Omega 81 (2018) 134–149

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Integrated assortment planning and store-wide shelf space allocation: An optimization-based approach[☆]

Tulay Flamand[a,*], Ahmed Ghoniem[b], Mohamed Haouari[c], Bacel Maddah[d]

[a] Division of Economics & Business, Colorado School of Mines, Golden, CO 80401, USA
[b] Department of Operations & Information Management, Isenberg School of Management, University of Massachusetts, Amherst, MA 01003, USA
[c] Department of Mechanical and Industrial Engineering, Qatar University, Qatar
[d] Department of Industrial Engineering and Management, American University of Beirut, Beirut, Lebanon

ARTICLE INFO

Article history:
Received 4 March 2016
Accepted 19 October 2017
Available online 21 October 2017

Keywords:
Assortment selection
Shelf space allocation
Optimization-based heuristic
Retail prescriptive analytics

ABSTRACT

This paper investigates retail assortment planning along with store-wide shelf space allocation in a manner that maximizes the overall store profit. Each shelf comprises a set of contiguous segments whose attractiveness depends on the store layout. The expected profit accruing from allocating space to a product category depends not only on shelf segment attractiveness, but also on the profitability of product categories, their expected demand volumes, and their impulse purchase potential. Moreover, assortment affinities and allocation affinity/disaffinity considerations are enforced amongst certain pairs of interdependent product categories. A mixed-integer programming model is developed as a standalone approach to the problem and is also embedded in an optimization-based heuristic. The latter employs an initial feasible solution that is iteratively refined by re-optimizing subsets of shelves that are selected using a probabilistic scheme. A motivational case study in the context of grocery stores demonstrates the usefulness of the methodology and insights into the structure of optimal solutions are discussed. We show that the model selects a composite assortment of fast-movers and high-impulse product categories and constructs an effective retail shelf space allocation that promotes shopping convenience and unplanned purchases. Further, our computational study examines a testbed of 50 instances involving up to 800 product categories and 100 shelves for which our heuristic consistently yields solutions within 0.5% optimal in manageable times and drastically outperforms CPLEX with a time limit.

1. Introduction and motivation

Retailers continually face the challenge of utilizing scarce shelf space in a fashion that maximizes customer footprint (the total number of customers visiting the store), conversion from visiting to purchasing, and, eventually, profit [38,42]. While attracting customers to a store is strongly dependent on the breadth of the offered assortment (e.g. [6] and references therein), conversion depends mainly on in-store activities in relation with space allocation and store layout (e.g. [38]). Therefore, shelf space allocation and assortment planning are two important and intertwined facets of retailing decision making that allow improving both footprint and conversion. In this paper, we investigate these two critical decisions with a focus on super market applications.

In the context of grocery stores, the assortment must necessarily include product categories related to fast-moving consumer goods, i.e., “basic”, high-sales product categories that are usually at low cost/profit margin (e.g., bread, eggs, milk). The rest of the store assortment commonly comprises impulsive categories which may be predominantly purchased without prior planning. Impulsive purchasing is typical, for example, for *hedonic* products, which are associated with “procuring pleasure” such as candy and cake [28]. However, due to the scarcity of shelf space, especially in smaller legacy stores or stores located in densely populated urban zones, certain categories may not be sufficiently promising for inclusion in the store. As the retailer chooses a mix of product categories, including low-profit fast-movers and high-profit impulse product categories, (s)he must jointly optimize store-wide shelf space allocation. On the one hand, allocating fast-movers to convenient areas of the store can ensure a positive shopping experience for planned purchases and enhance customer loyalty. On the other hand, reserving (some of the) highly visible shelf areas for high-impulse product categories can stimulate unplanned purchases and

☆ This manuscript was processed by Associate Editor Kuhn.

* Corresponding author.
E-mail addresses: tflamand@mines.edu (T. Flamand), aghoniem@isenberg.umass.edu (A. Ghoniem), mohamed.haouari@qu.edu.qa (M. Haouari), bacel.maddah@aub.edu.lb (B. Maddah).

https://doi.org/10.1016/j.omega.2017.10.006
0305-0483/

generate an additional stream of revenue. To identify good trade-offs for this challenging problem, we investigate integrated assortment planning and shelf space management problems.

Given the tactical nature of the present work, we consider the store layout as a flat store from the bird's-eye view. We address assortment and shelf-space allocation considering the selection/allocation of "product categories." A retail product category includes several substitutable brands/individual SKUs that serve the same need for the customer (e.g., milk is viewed as one product category). Throughout the paper, the terms "product category" and "product" are used interchangeably. Each product category is characterized by its relative profitability and lower/upper bounds on its space allocation. Such bounds can be informed by historical practices, benchmarking against competitors, or marketing strategies that seek to create a particular store image or to give more visibility to new products or products on promotion. We consider four types of product category interdependence. The first relationship, henceforth referred to as *symmetric assortment affinity*, stipulates that two product categories should be simultaneously selected or not retained in the assortment (e.g., American coffee and coffee filters) due to cross-selling. The second relationship, referred to as *asymmetric assortment affinity*, requires that if a product category is selected, its dependent product category should be also selected in the assortment, and both product categories should be assigned to the same shelf. For instance, cake icing would not be selected in the assortment unless cakes are offered. The third relationship, referred to as *allocation disaffinity*, precludes the assignment of certain pairs of product categories to the same shelf (e.g., detergent and bread), if they were both selected in the assortment. Finally, the fourth relationship, referred to as *allocation affinity*, requires the assignment of certain pairs of product categories to the same shelf (e.g., bath needs and bath tissue categories), if they were both selected in the assortment. Note that allocation affinity allows including only one of two categories (e.g. bath needs) in the assortment, and not including the other (e.g. bath tissues).

The main contributions of this paper are as follows: (i) We develop a model that jointly examines assortment planning and store-wide shelf space allocation decisions; (ii) we propose a heuristic approach that constructs a high-quality initial solution which is further enhanced using a large-scale neighborhood local search procedure; (iii) we test our model and solution approach on data related to a grocery store in the US and report managerial insights based on the optimized assortment and allocation decisions; and (iv) we demonstrate that our proposed methodology can provide near-optimal solutions (with an optimality gap below 0.5%) to large-scale problem instances involving up to 800 product categories and 100 shelves in manageable CPU times.

The remainder of the paper is organized as follows. Section 2 reviews the literature along the lines of the two main retailing decision problems of assortment and space planning. Section 3 provides a formal problem statement along with our notation and a mathematical programming formulation. Section 4 delineates the proposed optimization-based heuristic including an initialization procedure and a local improvement procedure. Section 5 discusses a study motivated by the examination of a grocery store and managerial insights into the optimized solution. Section 6 presents the results of an extensive computational study that was carried out to assess the efficacy of the proposed heuristic and the quality of its solutions. Section 7 concludes the paper with a summary of our findings and suggested directions for future research.

2. Literature review

Assortment and allocation decisions have mostly been investigated in isolation in the literature [23]. On the one hand, assortment decisions tend to focus on substitution and complementarity effects between products under deterministic or probabilistic consumer choice models, independently from store layout or shelf space capacity considerations. On the other hand, shelf space management problems generally assume a predetermined assortment of products and focus on their shelf space allocation, typically to a limited set of shelves. This paper constitutes a step toward the integration of these two decisions and provides an effective solution approach for the underlying combinatorial optimization problem.

2.1. Assortment planning

Assortment planning is a key decision-making process whereby a retailer determines a profitable product mix to carry within a store to maximize the opportunity for sale [10]. Given limited shelf space and budget restrictions, assortment planning seeks to determine the following: (i) Categories to carry in a store or department (breadth of the assortment); (ii) the number of stock keeping units (SKUs) to keep within a category (the depth of the assortment); and (iii) the inventory level for each SKU [36,39]. Accordingly, two types of assortment planning problems have been examined in the extant literature – focusing on the decision contexts in (i) and (ii), with the latter being more popular.

The popularity on assortment planning that examines the selection of SKUs is reflected by the surveys by [36] and [39]. This research stream typically investigates the trade-offs related to demand substitution across the assortment and the effect of size and composition of the assortment on operational cost. For example, [24] propose a model with a stochastic demand that jointly optimizes assortment size and order volumes for each selected item in the assortment by taking into account out-of-assortment and out-of-stock substitution effects. [25] also consider demand substitution effects in an integrated assortment and shelf-space allocation problem, respectively. This research area continues to receive high attention, with recent works falling in two main sub-steams: (1) Works that utilize stylized models and consider a stochastic demand over a single-period selling horizon; this applies to fashion goods (e.g., [40] and references therein) and (2) works that employ mathematical programming in a deterministic setting; this applies to fast-moving consumer goods (e.g., [16,17], and references therein).

Assortment planning related to which categories to offer in a store has received limited attention in the literature. Related works consider trade-offs associated with the effect of holding ordinary vs. unique products in the assortment on customers' store choice (e.g., [31,33,45]). Our work in this paper is distinct in that we consider trade-offs related to balancing the assortment offering from fast-moving, low-profit-margin products and slow-moving, high-margin impulsive products.

2.2. Space allocation

A large body of works in shelf space management addresses, at a more operational level, the construction of detailed planograms (e.g., [3]). These tasks usually take into account the impact of shelf space allocation decisions on demand levels with such considerations as space elasticity, location elasticity, or cross elasticity [1,7,9,11]. Hubner and Schaal [26] also take into account space elasticity considerations in a shelf-space allocation problem for high impulse products, where the demand is stochastic. The models which take into account these considerations are usually non-linear and are reformulated by using linearization techniques [15,21,29]. The focus of such studies is primarily on optimizing the allocation of shelf space to products using a set of capacitated shelves taken in isolation without store layout considerations [4,8,12,43,46,48]. Although [15] take into account all shelves and products in the store for the proposed shelf-space allocation problem, store-wide allocation considerations, such as traffic density of the store, are not taken into account. Some studies examine shelf-space allocation problem jointly along with other considerations. For example, Murray et al. [41] investigate an integrated problem to jointly optimize shelf-space allocation, display and pricing decisions. Frontoni et al. [14] propose an integer programming model for a shelf-space allocation problem to reduce out-of-stock occurrences. Hubner and Schaal [27] investigate a joint optimization concerning shelf-space allocation and in-store replenishment decisions by taking into account backroom space as well as shelf-space in the store. Some of these studies also consider integrated shelf space and assortment planning (e.g. [2,4,10,22,25,48]), as we do in this paper (interested readers can refer to [32] for a detailed review on assortment and shelf space planning models). However, the focus of these integrated works is on constructing operational planograms, examining the allocation of a single category to a single shelf. Our work is more tactical and focuses on selecting product categories and ascertaining their store-wide shelf space allocation.

