264	0	919	**5**	**59**	0	990	0	748
250-80	**5**	**256**	0	3268	0	1192	**4**	**117**	0	679	0	668	**3**	**54**	0	738	0	270
400-20	5	**1**	5	2058	5	1349	**5**	**1**	1	2398	1	961	**5**	**1**	0	1732	0	817
400-30	5	**17**	5	7040	5	2896	**5**	**313**	0	1874	0	728	**4**	**352**	0	631	0	799
400-40	**5**	**46**	0	8608	0	1571	**5**	**13**	0	1185	0	845	**4**	**44**	0	1016	0	736
400-50	**5**	**33**	0	3279	0	854	**3**	**120**	0	935	0	635	**3**	**7**	0	575	0	640
400-80	**2**	**208**	0	704	0	537	0	**12**	0	321	0	189	0	**0**	0	310	0	1

We finish evaluating the impact of constraint (4), solving the single store instances in [6] with B&C and C&B and comparing it to the Benders decomposition method proposed in that paper. As Table 10 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 [6] 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

Table 10: Comparison of run TIME of instances [6]. Stores number $|\mathcal{J}| = 1$. $p = \infty$ means unconstrained p.

	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

11 and Table 12 show the average GAP and $TIME$ over five instances for each $|\mathcal{I}|$-$|\mathcal{J}|$-$|\mathcal{K}|$ combination. In the three methods, we deactivated any cuts generated by default in CPLEX, and used as cut the valid inequality (4). The best results were obtained by $C\&B$ in terms of GAP and $TIME$. In fact, this approach obtained the optimal solution for most instances.

Table 11: 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*
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 11 and Table 12 show, as the store capacity increases, the time required to solve the problem tends to decrease. The problem was 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.

5. Managerial insights and Conclusions

Available results in the literature linked to product line optimization, also called assortment planning problem, optimize assortment considering that the firm has only one store, so the geographical dimension is absent. None of them consider the travel cost of the clients. This assumes that, if a firm has multiple stores distributed geographically, either all stores have the same assortment, or the assortment at each store is optimized separately considering only the local demand. The models exposed in this paper allow customers to choose different stores to make a purchase,

Table 12: 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~~3	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
400-20	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
400-30	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
400-40	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
400-50	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
400-80	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

as long as they are within reasonable distances. This is especially common in expensive purchases. The customers can decide to travel farther away to purchase at a lower price or perhaps to find a product that is unavailable at the closest store. Using a multiple store (PLOMS) model, the firm decides the assortments and prices in all its stores simultaneously and takes advantage of the mobility of the customers, obtaining an increased profit.

We illustrate the advantages of the PLOMS with a simple example. The example considers one product, three stores m_j with a single capacity, and six customers c_i which have known reservation prices r_i. Table 13 shows the input parameters of the example: travel costs for customer c_i traveling to store m_j, and reservation price of each customer. The customers' utility includes the reservation price, the product cost and the travel cost. In this instance there are three possible prices: 110, 100 and 90. Table 14 shows the results of three policies: (i) same assortment and same prices in all stores, (ii) a separate assortment problem being solved for each store, in which assortment and prices are set considering only the purchase decisions of the set of customers for which the store is the closest and (iii) the PLOMS model. Shown in the Table are the optimal prices, the stores at which customers purchase the product, and the utility obtained by each customer for the purchase, for the three policies. The Table also shows the firm earnings (firm e.) and the customers' utility (cust. u.) for each policy, in the two last rows, and the closest stores for each customer, in the last column. The company earnings are just the sum of the prices paid by the customers, and the customers' total utility is the sum of all the individual utilities.

First of all, the same assortment and the separate stores policies are restricted cases of PLOMS, as the first is obtained by constraining all stores to have the same assortment and the second one is obtained by restricting the store choice of the customers to the closest store. Thus, PLOMS dominates over the remaining policies. For our toy example, the Table shows that PLOMS results in the highest firm earnings, followed by separate stores and next by same assortment policies. Note that the sum of the firm earnings and the total customers' utility is very close for the three policies, which could suggest a zero-sum game. However, this is not so, as the utility of the customers includes factors that are not in the firm's earnings.

Even a very small instance as this, shows how there are cases, uncovered by PLOMS, in which a customer may choose to patronize a store that is not the closest. It is the case of customer 2, who goes to store 1, when the firm sets a higher price in store 2. The firm, on the other side of the street, has the knowledge of the reservation prices and can increase the price in store 2, because it will not loose customer 2, but will push her to make a longer trip to get the product.

This very small example not only shows the advantage of the PLOMS approach, but it also shows how the customers' decisions change when faced with multiple store options, generating a different allocation structure.