Tactical store-wide shelf space allocation problems have received recent attention in the academic literature. Botsali [5] investigates decisions regarding retail store layout configurations and shelf space allocation of product categories utilizing a heuristic approach. Ke and van Ryzin [34] propose a heuristic methodology that examines a potential change in product locations of a category for a given store layout. Flamand et al. [13] introduce a tactical, store-wide shelf space management problem where shelves are comprised of smaller, adjacent segments that vary in attractiveness. A product category (e.g., tea or oil) is viewed at an aggregate level. The reward resulting from assigning a product category to a shelf depends on the attractiveness of the shelf segments it is assigned to and its aggregate gross profit. The authors develop an efficient, exact solution procedure that allocates *clustered* product categories to store shelves and solve large-scale problem instances in manageable times, in a way that maximizes the average profit from impulse product categories per customer basket. Ghoniem et al. [18] study the *unclustered* variant of the problem, where product categories are taken individually, as a generalized assignment problem with location/allocation considerations for which they develop preprocessing schemes, valid inequalities, and a branch-and-price algorithm that significantly outperforms CPLEX for small- and mid-sized instances. To address large-scale problem instances, Ghoniem et al. [19] develop an efficient variable neighborhood heuristic. In this paper, we address an integrated assortment planning and store-wide shelf space allocation problem also at a tactical level as in [5,13,34] and [18,19].

3. Mathematical programming formulation

This section introduces our notation along with an optimization model that forms a cornerstone for subsequent developments in the solution methodology.

The sets and indices which are considered in the proposed model are as follows:

- $\mathcal{N} = \{1, \ldots, n\}$: Set of product categories (e.g. coffee, tea, soda), indexed by j.
- $\mathcal{F} \subset \mathcal{N}$: Set of fast-movers, i.e., high-sales product categories. These can be determined using an 80-20 rule: 20% of product categories contribute nearly 80% of the expected sales and are included in $\mathcal{F}$. Such product categories are usually sold at low profit or below their market cost (loss leaders) to attract customers.
- $\mathcal{I} \equiv \mathcal{N} \setminus \mathcal{F}$: This set includes a variety of slow-moving product categories characterized by relatively higher profit margins, low expected sales, and good potential for impulse purchases.
- $\mathcal{L}$: Set of product category pairs $(j, j') \in \mathcal{N}^2$ which have *allocation disaffinity*. Such pairs of product categories can be simultaneously or individually chosen in the assortment, but they should not be allocated to the same shelf if they were both selected (e.g., detergent and bread).
- $\mathcal{H}_1$: Set of product category pairs $(j, j') \in \mathcal{N}^2$ which are related by *symmetric assortment affinity*. These product category pairs must be selected together or neither of them should be in the assortment. In the event they were both selected, they should be allocated to the same shelf (e.g., coffee and coffee filter).
- $\mathcal{H}_2$: Set of product category pairs $(j, j') \in \mathcal{N}^2$ which exhibit *asymmetric assortment affinity*. If product category j is selected, then j' must be selected as well and both product categories should be assigned to the same shelf. Note that j' may be selected without selecting j (e.g., cake icing and cake).
- $\mathcal{H}_3$: Set of product category pairs $(j, j') \in \mathcal{N}^2$ which exhibit *allocation affinity*. If product categories j and j' are selected in the assortment, both product categories should be assigned to the same shelf. Note that each product category may be selected individually in the assortment (e.g., bath needs and bath tissue).
- $\mathcal{B} \equiv \{1, \ldots, m\}$: Set of shelves indexed by i.
- $\mathcal{K}_i$: Set of consecutive shelf segments along shelf i ($i \in \mathcal{B}$), indexed by k.
- $\mathcal{K} \equiv \cup_{i \in \mathcal{B}} \mathcal{K}_i$: Set of all shelf segments in the store.

In addition, the parameters which are considered in the proposed model are as follows:

- ρ_j: Profit margin for product category j, $j \in \mathcal{N}$. This can be determined as the average profit margin of category j.
- v_j: Expected demand volume for product category j, $j \in \mathcal{N}$. This can be based on historical data.
- $\gamma_j \in (0, 1]$: This reflects the impulse purchase potential, i.e., the likelihood of a product category to be purchased by a customer impulsively if it was noticed on its shelf. Assuming that all slow-movers sales are based on impulse behavior, this parameter is similar to the impulse purchase rate in the literature (e.g., [28,30,35]). For fast-movers (product categories that are related to planned purchases), assuming that sales are based only on the location within the store and the visibility (in terms of assigned shelf space), one can set $\gamma_j = 1, \forall j \in \mathcal{F}$.
- f_k: Traffic density (attractiveness) of segment k, that is the likelihood of the segment to be visited by a customer, $k \in \mathcal{K}$, $0 < f_k \leq 1$. This can be estimated based on the location of segment k in the store. Segments in close proximity to end caps and/or store entrances are known to experience higher traffic.
- $\Phi_j \equiv \gamma_j \times \rho_j \times v_j$: The largest possible profit of product category j; Φ_j would be realized in a store segment which is very attractive, having a traffic density of 1 (which will be certainly visited by a customer), $j \in \mathcal{N}$. Note that for $j \in \mathcal{F}$, Φ_j reduces to $\rho_j \times v_j$.
- ℓ_j/u_j: Minimum/maximum space requirement for product category j, $j \in \mathcal{N}$. These requirements represent lower and upper limits of shelf space allocated to product category j which can be determined by retailers based on historical demand levels. Lower limits can be dictated in order to avoid stock-outs [12], induce attention on specific products (especially new products) [1,4,8], as well as creating a specific store image [8]. Upper limits may also be employed to reserve some space to refresh the product mix of the store [47]. We suggest to estimate the minimum requirement of each product category by $\ell_j \geq D_j \gamma_j \varphi_j$, where D_j is the demand of product category j in a given replenishment period (daily, weekly, etc., based on the frequency of replenishment), γ_j is the impulse potential of j, and φ_j is

the maximum facing size of j. The logic behind this estimate is the following. First, we note that $D_j\gamma_j$ represents the impulse portion of the given demand in a replenishment period. On the other hand, $D_j\gamma_j\varphi_j$ represents the maximum total shelf space amount of product category j for the estimated impulse demand.

- φ_j: Minimum space to be allocated to product category j along any segment it is assigned to. This may be considered as the maximum length of the unit facings of SKUs in a product category j, $j \in \mathcal{N}$. Given that the facings are usually "close", this assumption would lead to some empty slots (when applied to smaller SKUs), which would not be too significant. This is because we consider the problem at a tactical level by taking into account product categories, not individual SKUs. A detailed planogram for each product category, which considers the allocation of individual SKUs with different length of unit facings without any shelf gaps, could be designed by using our optimized allocation in a further study.
- α_i/β_i: Smallest/largest index of a segment that belongs to shelf i, $i \in \mathcal{B}$.
- c_k: Capacity of segment k, $k \in \mathcal{K}$.
- $c^{\max}$: The maximum shelf segment capacity among all shelf segments, $c^{\max} = \max_{k\in\mathcal{K}} c_k$.
- $\mathcal{C}_i \equiv \sum_{k\in\mathcal{K}_i} c_k$: Capacity of shelf i, $i \in \mathcal{B}$.

The decision variables of the proposed model are introduced as in the following:

- $x_{ij} \in \{0, 1\}$: $x_{ij} = 1$ if and only if product category j is assigned to shelf i, $\forall i \in \mathcal{B},\ j \in \mathcal{N}$.
- $y_{kj} \in \{0, 1\}$: $y_{kj} = 1$ if and only if product category j is assigned to shelf segment k, $\forall k \in \mathcal{K},\ j \in \mathcal{N}$.
- s_{kj}: Amount of space allocated to product category j along segment k, $\forall k \in \mathcal{K},\ j \in \mathcal{N}$.
- $z_{jj'} \in \{0, 1\}$: $z_{jj'} = 1$ if and only if product categories j and j' are selected in the assortment simultaneously, $\forall j, j' \in \mathcal{N}$.
- $q_{kj} \in \{0, 1\}$: $q_{kj} = 1$ if and only if product category j is assigned to both shelf segments k and $k+1$, $\forall k \in \mathcal{K} \setminus \{\beta_i : i \in \mathcal{B}\},\ j \in \mathcal{N}$.

Our integrated assortment planning and, store-wide, shelf space allocation problem is then formulated as the following MIP, denoted by **APSA**:

$$\textbf{APSA} : \text{Maximize} \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{N}} \Phi_j \frac{f_k s_{kj}}{c_k} \tag{1a}$$

$$\text{subject to} \sum_{i\in\mathcal{B}} x_{ij} \le 1, \quad \forall j \in \mathcal{N} \tag{1b}$$

$$\sum_{j\in\mathcal{N}} s_{kj} \le c_k, \quad \forall k \in \mathcal{K} \tag{1c}$$

$$\ell_j \sum_{i\in\mathcal{B}} x_{ij} \le \sum_{k\in\mathcal{K}} s_{kj} \le u_j \sum_{i\in\mathcal{B}} x_{ij}, \quad j \in \mathcal{N} \tag{1d}$$

$$\varphi_j y_{kj} \le s_{kj} \le \min\{c_k, u_j\} y_{kj}, \quad \forall j \in \mathcal{N}, k \in \mathcal{K} \tag{1e}$$

$$s_{k_2,j} \ge c_{k_2}(y_{k_1,j} + y_{k_3,j} - 1), \quad \forall j \in \mathcal{N}, i \in \mathcal{B}, k_1, k_2, k_3 \in \mathcal{K}_i | k_1 < k_2 < k_3 \tag{1f}$$

$$y_{kj} \le x_{ij}, \quad \forall i \in \mathcal{B}, j \in \mathcal{N}, k \in \mathcal{K}_i \tag{1g}$$

$$x_{ij} \le \sum_{k\in\mathcal{K}_i} y_{kj}, \quad \forall i \in \mathcal{B}, j \in \mathcal{N} \tag{1h}$$

$$q_{kj} \ge y_{kj} + y_{k+1,j} - 1, \quad \forall i \in \mathcal{B}, j \in \mathcal{N}, k \in \mathcal{K}_i \setminus \{\beta_i\} \tag{1i}$$

$$\sum_{j\in\mathcal{N}} q_{kj} \le 1, \quad \forall i \in \mathcal{B}, k \in \mathcal{K}_i \setminus \{\beta_i\} \tag{1j}$$

$$x_{ij} + x_{ij'} \le 1, \quad \forall (j, j') \in \mathcal{L}, i \in \mathcal{B} \tag{1k}$$

$$x_{ij} - x_{ij'} = 0, \quad \forall (j, j') \in \mathcal{H}_1, i \in \mathcal{B} \tag{1l}$$

$$x_{ij} \le x_{ij'}, \quad \forall (j, j') \in \mathcal{H}_2, i \in \mathcal{B} \tag{1m}$$

$$x_{ij} - x_{ij'} \le 1 - z_{jj'}, \quad \forall (j, j') \in \mathcal{H}_3, i \in \mathcal{B} \tag{1n}$$

$$x_{ij} - x_{ij'} \ge -1 + z_{jj'}, \quad \forall (j, j') \in \mathcal{H}_3, i \in \mathcal{B} \tag{1o}$$

$$z_{jj'} \le \sum_{i\in\mathcal{B}} x_{ij}, \quad \forall j, j' \in \mathcal{H}_3 \tag{1p}$$

$$z_{jj'} \le \sum_{i\in\mathcal{B}} x_{ij'}, \quad \forall j, j' \in \mathcal{H}_3 \tag{1q}$$

$$z_{jj'} \ge \sum_{i\in\mathcal{B}} x_{ij} + \sum_{i\in\mathcal{B}} x_{ij'} - 1, \quad \forall j, j' \in \mathcal{H}_3 \tag{1r}$$

$$\mathbf{x}, \mathbf{y}, \mathbf{z} \text{ binary}, \mathbf{s}, \mathbf{q} \ge 0. \tag{1s}$$