The table shows a sensitivity analysis of the showcase capacity. We study the sensibility analysis for the case with 100 customers, 4 malls, and 80 products fixed in a scenario described in 4.1. The summery of the results are shown below in Table 15. The column *clients x stores* shows statistics of the total, average, and standard deviation of the number of customers that each store satisfies. The column *closests clients* shows the statistics of the total, average,

Table 13: Reservation price r and travel cost d of the buyers.

	d			
	m1	m2	m3	r
c1	24	**2**	31	113
c2	10	**9**	23	118
c3	**7**	16	30	115
c4	8	22	**6**	107
c5	15	17	**7**	116
c6	13	24	**5**	104

Table 14: Buyers utility for each price ($r - p - d$). Unique price strategy (i) are numbers enclosed in squares. Separate problem strategy (ii) are the numbers enclosed in circles. The PLOMS strategy (iii) are the red numbers.

	same assortment			separate stores			PLOMS			
	price	store	cust. utility	price	store	cust. utility	price	store	cust. utility	closest store
c1	90	m2	21	100	m2	11	110	m2	1	m2
c2	90	m2	19	100	m2	9	100	m1	8	m2
c3	90	m1	18	100	m1	8	100	m1	8	m1
c4	90	m3	11	90	m3	11	90	m3	11	m3
c5	90	m3	19	90	m3	19	90	m3	19	m3
c6	90	m3	9	90	m3	9	90	m3	9	m3
firm e.	540			570			580			
cust. u.	97			67			56			

and standard deviation of the clients that are bought in the store closest to them. The column *items x stores* shows the statistics of the average total and deviation of the number of products in each store. In this case, the total reflects the number of different products displayed in the total number of stores. Finally, the utility of the signature and the computational time of each instance are shown.

Given that the firm's assortment decision is operational, we did not perform a sensitivity analysis on the expansion of stores, which is a strategic decision, nor on the number of products offered since this could depend on other actors. Table 15 shows that the inclusion of more capacity increases the profit; however, it is capped. In particular, with an equal capacity of 8 for all stores, the profit is equal to adding a new unity of capacity. In the extreme case, if the decision-maker offers all the products available on display, customers could choose the most attractive product, which is not necessarily the company's highest profit. Following the above, we can see that it is necessary to only 18 products to maximize the profit in this case. This means that it only 18 products, it is possible to capture the maximizing benefits. Note that this output may be used in the marketing area to develop a strategy targeted for this specif subset of products. In addition, the commercial area may decide if realize a markdown for the rest of the products, in order to a greater opportunity to be sold.

Similarly, it is observed that by increasing the capacity, the number of customers who buy in their nearest store increases from 20 to 75 customers. This is because there is a greater chance of finding an attractive product in a nearby store. Also, the column *items x stores* shows that there exist stores more attractive than others, that concentrate more products accumulating until 14 items in a store, that coincide with that there exist stores that satisfied until 37 *closest clients* and until 46 clients in general. It is also observed that increasing the capacity makes it possible to capture customers who previously could not buy given the available products. Because we assume that customers are rational, so they do not buy a pair (product, store) that gives them a negative utility.

From a computational point of view, the times are reduced as the demand increases, being the problem with a higher computational cost when the capacity is equal to one. As mentioned in the previous example, the problem with a capacity equal to one is equivalent to a problem of price allocation in each store for a product.

From an optimization point of view, it is observed that there is a degenerate solution, it is mean that a different

configuration of assortment exists to obtain the exact optimal solution. From the point of view, it is allowed to the decision-maker a configuration subject to other parameters like a min/max stores in each store, number of customers (or segment customer) that satisfy each store, among others.

Table 15: Sensibility analysis. Considering $|\mathbb{I}| = 100$, $|\mathbb{J}| = 4$, $|\mathbb{K}| = 80$.

	clients x stores			closest clients			items x stores				
Capacity	total	max	min	total	max	min	total	max	min	profit	time
1	74	23	14	20	11	2	4	1	1	17955	61
2	95	27	21	24	9	4	8	2	2	23246	69
3	99	32	19	27	15	2	12	3	3	24460	7
4	100	34	13	41	21	4	15	4	4	24611	8
5	100	37	13	48	23	5	16	5	5	24680	7
6	100	39	14	51	24	7	17	6	6	24696	8
7	100	40	14	62	29	8	18	7	7	24708	3
8	100	36	17	69	30	11	18	8	8	24709	3
9	100	39	19	75	35	12	18	9	8	24709	3
10	100	46	16	74	37	10	18	10	7	24709	3
11	100	31	19	67	26	10	18	11	7	24709	4
12	100	36	17	71	31	10	18	12	8	24709	3
13	100	35	17	75	31	10	18	12	8	24709	3
14	100	31	21	66	26	12	18	11	7	24709	3
15	100	45	14	69	35	9	18	12	7	24709	3
16	100	37	19	76	33	11	18	11	8	24709	3
17	100	30	20	65	26	12	18	10	7	24709	3
18	100	31	16	69	26	11	18	13	7	24709	2
19	100	33	17	75	30	12	18	14	8	24709	2
20	100	33	15	67	26	9	18	12	6	24709	3

A bi-level formulation is proposed to cover the interaction between the customers' choice and the brands.