The objective function (1a) maximizes a measure of the overall store profitability and can be written as follows, giving visibility to fast-movers and impulse product categories: $\sum_{j\in\mathcal{F}} \sum_{k\in\mathcal{K}} \rho_j \upsilon_j \frac{f_k s_{kj}}{c_k} + \sum_{j\in\mathcal{I}} \sum_{k\in\mathcal{K}} \rho_j \upsilon_j \gamma_j \frac{f_k s_{kj}}{c_k}$. Here, $0 \le \frac{f_k s_{kj}}{c_k} \le 1$ represents the visibility of product category j on shelf segment k, that is the likelihood of product category j to be noticed by a customer on shelf segment k. The first term of the objective seeks a most convenient allocation of selected fast-movers in order to increase footprint and overall loyalty, whereas the second term focuses on assigning relatively high-impulse product categories to attractive shelf segments in order to generate an additional stream of revenue based on unplanned purchases. It is a compromise between these two goals that the model pursues.

Remark. The term $\frac{s_{kj}}{c_k}$ in the objective function (1a) represents the percentage of the capacity of segment k that is used by the product category j. If capacities of shelf segments, c_k, $\forall k \in \mathcal{K}$, are significantly different than each other in an instance, we suggest using the term $\frac{s_{kj}}{c^{\max}}$ instead. It ensures that it is different if a product category uses ϵ% of a very small segment or ϵ% of a very large segment.

In this study, for each product category j, its associated demand is deterministic, given at an average level over all possible scenarios. The model can be extended to the stochastic demand version by providing a demand distribution for each product category. In addition, the model can be extended by taking into account the effect of space-elasticity on the demand. The demand formulation along with the elasticity effects can be found in many studies in the literature (e.g., [8,20,26,29,48]). Constraint (1b) guarantees that each product category can be assigned to at most one shelf. Constraint (1c) requires the space along any shelf segment not to exceed its capacity. Constraint (1d) ensures that the total space allocated to a product category lies between its minimum and maximum space requirements if this product category is selected in the assortment. Constraint (1e) enforces conditional lower and upper bounds on the space a product category may be allocated along any shelf segment. Constraint (1f) requires any product category that is assigned to a pair of shelf segments, $k_1 < k_2$, along the same shelf to entirely fill any intermediate segments k_2 ($k_1 < k_2 < k_3$). Constraint (1g) guarantees that a product category cannot be assigned to a shelf segment unless it is allocated to its associated shelf. Constraint (1h) requires any product category that is assigned to a shelf to be assigned to at least one of its segments. Constraints (1i) and (1j) reflect that at most one product category can run over two contiguous segments. Constraint (1k) prevents pairs of incompatible product categories having a lack of affinity from being assigned to the same shelf. Note that this constraint does not prevent product category pairs which have a lack of affinity to be selected in the assortment simultaneously. Constraint (1l) requires product category pairs having symmetric assortment affinity to be assigned to the same shelf, otherwise, both product categories in the pair will not be selected in the assortment. Constraint (1m) addresses asymmetric assortment affinity and ensures that if product category j is selected, product category j' should be selected as well, and both product categories should be assigned to the same shelf. Constraints (1n) and (1o) require pairs of product categories having an allocation affinity to be assigned to the same shelf if they are selected in the assortment together. Note that these constraints do not prevent each product category to be selected in the assortment individually. Constraints (1p)–(1r) are linearization constraints that linearize the multiplication of $\sum_{i\in\mathcal{B}} x_{ij}$ and $\sum_{i\in\mathcal{B}} x_{ij'}$. Note that $\sum_{i\in\mathcal{B}} x_{ij}$ is 1 if product category j is selected in the assortment, 0 otherwise. Hence the multiplication of these two terms is 1 if both product categories are selected in the assortment, 0 otherwise. Constraint (1s) enforces logical binary and non-negativity restrictions on the decision variables.

Further, consider the following set: $\mathcal{R} \equiv \{(k_1, k_2, j) \in \mathcal{K} \times \mathcal{K} \times \mathcal{N} : k_1 < k_2 \text{ and } \sum_{h=k_1+1}^{k_2-1} c_h > u_j - 2\varphi_j\}$: The set of triplets ($k_1$, k_2, j) such that product category j cannot be assigned to segments k_1 and k_2 simultaneously, because the maximum space j can be allocated, u_j, cannot cover all intermediate segments between k_1 and k_2. To enhance the formulation, the following valid inequality can be added to the model:

$$y_{k_1 j} + y_{k_2 j} \le 1, \quad \forall (k_1, k_2, j) \in \mathcal{R}. \tag{2}$$

This constraint eliminates the simultaneous assignments of product category j to segments k_1 and k_2 if the maximum space requirement of j cannot cover all intermediate segments between k_1 and k_2. In our experience, the inclusion of this valid inequality significantly enhances the model tractability. Constraint (1f) can be also preprocessed for a similar reason and also contributes to improving the tractability of the model:

$$\begin{aligned} & s_{k_2 j} \ge c_{k_2}(y_{k_1,j} + y_{k_3,j} - 1), \forall j \in \mathcal{N}, i \in \mathcal{B}, \\ & \quad k_1, k_2, k_3 \in \mathcal{K}_i | k_1 < k_2 < k_3, (k_1, k_3, j) \notin \mathcal{R}. \end{aligned} \tag{3}$$

The computational results of testing these valid inequalities are reported in the Appendix A.

The discussed assortment planning and store-wide shelf allocation problem can be viewed as a discrete knapsack problem where each knapsack is divided into segments that are characterized by different attractiveness levels and capacities. However, the problem at hand is computationally more challenging than classical knapsack type problems due to the knapsack discretization it involves in order to reflect different levels of shelf space attractiveness or visibility to the consumer. The proposed model is NP-hard. In fact, the special case where products have fixed shelf space (i.e., lower bound equals upper bound for the shelf space of any product) and where all shelves segments are identical reduces to a classical set packing problem, which is known to be NP-hard. Moreover, the problem of finding optimal shelf space allocation where the assortment is fixed *a priori*, is shown to be NP-hard in [13] even for a single-shelf problem.

4. Optimization-based heuristic approach

In this section, we develop an optimization-based heuristic procedure coupled with local improvement schemes. The proposed heuristic invokes two procedures: (i) An initialization procedure that constructs an initial feasible solution by iteratively optimizing single-shelf problems (i.e. the single-shelf variant of Model APSA) and (ii) a local improvement procedure that iteratively seeks an enhanced solution by re-optimizing selected subsets of shelves.

4.1. Initialization procedure

For any shelf $i \in \mathcal{B}$, let $\lambda_i \equiv \frac{\sum_{k\in\mathcal{K}_i} f_k c_k}{\sum_{k\in\mathcal{K}_i} c_k}$ be a measure of its relative attractiveness. Let $\sigma = (\sigma_1, \ldots, \sigma_m)$ denote a permutation of the m shelves that is obtained by sorting them in non-increasing order of relative attractiveness. A constructive heuristic requires optimally allocating product categories to the m shelves in turn according to the permutation ordering. A pseudocode of the proposed methodology is delineated in Algorithm 1.

Algorithm 1 Initialization procedure.

1: Input σ. Set $i = 1$, $\mathcal{S} = \emptyset$ ($\mathcal{S}$ refers to the set of selected product categories)
2: Set $i^* \leftarrow \sigma_i$
3: Find an optimal allocation for shelf i^*, regarding product categories in $\mathcal{N} \setminus \mathcal{S}$. Let $\mathcal{N}_{i^*}$ be the set of product categories that are packed into σ_i.
4: Set $S \leftarrow S \cup \mathcal{N}_{i^*}$
5: **for all** $((j_1, j_2) \in \mathcal{H}_3)$ **do**
6:     **if** $(j_1 \in \mathcal{N}_{i^*})$ **then**
7:         Set $S \leftarrow S \cup \{j_2\}$
8:     **end if**
9:     **if** $(j_2 \in \mathcal{N}_{i^*})$ **then**
10:         Set $S \leftarrow S \cup \{j_1\}$
11:     **end if**
12: **end for**
13: **if** $(i = m)$ or $(\mathcal{S} = \mathcal{N})$ **then**
14:     Stop.
15: **end if**.
16: Set $i \leftarrow i + 1$, Go to Step 3.