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 PLOMS 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 [6]. 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 [6]. For one store, using synthetic data in [6], the Benders decomposition algorithm solved the instances with 500 and 1000 clients, faster.

There are several pathways to expand the research in this paper. For products in the category of appliances, it is reasonable to consider that customers purchase at most one unit. For other categories of products, however, e.g., clothing, consumers can buy a bundle of products [18], which is an extension worth exploring. For this extension, there are several possibilities: the customer can purchase either all the products in the bundle in one store or she can visit several stores, purchasing part of the bundle in each of them. The travel cost will be included in the customer's decision. Adding temporal decisions of both of the consumer and the firm, to include inventory, product obsolescence, and markdown pricing, can be considered in future research.

6. Acknowledgements

The authors gratefully acknowledge the support by Grants FONDECYT 1190064, CONICYT PIA AFB180003 and INRIA Associated Team BIO-SEL. 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. Powered@NLHPC: This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02) of the Universidad de Chile.

References

[1] Aras, N. and Küçükaydın, H. (2017). Bilevel models on the competitive facility location problem. In Spatial Interaction Models, pages 1–19. Springer.
[2] Beasley, J. E. (1990). Or-library: distributing test problems by electronic mail. Journal of the operational research society, 41(11):1069–1072.
[3] Bechler, G., Steinhardt, C., Mackert, J., and Klein, R. (2021). Product line optimization in the presence of preferences for compromise alternatives. European Journal of Operational Research, 288(3):902–917.
[4] Belloni, A., Freund, R., Selove, M., and Simester, D. (2008). Optimizing product line designs: Efficient methods and comparisons. Management Science, 54(9):1544–1552.
[5] Bertsimas, D. and Mišić, V. V. (2017). Robust product line design. Operations Research, 65(1):19–37.
[6] Bertsimas, D. and Mišić, V. V. (2019). Exact first-choice product line optimization. Operations Research, 67(3):651–670.
[7] Besbes, O. and Sauré, D. (2016). Product assortment and price competition under multinomial logit demand. Production and Operations Management, 25(1):114–127.
[8] Bhatnagar, A. and Syam, S. S. (2014). Allocating a hybrid retailer's assortment across retail stores: Bricks-and-mortar vs online. Journal of Business Research, 67(6):1293–1302.
[9] Breidert, C. (2007). Estimation of willingness-to-pay: Theory, measurement, application. Springer Science & Business Media.
[10] Briest, P. and Krysta, P. (2011). Buying cheap is expensive: Approximability of combinatorial pricing problems. SIAM Journal on Computing, 40(6):1554–1586.
[11] Cánovas, L., García, S., Labbé, M., and Marín, A. (2007). A strengthened formulation for the simple plant location problem with order. Operations Research Letters, 35(2):141–150.
[12] Chen, M., Chen, Z.-L., Pundoor, G., Acharya, S., and Yi, J. (2015). Markdown optimization at multiple stores. IIE Transactions, 47(1):84–108.
[13] Chen, M.-C. and Lin, C.-P. (2007). A data mining approach to product assortment and shelf space allocation. Expert Systems with Applications, 32(4):976–986.
[14] Davis, J. M., Gallego, G., and Topaloglu, H. (2014). Assortment optimization under variants of the nested logit model. Operations Research, 62(2):250–273.
[15] Désir, A., Goyal, V., Jiang, B., Xie, T., and Zhang, J. (2020). Robust assortment optimization under the markov chain model. Management Science, 66(2):698–721.
[16] Dobson, G. and Kalish, S. (1993). Heuristics for pricing and positioning a product-line using conjoint and cost data. Management Science, 39(2):160–175.
[17] Eiselt, H. A., Marianov, V., and Drezner, T. (2015). Competitive location models. In Location science, pages 365–398. Springer.
[18] Fang, Y., Sun, L., and Gao, Y. (2017). Bundle-pricing decision model for multiple products. Procedia computer science, 112:2147–2154.
[19] Ferreira, K. J. and Goh, J. (2021). Assortment rotation and the value of concealment. Management Science, 67(3):1489–1507.
[20] Flamand, T., Ghoniem, A., Haouari, M., and Maddah, B. (2017). Integrated assortment planning and store-wide shelf space allocation: An optimization-based approach. Omega, 81:134–149.
[21] Ghoniem, A. and Maddah, B. (2015). Integrated retail decisions with multiple selling periods and customer segments: optimization and insights. Omega, 55:38–52.
[22] Ghoniem, A., Maddah, B., and Ibrahim, A. (2016). Optimizing assortment and pricing of multiple retail categories with cross-selling. Journal of Global Optimization, 66(2):291–309.
[23] Green, P. E. and Krieger, A. M. (1985). Models and heuristics for product line selection. Marketing Science, 4(1):1–19.
[24] Green, P. E. and Krieger, A. M. (1989). Recent contributions to optimal product positioning and buyer segmentation. European Journal of Operational Research, 41(2):127–141.
[25] Hansen, P., Kochetov, Y., and Mladenovi, N. (2004). Lower bounds for the uncapacitated facility location problem with user preferences. Cahiers du Gerad, G-2004-24.
[26] Heilporn, G., Labbé, M., Marcotte, P., and Savard, G. (2010). A parallel between two classes of pricing problems in transportation and marketing. Journal of Revenue and Pricing Management, 9(1):110–125.
[27] Hübner, A. and Schaal, K. (2017). An integrated assortment and shelf-space optimization model with demand substitution and space-elasticity effects. European Journal of Operational Research, 261(1):302–316.
[28] Hübner, A. H. and Kuhn, H. (2012). 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.
[29] Jagpal, S. et al. (2008). Fusion for profit: how marketing and finance can work together to create value. Oxford University Press.
[30] Jedidi, K. and Jagpal, S. (2009). Willingness to pay: measurement and managerial implications. Handbook of pricing research in marketing, pages 37–60.
[31] Jena, S. D., Lodi, A., Palmer, H., and Sole, C. (2020). A partially ranked choice model for large-scale data-driven assortment optimization. Informs Journal on Optimization, 2(4):297–319.
[32] Kök, A. G., Fisher, M. L., and Vaidyanathan, R. (2008). Assortment planning: Review of literature and industry practice. In Retail supply chain management, pages 99–153. Springer.
[33] Kök, A. G., Fisher, M. L., and Vaidyanathan, R. (2015). Assortment planning: Review of literature and industry practice. In Retail supply chain management, pages 175–236. Springer.

[34] Li, G., Rusmevichientong, P., and Topaloglu, H. (2015). The d-level nested logit model: Assortment and price optimization problems. Operations Research, 63(2):325–342.
[35] McBride, R. D. and Zufryden, F. S. (1988). An integer programming approach to the optimal product line selection problem. Marketing Science, 7(2):126–140.
[36] Moon, I., Park, K. S., Hao, J., and Kim, D. (2017). Joint decisions on product line selection, purchasing, and pricing. European Journal of Operational Research, 262(1):207–216.
[37] Mou, S., Robb, D. J., and DeHoratius, N. (2018). Retail store operations: Literature review and research directions. European Journal of Operational Research, 265(2):399–422.
[38] Nip, K., Wang, Z., and Wang, Z. (2021). Assortment optimization under a single transition model. Production and Operations Management, 30(7):2122–2142.
[39] Qi, M., Mak, H.-Y., and Shen, Z.-J. M. (2020). Data-driven research in retail operations—a review. Naval Research Logistics (NRL), 67(8):595–616.
[40] Shin, H., Park, S., Lee, E., and Benton, W. (2015). A classification of the literature on the planning of substitutable products. European Journal of Operational Research, 246(3):686–699.
[41] Shioda, R., Tunçel, L., and Myklebust, T. G. (2011). Maximum utility product pricing models and algorithms based on reservation price. Computational Optimization and Applications, 48(2):157–198.
[42] Small, K. A. (2012). Valuation of travel time. Economics of transportation, 1(1-2):2–14.
[43] Vasilyev, I. L. and Klimentova, K. B. (2010). The branch and cut method for the facility location problem with client's preferences. Journal of Applied and Industrial Mathematics, 4(3):441–454.
[44] Yücel, E., Karaesmen, F., Salman, F. S., and Türkay, M. (2009). Optimizing product assortment under customer-driven demand substitution. European Journal of Operational Research, 199(3):759–768.
[45] Zufryden, F. S. (1986). A dynamic programming approach for product selection and supermarket shelf-space allocation. Journal of the operational research society, 37(4):413–422.