In Step 3, an optimal packing of shelf i^* can be constructed by solving the following single-shelf variant of Model APSA where the binary variable w_j equals 1 if and only if product category j is assigned to shelf i^*:

$$\textbf{SSP}(\mathbf{i}^*) : \text{Maximize} \sum_{k\in\mathcal{K}_{i^*}} \sum_{j\in\mathcal{N}\setminus\mathcal{S}} \Phi_j \frac{f_k s_{kj}}{c_k} \tag{4a}$$

$$\text{subject to} \sum_{j\in\mathcal{N}\setminus\mathcal{S}} s_{kj} \le c_k, \quad \forall k \in \mathcal{K}_{i^*} \tag{4b}$$

$$\ell_j w_j \le \sum_{k\in\mathcal{K}_{i^*}} s_{kj} \le u_j w_j, \quad \forall j \in \mathcal{N} \setminus \mathcal{S} \tag{4c}$$

$$\varphi_j y_{kj} \le s_{kj} \le \min\{c_k, u_j\} y_{kj}, \quad \forall j \in \mathcal{N} \setminus \mathcal{S}, k \in \mathcal{K}_{i^*} \tag{4d}$$

$$s_{k_2} \geq c_{k_2}(y_{k_1,j} + y_{k_3,j} - 1), \quad \forall j \in \mathcal{N} \setminus \mathcal{S}, k_1, k_2, k_3 \in \mathcal{K}_{i^*} | k_1 < k_2 < k_3, (k_1, k_3, j) \notin \mathcal{R} \tag{4e}$$

$$y_{kj} \leq w_j, \quad \forall j \in \mathcal{N} \setminus \mathcal{S}, k \in \mathcal{K}_{i^*} \tag{4f}$$

$$w_j \leq \sum_{k \in \mathcal{K}_{i^*}} y_{kj}, \quad \forall j \in \mathcal{N} \setminus \mathcal{S} \tag{4g}$$

$$q_{kj} \geq y_{kj} + y_{k+1,j} - 1, \quad \forall j \in \mathcal{N} \setminus \mathcal{S}, \; k \in \mathcal{K}_{i^*} \setminus \{\beta_{i^*}\} \tag{4h}$$

$$\sum_{j \in \mathcal{N} \setminus \mathcal{S}} q_{kj} \leq 1, \quad \forall k \in \mathcal{K}_{i^*} \setminus \{\beta_{i^*}\} \tag{4i}$$

$$w_j + w_{j'} \leq 1, \quad \forall (j, j') \in \mathcal{L} \tag{4j}$$

$$w_j - w_{j'} = 0, \quad \forall (j, j') \in \mathcal{H}_1 \tag{4k}$$

$$w_j \leq w_{j'}, \quad \forall (j, j') \in \mathcal{H}_2 \tag{4l}$$

$$y_{k_1 j} + y_{k_2 j} \leq 1, \quad \forall (k_1, k_2, j) \in \mathcal{R} | k_1, k_2 \in \mathcal{K}_{i^*} \tag{4m}$$

$$\mathbf{w}, \mathbf{y}, \mathbf{q} \text{ binary}, \mathbf{s} \geq 0. \tag{4n}$$

Note that the categories in set $\mathcal{H}_3$ are not included in the model SSP(i^*) and are treated separately in the Algorithm 1. Recall that $\mathcal{H}_3$ is a set of product category pairs (j_1, j_2), where either product categories j_1 and j_2 are included in the assortment together and allocated to the same shelf, or j_1 is included in the assortment, while j_2 is not, or j_1 is not included in the assortment, while j_2 is included. Hence, in the initialization procedure, one needs to know if either j_1 or j_2 is assigned individually to a shelf in the previous iterations. For example, assume that two single shelf problems are iteratively solved. If j_1 and j_2 are not assigned to Shelf 1, they can be assigned to Shelf 2 individually or simultaneously. However if one of them, for instance, j_1, is assigned to Shelf 1, then j_2 cannot be assigned to Shelf 2 (if both are selected in the assortment, they have to be on the same shelf). Hence, Constraints (1n) and (1o) cannot be directly included into the single shelf model, since it has to be recorded whether one of these product categories is assigned to a shelf or not in the previous iterations.

4.2. MIP-based re-optimization procedure

This improvement procedure takes as input the feasible solution constructed using the initialization procedure. It then iteratively selects τ shelves using a probabilistic selection scheme. In every iteration, available shelves, i.e. shelves that have not been selected yet for re-optimization, are sorted in a non-increasing order of their current objective value contribution. Then, the sorted, available shelves are evenly divided into τ "levels", where the first level comprises the subset of shelves having the greatest objective values and the last level includes the shelves having the least objective values. (If the total number of available shelves is not divisible by τ, one or more levels may include slightly more shelves than others.) From each level, one shelf is randomly selected, resulting in a selection of τ shelves. The associated τ-shelf Model APSA has the following input: the τ shelves, their allocated product categories and shelf space, and all product categories that are not in the assortment (if any). In re-optimizing the content of these τ shelves, some product categories may be swapped between the shelves, others may be eliminated from their assortment, new product categories may be introduced, and the shelf space of all selected product categories may be re-adjusted. The algorithm proceeds with the iterative selection and re-optimization of τ different shelves, until the number of available shelves for re-optimization becomes smaller than τ. At this point, the algorithm checks whether any stopping criterion is met. If this is the case, it terminates; otherwise, it starts a new loop where all shelves are deemed available for re-optimization. The pseudo-code of this procedure is delineated in Algorithm 2.

Algorithm 2 MIP-based re-optimization procedure.

Derive an upper bound by solving the continuous relaxation of the problem
Construct an initial feasible solution with a total reward $r^* = \sum_{i \in \mathcal{B}} r_i$ using Algorithm 1
incumbent $\leftarrow r^*$
$\hat{\kappa} \leftarrow (x, y, s)$
$\tau \leftarrow 4$
repeat
    tempset $\leftarrow \mathcal{B}$
    while (|tempset| > $|\mathcal{B}|$ (mod) τ) **do**
        Let $\Delta = (\Delta_1, \ldots, \Delta_m)$ denote a permutation of the shelves in tempset that is obtained by sorting them innon-increasing order of their current objective value contribution
        Ω = round($|\Delta|/\tau$)
        for all $k = 1, \ldots, \tau$ **do**
            Randomly select a shelf i between $\Delta_{(k-1)\Omega+1}$ and $\Delta_{k\Omega}$
            tempset $\leftarrow$ tempset - $\{i\}$
        end for
        Solve Model APSA to re-optimize the selected subset of shelves by considering all the product categories currently allocated on them as well as the unassigned product categories
        incumbent $\leftarrow r^*_{new}$
        $\hat{\kappa}^{new} \leftarrow (x^{new}, y^{new}, s^{new})$
        $\hat{\kappa} \leftarrow \hat{\kappa}^{new}$
    end while.
until at least one stopping criterion is met

The following stopping criteria are used in our study:

- If the relative gap between the LP-based upper bound and the incumbent solution is less than or equal to ϵ%.
- If the algorithm traverses all shelves for a specified number of times without any improvement in the incumbent.
- If the algorithm reaches a specified time limit.

5. Case study: a mid-sized grocery store

This section presents a case study, based on our interaction with retailers in the Northeastern USA, that has motivated our work and demonstrates the usefulness of our proposed methodology. We consider a retailer who owns multiple stores and desires to open a mid-sized store. Given a total of 100 candidate product categories, listed and numbered in Table 1 under their associated groups, the retailer seeks to identify an optimal assortment along with a store-wide allocation for the grid layout represented in Fig. 2.

Product categories differ by their expected sales volumes (based on historical or projected data from other stores), profit margins, potential for impulse purchases, and lower/upper bounds on their shelf space requirements. Fig. 1 displays the demand volumes of individual product categories their cumulative curve for a 6-month period, based on the data provided in Table 2. The darker bars in the figure identify the fast-movers – the high-sales categories that collectively contribute nearly 80% of the total expected sales in the input data. Specifically, fast-movers include the following 20 product categories for this grocery store with a cumulative sales volume of 72% of all input categories: Bread (9), Canned Vegetables (17), Cigarettes (23), Juice (26), Soda (27), Cheese (32), Milk

Table 1
Groups and their associated categories in a grocery store.

#	Group	Category
1	Alcohol	Liquor (1), Champagne (3), Vodka (5), Whiskey (6), Wine (7)
2	Light alcohol	Beer (2), Energy Drinks (4)
3	Bread	Croissant (8), Bread (9), Sandwich (10), Bagel (11), Toast (12), Bread Crumbs (73)
4	Candy	Chewing Gum (13), Lollipop (14), Marshmallow (15), Candy (16), Chocolate (21), Chocolate Chips (22)
5	Ready made food	Ready Made Food (18), Frozen Sea Food (62), Dinners (71), Pizza (72)
6	Breakfast	Hot Cereals (19), Cold Cereals (20), Peanut Butter & Jelly (74), Honey & Syrups (97)
7	Cigarettes	Cigarettes (23), Cigars (24)
8	Cold beverages	Iced Tea (25), Juice (26), Soda (27), Water (64)
9	Dairy 1	Butter (31), Eggs (33), Milk (34), Cookie Dough (65)
10	Dairy 2	Sour Cream (66), Yogurt (67)
11	Cheese	Cheese (32), Packed Cheese (35), Specialty Cheese (36)
12	Canned food	Canned Meat (37), Canned Sea Food (61)
13	Desserts	Boxed Desserts (39), Cakes (40), Ready Made Desserts (41), Spreaded Desserts (42), Pies and Toppings (68)
14	Meat	Sliced Deli (38), Packed Meat (43), Unpacked Meat (44)
15	Hot beverages & cookies	Cookies (28), Gourmet Cookies (29), Biscuits (30), Coffee (45), Tea (46), Herbal Tea (93)
16	Nuts & potato chips	Chips (47), Nuts (48), Popcorn (49), Snacks (91), Rice Cakes (94)
17	Pasta	Pasta (50), Pasta Sauce (92)
18	Powders	Grain (51), Rice (52), Soup (53), Spice (54), Sugar-Salt (55), Flour (56)
19	Sauces & syrups	Creams (57), Dips (58), Oil (59), Sweet Sauce (60)
20	Vegetables	Canned Vegetables (17), Vegetables (63)
21	Frozen	Ice Cream (69), Ice (70)
22	Bath tissue	Bath Tissue (77)
23	Paper towels	Paper Towels (79)
24	Bath needs	Facial Tissue (76), Bath Needs (80)
25	Paper & plastic needs	Cups & Plates (75), Wraps & Bags (78)
26	Cleaning supplies	Fabric Softeners (81), Laundry Detergents (85)
27	Household essentials	Household Cleaner (82), Bleach (83), Wipes (84), Dish Detergents (86)
28	Condiments	Vinegar (87), Ketchup (88), Pickles & Olives (89), Salad Dressings (90)
29	Canned fruit	Canned Fruit (95)
30	Cake supplies	Cake Decorations (96), Cake Mixes (98)
31	Baby needs	Baby Food (99), Diapers (100)

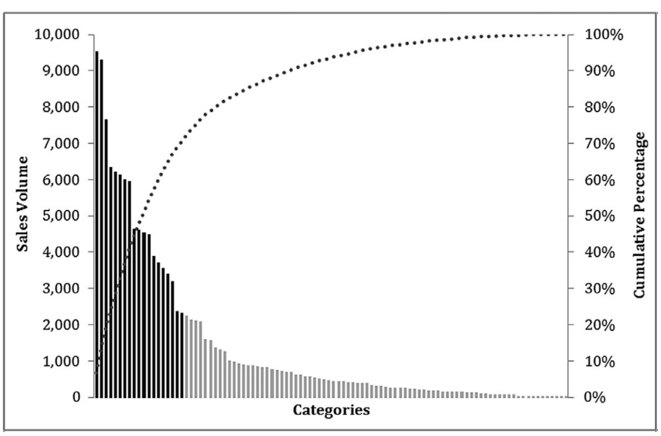

Fig. 1. Sales volume and cumulative percentage of product categories.

(34), Packed Cheese (35), Specialty Cheese (36), Unpacked Meat (44), Coffee (45), Vegetables (63), Water (64), Ice (70), Dinners (71), Pizza (72), Household Cleaner (82), Salad Dressings (90), Pasta Sauce (92), and Canned Fruit (95). In contrast, product categories having a strong impulse purchase potential, such as chocolate or cookies, are commonly purchased in an unplanned fashion.

We have also estimated impulse purchase rate of a product category j, γ_j, based on various studies in the literature. Studies suggested that hedonic products that are associated with "pleasure" (e.g., chocolates or cakes) [28,30] as well as low-priced products, and products on sale (Kacen et al. 2012) tend to have higher impulse purchase potential. In our study, impulse purchase rates fell in three levels: low [0, 0.1), medium [0.1, 0.4), and high [0.4, 0.5]. In contrast, for any fast-mover, a value of 1 was assigned in lieu of an impulse purchase rate. In addition, the lower and upper limits are calculated by adding ± 10% of the currently/historically allocated shelf space amount of each product category.

In this paper, we consider the attractiveness (traffic density) of the different shelf segments in a grocery store, with three attractiveness levels – high, medium, and low. Shelf segments which are closer to end-of-aisles, entrances, or cashiers tend to be more visible, better visited, and therefore more attractive than shelf segments located in the middle of aisles. The attractiveness f_k of segment k was estimated using a uniform distribution over the interval [0, 0.5) for low attractiveness, [0.5, 0.8] for medium attractiveness, and (0.8, 1] for high attractiveness. Note that, the unit of measure for the store layout parameters (c_k, s_{kj}, ℓ_j, u_j and φ_j) is in "feet".

With a preset time limit of one CPU hour, CPLEX produced a solution with a 13.2% optimality gap, while the proposed heuristic yielded a solution with a 2.1% optimality gap in less than 4 minutes that is illustrated in Fig. 2. For the reader's convenience, fast-movers are circled and high-impulse categories are squared in Fig. 2. The assortment in the optimized solution comprises 72 (out of 100) product categories of which 19 are fast-movers. An analysis of the solution reveals the following for this mid-sized store:

- Almost all (19/20) fast-movers (circled in Fig. 2) were included in the assortment due to their relatively high volumes. This meets the retailer's objective of including high-sales product categories that can attract customers and strengthen their loyalty. It can, therefore, be pertinent to write Constraint (1b) as an equality constraint for fast-movers, thereby forcing them

Table 2
Demand volume (in units) of product categories for a grocery store.

j	Volume, v_j	Category	Volume, v_j	Category	Volume, v_j	Category	Volume, v_j
1	16	**26**	5,969	**51**	265	**76**	1,608
2	536	**27**	7,657	**52**	310	**77**	1,267
3	8	**28**	786	**53**	16	**78**	867
4	443	**29**	134	**54**	432	**79**	738
5	129	**30**	223	**55**	262	**80**	2
6	116	**31**	211	**56**	401	**81**	197
7	40	**32**	6,009	**57**	175	**82**	3,413
8	77	**33**	572	**58**	332	**83**	1,015
9	9,530	**34**	6,144	**59**	407	**84**	39
10	255	**35**	4,500	**60**	8	**85**	822
11	522	**36**	3,907	**61**	396	**86**	920
12	10	**37**	262	**62**	78	**87**	181
13	847	**38**	764	**63**	4,641	**88**	697
14	223	**39**	87	**64**	3,705	**89**	481
15	38	**40**	13	**65**	2,145	**90**	2,241
16	146	**41**	626	**66**	2,113	**91**	988
17	4,620	**42**	158	**67**	1,324	**92**	3,203
18	126	**43**	381	**68**	158	**93**	298
19	77	**44**	6,348	**69**	2,100	**94**	301
20	76	**45**	6,212	**70**	2,322	**95**	3,556
21	890	**46**	192	**71**	4,529	**96**	74
22	633	**47**	822	**72**	2,384	**97**	170
23	9,315	**48**	706	**73**	568	**98**	1,378
24	465	**49**	29	**74**	1,588	**99**	438
25	98	**50**	415	**75**	914	**100**	169

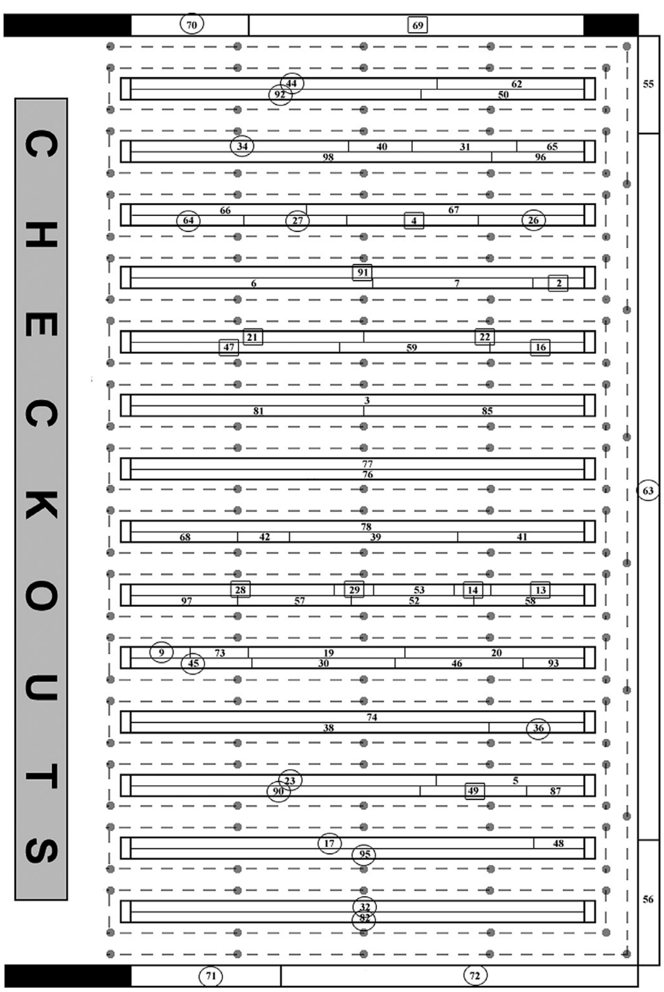

Fig. 2. Assortment and shelf space allocation using Model APSA.

into the assortment and letting the objective function guide their shelf space allocation.

- Fast-movers were assigned to some of the most attractive shelf segments and their allocated shelf space was set at their maximum space requirements. For example, Ice (70), Dinners (71), and Pizza (72) are three fast-movers that were allocated to the highly visited shelf segments on the store periphery. This has the potential of improving shopping convenience and customer loyalty.
- The solution comprises 13 high-impulse purchase categories. These were assigned the remaining attractive segments and were also allocated their maximum shelf space requirements. For instance, Ice Cream (69) was allocated to attractive shelf segments on the store periphery and the maximum possible shelf space amount was given to it. This has the potential of making such product categories highly visible to consumers, thereby stimulating a stream of lucrative unplanned purchases.
- The solution achieves a trade-off between the space allocation for fast-movers and that for high impulse product categories on a shelf, thereby jointly promoting customer convenience and impulse profit. It is also interesting to note that some high-impulse product categories share some shelves with fast-movers. For instance, Salad Dressing (90), a fast-mover, shares its shelf with Popcorn (49) which is a high-impulse product category. This creates further impulse sales opportunities as customers who are attracted to the shelf to buy fast-movers may also be tempted to consider high-impulse product categories in their vicinity.

Note that, in this case study, while the data (store layout, product categories, their profit margins) is realistic, the retailer was not available to discuss our improved shelf space allocation. To strengthen the practicality of the model, we provided an additional group-based solution by grouping the product categories based on their affinities (groups are given in Table 1), as most retailers assign those related product categories together in practice. For example, Vinegar, Ketchup, Pickles and Salad Dressing are considered as a group since in practice, these product categories are usually located on the same shelf. In this setting, the assortment decisions are separately made for each group, and the selected con-

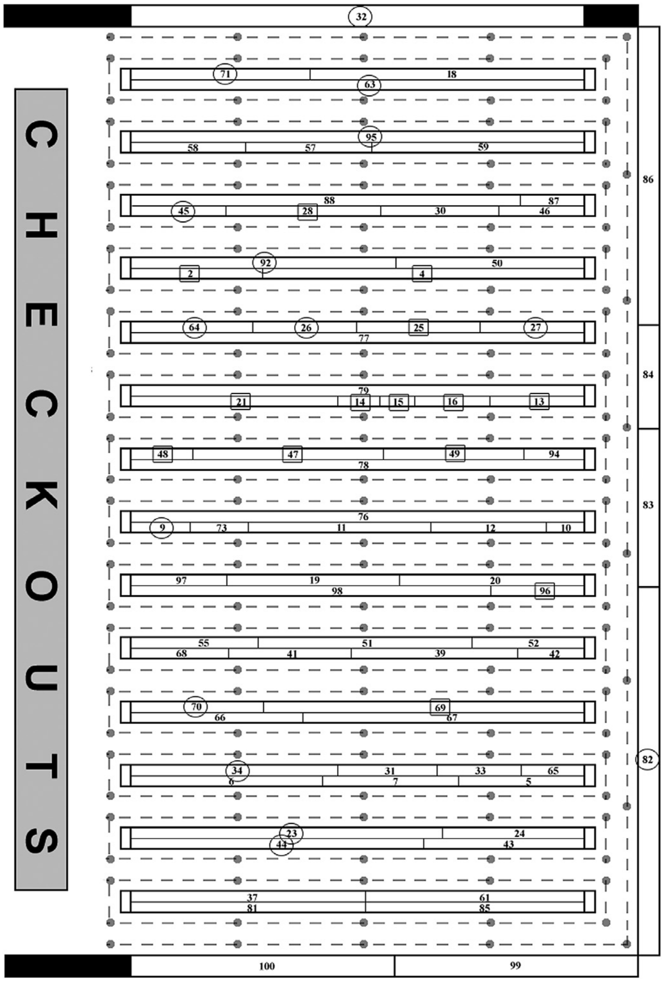

Fig. 3. Group-based allocation using Model APSA.

tent is optimally assigned to a shelf together. In addition, related groups, which include products having affinities with each other (e.g., Dairy 1 and Dairy 2 in Table 1) are allocated to the same aisle. The group-based procedure adopts and extends the procedure in [13] as detailed in the Appendix B. The group-based solution is shown in Fig. 3.

The group-based procedure yielded an objective value which is 15% less than the one obtained by the proposed heuristic. This is expected since additional restrictions are considered in the group-based model, i.e., many product categories are enforced to be allocated together. Fig. 3 revealed this practical solution while also yielding similar insights as our previously obtained solution. For example, similar to our previous solution, most of the fast-movers (circled in Fig. 3) were included in the assortment due to their relatively high volumes. Similarly, fast-movers were assigned to some of the most attractive shelf segments at their maximum space requirements. For instance, Milk (34) and Meat (44) are two fast-movers that were allocated to the highly visited shelf segments. In addition, high-impulse purchase categories such as Ice Cream (69) or Cookies (28) were also allocated to the remaining attractive shelf segments at their maximum possible shelf space amount. The group-based model also yielded a practical solution due to defined affinities among product categories of each group. For example, in the Alcohol group, product categories Vodka (5), Whiskey (6), Wine (7) are included in the assortment and assigned on the same shelf. Similarly, in the Candy group, Gum (13), Lollipop (14), Marshmallow (15), Candy (16), and Chocolate (21) are selected in the assortment and assigned together.

We also would like to emphasize that many retailers in the U.S do not always follow the traditional allocation patterns. For example, a well-known grocery chain in the U.S. allocates some of its frozen product categories along with ordinary categories (not frozen) in the same area. Particularly, bread, peanut butter/jelly, frozen pizza and snack cakes are allocated to the same area in their grocery stores. Similarly, another well-known grocery chain allocates bread and jelly together. On the other hand, it also allocates coffee and tortillas in the same area with those product categories. In their stores, tortillas are not allocated together with the Hispanic products category. This grocery chain also allocates snacks together with the eggs and dairy product categories. In addition, in the same grocery chain's stores, spices/sugar and juice/sports drinks are allocated together in the same area. Furthermore, it allocates candy category together with soup and canned vegetables.

6. Computational study

In this section, we assess the computational performance of the proposed model and the usefulness of the proposed heuristic approach for larger problem instances. Our testbed is constituted of five sets of randomly generated instances ranging from 30 shelves/240 items to 100 shelves/800 items. This broad spectrum of instance sizes can help the decision-maker define the granularity of the model, whereby items could be, at an aggregate level, categories or, at a finer level, individual products or SKUs. In this paper, we examine the problem at a tactical level where products represent categories (e.g., coffee, bread, etc.). It is worth to note that zooming in/out of the problem for different granularity levels may require additional considerations to be taken into account, therefore it may require additional constraints to be added into the proposed model. In case, individual SKUs are included in the analysis, then one needs two additional modeling constructs related to (i) constraints which keeps SKUs in a category remain adjacent on the same shelf, and (ii) reflecting the effect of substitution, which typically implies that a particular SKU would have a larger demand in a smaller assortment (i.e., out-of-assortment demand substitution).

Model APSA was coded in AMPL and solved using CPLEX 12.6, whereas our heuristic methodology was implemented in C++ under Visual Studio. All computational runs were made on a Dell XPS 8300 workstation having a Intel Core(TM) i7-2600 CPU 3.40 GHz processor and 12 GB of RAM.

6.1. Description of testbed

Our testbed includes five sets, each consisting of ten instances of the same size and sharing the same data characteristics. The data generation scheme for our testbed is as follows:

- All instances include shelves having the same capacity of 18 units ($\mathcal{C}_i = 18,\ i \in \mathcal{B}$). Each shelf is discretized into three segments of equal capacity ($c^{\max} = 6$).
- Sets 1–5 respectively include 30, 40, 50, 60, and 100 shelves and 240, 320, 400, 480, and 800 product categories.
- The minimum space requirement for product category j, denoted by ℓ_j, is randomly generated using a uniform distribution over the range of $[1, \frac{\mathcal{C}_i}{6}]$. The generated value of ℓ_j is rounded to the nearest integer.
- The maximum requirement for product category j, u_j, is randomly generated using a uniform distribution over the range of $[\ell_j, \frac{\mathcal{C}_i}{3}]$. The generated value of u_j is rounded to the nearest integer.
- The largest possible profit of product category j, Φ_j, is randomly generated using a uniform distribution over the range of [1, 25] with 2 decimal places.

Table 3
Performance of the heuristic vs. CPLEX.

Set	$(\lvert\mathcal{B}\rvert, \lvert\mathcal{N}\rvert)$	Inst.	Model APSA			Heuristic $\tau = 4,\ \epsilon = 0.5\%$	
			CPU(s)	B&B/C	Gap(%)	CPU(s)	Gap (%)
Set 1	(30, 240)	1	3,600	15,496	0.31	73.5	0.47
		2	3,600	14,778	0.18	54.3	0.49
		3	3,600	46,546	0.06	74.2	0.40
		4	3,600	16,619	0.12	51.4	0.48
		5	3,600	11,438	0.76	65.3	0.49
		6	3,600	84,95	0.47	77.9	0.48
		7	3,600	10,750	0.44	39.6	0.47
		8	3,600	10,930	0.26	73.3	0.43
		9	3,600	21,724	0.23	52.6	0.45
		10	3,600	9,738	0.65	60.2	0.43
Set 2	(40, 320)	1	3,600	2,661	0.66	138.5	0.41
		2	3,600	7,747	0.55	129.8	0.50
		3	3,600	2,602	0.79	132.8	0.45
		4	3,600	9,048	0.41	86.7	0.41
		5	3,600	3,426	0.96	169.8	0.42
		6	3,600	4,499	0.78	176.2	0.44
		7	3,600	5,764	1.01	157.6	0.43
		8	3,600	17,686	0.42	151.5	0.42
		9	3,600	16,110	0.65	139.1	0.47
		10	3,600	3,285	0.74	137.5	0.45
Set 3	(50, 400)	1	3,600	2,564	1.81	208.0	0.43
		2	3,600	5,410	1.39	180.4	0.45
		3	3,600	7,172	1.05	241.7	0.50
		4	3,600	1,654	1.61	309.8	0.48
		5	3,600	1,668	1.40	243.9	0.40
		6	3,600	3,021	1.17	196.8	0.41
		7	3,600	6,424	1.13	233.3	0.43
		8	3,600	3,910	1.07	261.2	0.39
		9	3,600	3,995	1.38	198.9	0.48
		10	3,600	5,431	1.38	265.1	0.47
Set 4	(60, 480)	1	3,600	1,706	2.12	350.4	0.48
		2	3,600	2,266	2.06	458.4	0.47
		3	3,600	1,698	1.59	345.3	0.43
		4	3,600	1,641	1.87	467.2	0.44
		5	3,600	1,824	1.40	449.2	0.43
		6	3,600	1,649	2.36	442.4	0.49
		7	3,600	2,047	2.19	363.7	0.49
		8	3,600	2,107	1.78	358.0	0.45
		9	3,600	1,754	1.94	439.9	0.46
		10	3,600	1,854	2.38	484.9	0.48
Set 5	(100, 800)	1	3,600	36	2.53	1,848.4	0.44
		2	3,600	12	2.56	1,862.1	0.43
		3	3,600	388	5.45	2,043.5	0.49
		4	3,600	18	2.76	2,355.7	0.47
		5	3,600	66	2.73	1,973.2	0.45
		6	3,600	123	2.62	1,803.6	0.43
		7	3,600	398	5.52	1,924.5	0.49
		8	3,600	0	5.73	2,302.9	0.44
		9	3,600	52	2.24	1,820.5	0.46
		10	3,600	24	1.89	1,987.2	0.49

- The attractiveness for segment k, f_k, is randomly generated such that every 20% of shelves share the same level of attractiveness t, where t = 5%, 25%, 45%, 65%, and 85%, respectively. In addition, the middle shelf segments have relatively lower attractiveness than the other segments, since in a retail store, shelf segments which are closer to the end-of-aisle are more attractive. The attractiveness of middle shelf segments is generated using a uniform distribution over the range of $[t, t + 0.05]$ with two decimal places and that of the end segments is generated using a uniform distribution over the range of $[t + 0.06, t + 0.1]$ with two decimal places.
- The minimum allocated space for product category j, φ_j, is set to 0.1 units in all our instances. Here, we assume that one facing of a product category can also be allocated across two consecutive shelf segments. In that case, we assume that the minimum space that can be allocated to each product category is around 0.1 units.

Note that we follow some guidelines from the literature while generating the instances to make them more challenging. For example, in the literature, for bin-packing problems, it is suggested that the instances including product categories having space requirements that are close to $\mathcal{C}/3$, where $\mathcal{C}$ is the capacity of a knapsack, tend to be harder [44]. Since the model we have proposed has similarities to bin-packing problems, this suggestion has also been applied to our data generation process.

6.2. Base results

Table 3 reports and contrasts our computational results for solving Model APSA using CPLEX with a time limit of 3,600 CPU sec-

Table 4
The effect of the neighborhood size on the performance of the heuristic.

Set	($\lvert\mathcal{B}\rvert$, $\lvert\mathcal{N}\rvert$)	Inst.	Model APSA		Heuristic $\tau = 2$, $\epsilon = 0.5\%$		Heuristic $\tau = 3$, $\epsilon = 0.5\%$		Heuristic $\tau = 4$, $\epsilon = 0.5\%$	
			CPU(s)	Gap(%)	CPU(s)	Gap (%)	CPU(s)	Gap (%)	CPU(s)	Gap (%)
Set 1	(30,240)	1	3,600	0.31	1,000	0.70	81.9	0.48	73.5	0.47
		2	3,600	0.18	1,000	0.68	81.4	0.49	54.3	0.49
		3	3,600	0.06	1,000	0.63	57.7	0.46	74.2	0.40
		4	3,600	0.12	1,000	0.63	104.7	0.48	51.4	0.48
		5	3,600	0.76	375.0	0.50	75.3	0.49	65.3	0.49
		6	3,600	0.47	1,000	0.62	175.6	0.48	77.9	0.48
		7	3,600	0.44	122.9	0.50	40.0	0.48	39.6	0.47
		8	3,600	0.26	1,000	0.67	116.7	0.49	73.3	0.43
		9	3,600	0.23	1,000	0.50	71.1	0.49	52.6	0.45
		10	3,600	0.65	230.8	0.50	84.3	0.42	60.2	0.43
Set 2	(40,320)	1	3,600	0.66	1,000	0.71	237.6	0.47	138.5	0.41
		2	3,600	0.55	1,000	0.59	156.4	0.45	129.8	0.50
		3	3,600	0.79	1,000	0.52	172.5	0.48	132.8	0.45
		4	3,600	0.41	145.3	0.49	114.8	0.45	86.7	0.41
		5	3,600	0.96	1,000	0.73	303.3	0.48	169.8	0.42
		6	3,600	0.78	1,000	0.55	284.2	0.50	176.2	0.44
		7	3,600	1.01	1,000	0.67	166.5	0.48	157.6	0.43
		8	3,600	0.42	1,000	0.59	259.4	0.49	151.5	0.42
		9	3,600	0.65	600.4	0.49	173.9	0.47	139.1	0.47
		10	3,600	0.74	1,000	0.53	163.2	0.49	137.5	0.45
Set 3	(50,400)	1	3,600	1.81	1,000	0.67	292.1	0.46	208.0	0.43
		2	3,600	1.39	1,000	0.52	298.1	0.50	180.4	0.45
		3	3,600	1.05	1,000	0.69	299.3	0.46	241.7	0.50
		4	3,600	1.61	1,000	0.62	379.2	0.46	309.8	0.48
		5	3,600	1.40	1,000	0.63	427.5	0.49	243.9	0.40
		6	3,600	1.17	1,000	0.54	257.5	0.50	196.8	0.41
		7	3,600	1.13	1,000	0.67	279.8	0.49	233.3	0.43
		8	3,600	1.07	1,000	0.63	388.6	0.48	261.2	0.39
		9	3,600	1.38	1,000	0.69	367.1	0.45	198.9	0.48
		10	3,600	1.38	1,000	0.71	422.3	0.45	265.1	0.47
Set 4	(60,480)	1	3,600	2.12	1,000	0.75	454.7	0.47	350.4	0.48
		2	3,600	2.06	1,000	0.99	590.3	0.49	458.4	0.47
		3	3,600	1.59	1,000	0.71	581.3	0.48	345.3	0.43
		4	3,600	1.87	1,000	0.84	520.4	0.50	467.2	0.44
		5	3,600	1.40	1,000	0.66	411.2	0.45	449.2	0.43
		6	3,600	2.36	1,000	0.81	591.3	0.49	442.4	0.49
		7	3,600	2.19	1,000	0.78	542.7	0.47	363.7	0.49
		8	3,600	1.78	1,000	0.61	575.5	0.46	358.0	0.45
		9	3,600	1.94	1,000	0.73	531.8	0.44	439.9	0.46
		10	3,600	2.38	1,000	0.93	535.6	0.49	484.9	0.48
Set 5	(100,800)	1	3,600	2.53	1,000	1.39	2,958.9	0.48	1,848.4	0.44
		2	3,600	2.56	1,000	1.48	2,995.1	0.49	1,862.1	0.43
		3	3,600	5.45	1,000	1.34	3,078.4	0.49	2,043.5	0.49
		4	3,600	2.76	1,000	1.55	2,676.5	0.49	2,355.7	0.47
		5	3,600	2.73	1,000	1.35	2,317.9	0.46	1,973.2	0.45
		6	3,600	2.62	1,000	0.98	2,243.0	0.45	1,803.6	0.43
		7	3,600	5.52	1,000	1.42	3,013.1	0.48	1,924.5	0.49
		8	3,600	5.73	1,000	1.54	2,289.2	0.47	2,302.9	0.44
		9	3,600	2.24	1,000	1.31	2,310.4	0.47	1,820.5	0.46
		10	3,600	1.89	1,000	1.59	2,642.3	0.48	1,987.2	0.49

onds and our proposed optimization-based heuristic over the five data sets introduced in Section 6.1. Column 1 in Table 3 provides a reference to the data set and Column 2 specifies the number of shelves and the number of product categories. Column 3 provides the instance number within its data set. Columns 4–6 report the CPU time (in seconds), the number of branch&bound/cut nodes explored and the % solver optimality gap at termination, respectively. Further, Columns 7–8 report the CPU time (in seconds) and the % optimality gap for the proposed heuristic. The % optimality gap for the proposed heuristic is computed using the best upper bound (BUB) found by CPLEX at termination, i.e. % Optimality Gap $= \frac{\text{BUB}-\text{Incumbent}}{\text{BUB}} \times 100$. The heuristic results are reported for $\tau = 4$, i.e. the re-optimization procedure considered four-shelf subproblems, and the optimality target was set to 0.5%.

CPLEX failed to solve all 50 instances in our testbed to optimality within the preset time limit of 3,600 CPU seconds. As the number of shelves and product categories was increased, the time spent at each node also increased. Specifically, for the instances of Set 5, the number of branch-and-bound/cut nodes explored is smaller than those of the other sets. For all of our instances, during one CPU hour time limit, the dual bound was not improved upon much. Since a good solution was not found during early iterations, most of the effort was spent to improve the solution during later iterations. For example, for Set 5, the duality gap at the root node was 5.7% at an average, while the final gap was 3.4% at an average in one CPU hour time limit. During the time limit, no improvement occured on the upper bound. Hence, all of the effort to enhance the optimality gap from 5.7% to 3.4% was spent to improve the incumbent. The solver optimality gaps at termination gradually

Table 5
The Effect of Target Optimality Gap on the Performance of the Heuristic.

Set	($\|\mathcal{B}\|, \|\mathcal{N}\|$)	*Inst.*	*Model APSA*		*Heuristic* $\tau=4,\ \epsilon=1.5\%$		*Heuristic* $\tau=4,\ \epsilon=1\%$		*Heuristic* $\tau=4,\ \epsilon=0.5\%$	
			CPU(s)	*Gap(%)*	*CPU(s)*	*Gap (%)*	*CPU(s)*	*Gap (%)*	*CPU(s)*	*Gap (%)*
Set 1	(30,240)	1	3,600	0.31	28.9	1.06	41.6	0.74	73.5	0.47
		2	3,600	0.18	31.3	1.32	43.4	0.64	54.3	0.49
		3	3,600	0.06	31.6	0.98	31.6	0.98	74.2	0.40
		4	3,600	0.12	29.9	0.81	30.8	0.81	51.4	0.48
		5	3,600	0.76	32.5	1.00	33.3	1.00	65.3	0.49
		6	3,600	0.47	38.9	1.36	54.3	0.74	77.9	0.48
		7	3,600	0.44	29.6	0.63	30.3	0.63	39.6	0.47
		8	3,600	0.26	32.3	1.06	47.8	0.62	73.3	0.43
		9	3,600	0.23	16.7	1.34	32.2	0.83	52.6	0.45
		10	3,600	0.65	29.9	0.93	30.6	0.93	60.2	0.43
Set 2	(40,320)	1	3,600	0.66	68.6	0.98	69.1	0.98	138.5	0.41
		2	3,600	0.55	68.9	0.99	69.4	0.99	129.8	0.50
		3	3,600	0.79	66.2	0.81	66.7	0.81	132.8	0.45
		4	3,600	0.41	62.0	0.61	62.2	0.61	86.7	0.41
		5	3,600	0.96	69.1	1.13	100.6	0.71	169.8	0.42
		6	3,600	0.78	87.9	1.05	114.1	0.70	176.2	0.44
		7	3,600	1.01	65.0	1.05	86.7	0.71	157.6	0.43
		8	3,600	0.42	82.3	0.99	83.3	0.99	151.5	0.42
		9	3,600	0.65	82.6	0.83	83.1	0.83	139.1	0.47
		10	3,600	0.74	75.7	1.00	102.4	0.71	137.5	0.45
Set 3	(50,400)	1	3,600	1.81	122.5	0.81	123.2	0.81	208.0	0.43
		2	3,600	1.39	107.6	0.85	107.7	0.85	180.4	0.45
		3	3,600	1.05	122.3	1.08	162.8	0.70	241.7	0.50
		4	3,600	1.61	130.5	1.01	177.1	0.74	309.8	0.48
		5	3,600	1.40	131.7	0.83	131.8	0.83	243.9	0.40
		6	3,600	1.17	124.6	0.76	125.2	0.76	196.8	0.41
		7	3,600	1.13	113.3	0.93	113.1	0.93	233.3	0.43
		8	3,600	1.07	153.3	1.02	151.2	0.76	261.2	0.39
		9	3,600	1.38	116.0	0.81	118.1	0.81	198.9	0.48
		10	3,600	1.38	119.1	0.89	120.2	0.89	265.1	0.47
Set 4	(60,480)	1	3,600	2.12	204.9	0.91	204.8	0.91	350.4	0.48
		2	3,600	2.06	226.2	1.01	322.0	0.80	458.4	0.47
		3	3,600	1.59	205.9	0.90	206.3	0.90	345.3	0.43
		4	3,600	1.87	250.6	1.05	339.2	0.78	467.2	0.44
		5	3,600	1.40	222.3	0.88	222.7	0.88	449.2	0.43
		6	3,600	2.36	314.6	0.94	315.4	0.94	442.4	0.49
		7	3,600	2.19	214.7	1.01	292.2	0.62	363.7	0.49
		8	3,600	1.78	212.8	0.82	212.5	0.82	358.0	0.45
		9	3,600	1.94	225.6	0.89	225.9	0.89	439.9	0.46
		10	3,600	2.38	241.7	0.94	242.5	0.94	484.9	0.48
Set 5	(100,800)	1	3,600	2.53	1,106.9	0.84	1,103.4	0.84	1,848.4	0.44
		2	3,600	2.56	1,113.4	0.87	1,116.5	0.87	1,862.1	0.43
		3	3,600	5.45	1,244.8	0.93	1,254.2	0.93	2,043.5	0.49
		4	3,600	2.76	1,121.0	0.88	1,117.6	0.88	2,355.7	0.47
		5	3,600	2.73	1,185.5	0.82	1,191.1	0.82	1,973.2	0.45
		6	3,600	2.62	1,092.1	0.76	1,099.5	0.76	1,803.6	0.43
		7	3,600	5.52	1,126.7	0.89	1,128.2	0.90	1,924.5	0.49
		8	3,600	5.73	1,125.4	0.90	1,128.2	0.90	2,302.9	0.44
		9	3,600	2.24	1,112.0	0.88	1,112.7	0.88	1,820.5	0.46
		10	3,600	1.89	1,119.2	0.87	1,122.0	0.87	1,987.2	0.49

deteriorated as the size of the instances increased, ranging from 0.48% for Set 1 to 3.4% for Set 5 on average. In contrast, our heuristic consistently yielded near-optimal solutions, with an optimality gap of 0.45% on average and a computational effort ranging from about 1 CPU minute for Set 1 (30 shelves, 240 product categories) to nearly 30 CPU minutes for Set 5 (100 shelves, 800 product categories). As such, the heuristic constructs higher quality solutions compared to CPLEX (excluding Set 1) and achieves substantial CPU savings, ranging from 44.6% for Set 5 to 98.2% for Set 1, on average.

6.3. Effect of neighborhood size, target optimality gap and affinity relationship

The goal of this section is to conduct sensitivity analysis with respect to the following three aspects: (i) The impact of τ, the number of shelves selected in the re-optimization procedure in Section 4.2, on the computational effort of the heuristic, (ii) the effect of setting increasingly tighter optimality gaps for the heuristic, and (iii) the effect of adding different (dis)affinity relationships. The first consideration is henceforth referred to as the neighborhood size, as τ dictates a neighborhood of solutions that can be locally explored. Intuitively, as τ increases, the re-optimization subproblem may become more computationally challenging, as it enables the exploration of larger, and more promising, neighborhoods of solutions. As far as the optimality target is concerned, we seek to assess the increment in the computational effort of the heuristic as the optimality gap is set to 1.5%, 1%, and 0.5%. For the (dis)affinity considerations, we tested base instances by adding them five pairs of allocation disaffinity ($\mathcal{L}$), five pairs of symmetric assortment affinity ($\mathcal{H}$), five pairs of asymmetric assortment affinity ($\mathcal{H}_2$), and five pairs of allocation affinity ($\mathcal{H}_3$), respectively, as well as adding these 20 pairs all together.

Table 6
The effect of affinity relationships for the heuristic.

Set	*Inst.*	No Affinity		$\mathcal{L}$		$\mathcal{H}$		$\mathcal{H}_2$		$\mathcal{H}_3$		$(\mathcal{L}, \mathcal{H}, \mathcal{H}_2, \mathcal{H}_3)$	
		CPU(s)	*Gap(%)*	*CPU(s)*	*Gap (%)*	*CPU(s)*	*Gap (%)*	*CPU(s)*	*Gap (%)*	*CPU(s)*	*Gap (%)*	*CPU(s)*	*Gap (%)*
Set 1	1	73.5	0.47	64.2	0.48	120.0	0.71	86.5	0.44	74.3	0.48	120.0	0.75
	2	54.3	0.49	56.3	0.43	120.0	0.56	59.0	0.45	84.5	0.38	120.0	0.75
	3	74.2	0.40	61.6	0.48	66.2	0.46	47.1	0.49	57.6	0.42	97.9	0.49
	4	51.4	0.48	48.1	0.48	120.0	0.55	114.2	0.47	92.2	0.47	120.0	0.77
	5	65.3	0.49	45.4	0.48	120.0	0.51	80.2	0.47	48.8	0.46	120.0	0.47
	6	77.9	0.48	119.7	0.48	120.0	0.97	120.0	0.44	71.2	0.46	120.0	1.01
	7	39.6	0.47	32.1	0.45	120.0	0.58	34.8	0.40	36.2	0.44	120.0	0.49
	8	73.3	0.43	79.7	0.47	120.0	0.60	54.9	0.46	89.2	0.49	120.0	0.63
	9	52.6	0.45	67.5	0.41	120.0	0.65	65.0	0.48	89.6	0.49	120.0	0.86
	10	60.2	0.43	68.2	0.49	120.0	0.57	74.0	0.48	74.8	0.42	120.0	0.65
Set 2	1	138.5	0.41	110.9	0.42	180.0	0.58	180.0	0.50	130.8	0.42	180.0	0.60
	2	129.8	0.50	134.8	0.49	157.4	0.43	168.3	0.45	130.5	0.42	180.0	0.50
	3	132.8	0.45	122.0	0.49	180.0	0.48	140.7	0.45	180.0	0.49	180.0	0.59
	4	86.7	0.41	150.3	0.36	180.0	0.57	114.4	0.43	123.2	0.46	180.0	0.62
	5	169.8	0.42	145.6	0.47	180.0	0.68	152.8	0.49	131.3	0.49	180.0	0.94
	6	176.2	0.44	167.0	0.37	146.3	0.44	180.0	0.50	180.0	0.40	180.0	0.69
	7	157.6	0.43	116.5	0.48	180.0	0.73	156.9	0.44	106.9	0.39	180.0	0.79
	8	151.5	0.42	147.7	0.46	180.0	0.58	132.8	0.48	163.4	0.43	180.0	0.69
	9	139.1	0.47	135.2	0.44	180.0	0.49	161.4	0.42	139.3	0.40	141.7	0.45
	10	137.5	0.45	100.6	0.40	168.5	0.47	180.0	0.53	150.6	0.46	180.0	0.85
Set 3	1	208.0	0.43	282.3	0.38	267.6	0.45	285.7	0.47	294.2	0.48	300.0	0.48
	2	180.4	0.45	253.9	0.40	203.9	0.47	300.0	0.42	205.7	0.45	229.0	0.47
	3	241.7	0.50	209.9	0.49	282.5	0.45	259.5	0.37	223.8	0.47	300.0	0.55
	4	309.8	0.48	230.4	0.42	300.0	0.69	300.0	0.45	269.8	0.46	300.0	0.73
	5	243.9	0.40	266.3	0.45	300.0	0.64	232.5	0.47	222.3	0.49	300.0	0.66
	6	196.8	0.41	231.0	0.42	300.0	0.45	188.4	0.49	189.7	0.47	287.8	0.47
	7	233.3	0.43	285.9	0.46	300.0	0.55	234.7	0.49	300.0	0.47	300.0	0.49
	8	261.2	0.39	246.8	0.49	262.0	0.48	300.0	0.44	250.7	0.46	283.7	0.48
	9	198.9	0.48	235.0	0.49	272.1	0.46	240.1	0.48	272.1	0.47	300.0	0.55
	10	265.1	0.47	222.4	0.41	300.0	0.48	300.0	0.49	231.3	0.41	300.0	0.63
Set 4	1	350.4	0.48	337.9	0.44	337.8	0.48	414.9	0.44	388.1	0.45	480.0	0.48
	2	458.4	0.47	414.1	0.47	468.1	0.48	480.0	0.55	407.0	0.43	480.0	0.65
	3	345.3	0.43	335.4	0.37	331.2	0.44	335.7	0.41	391.1	0.38	480.0	0.45
	4	467.2	0.44	342.9	0.45	398.0	0.45	480.0	0.50	427.3	0.43	480.0	0.56
	5	449.2	0.43	360.4	0.49	480.0	1.03	319.0	0.49	450.8	0.45	480.0	0.55
	6	442.4	0.49	405.8	0.44	434.6	0.43	483.0	0.47	345.0	0.48	480.0	0.50
	7	363.7	0.49	433.4	0.49	473.2	0.46	453.1	0.43	437.5	0.45	480.0	0.58
	8	358.0	0.45	317.0	0.44	357.3	0.42	406.9	0.36	405.2	0.47	480.0	0.50
	9	439.9	0.46	311.3	0.41	505.3	0.48	480.0	0.42	359.0	0.41	480.0	0.57
	10	484.9	0.48	355.9	0.48	480.0	0.58	448.1	0.46	466.0	0.44	480.0	0.52
Set 5	1	1,848.4	0.44	1,893.3	0.41	1,831.0	0.46	1,886.1	0.47	1,777.5	0.42	2,400.0	0.48
	2	1,862.1	0.43	1,943.0	0.44	1,930.8	0.43	1,912.7	0.48	1,886.6	0.42	2,058.5	0.42
	3	2,043.5	0.49	1,915.1	0.45	2,030.4	0.47	2,367.5	0.42	2,111.2	0.48	2,232.7	0.46
	4	2,355.7	0.47	2,400.0	0.48	2,400.0	0.64	2,004.5	0.47	1,988.4	0.49	2,400.0	0.60
	5	1,973.2	0.45	1,952.5	0.43	1,964.3	0.47	2,239.0	0.47	1,829.9	0.46	2,400.0	0.51
	6	1,803.6	0.43	1,794.6	0.41	1,842.0	0.47	1,883	0.41	1,765.5	0.40	1,833.9	0.41
	7	1,924.5	0.49	1,930.5	0.43	2,400.0	0.48	1,849.8	0.43	2,292.8	0.42	2,400.0	0.46
	8	2,302.9	0.44	2,147.7	0.47	2,382.9	0.45	2,216.2	0.43	1,913.2	0.49	2,400.0	0.56
	9	1,820.5	0.46	2,316.7	0.45	2,170.6	0.49	2,195.7	0.48	1,774.5	0.42	2,400.0	0.54
	10	1,987.2	0.49	1,863.9	0.43	2,400.0	0.48	1,930.0	0.47	1,862.1	0.42	2,300.5	0.48

To examine the impact of the neighborhood size, we consider pairs ($\tau = 2$), triplets ($\tau = 3$), and quadruplets of shelves ($\tau = 4$) with an optimality gap set to 0.5% for these three cases. The results are reported in Table 4 for Model APSA and the heuristic for $\tau = 2, 3,$ or 4. For $\tau = 2$, the heuristic failed to solve 45/50 instances within 1000 seconds time limit and yielded an optimality gap of 0.79% on average. It is our observation that the re-optimization of pairs of shelves results in significant improvements in early iterations, followed by a tailing-off effect and slow, marginal improvements in the later iterations. By expanding the neighborhood to triplets of shelves ($\tau = 3$), near-optimal solutions become accessible for all instances in Sets 1–4, with a run time ranging from one CPU minute for Set 1 to about 8–9 CPU minutes for Set 4. To fully explore the potential of $\tau = 3$, no time limit was imposed for Set 5, and the heuristic yielded solutions within 0.5% optimal over 40 CPU minutes on average. Increasing the neighborhood size to $\tau = 4$ achieved a 25.7% CPU savings over $\tau = 3$ and resulted in a faster convergence to solutions within 0.45% optimality gap on average.

Table 5 reports our results on the computational impact of tighter optimal gap targets for the heuristic. Considering $\tau = 4$, the optimality gap target is set in turn to $\epsilon = 1.5\%$, 1%, and 0.5%. For $\epsilon = 1.5\%$, the heuristic yielded an overall optimality of 0.93% as 34/50 instances exhibited an optimality gap less than or equal to 1%. Hence, CPU times for $\epsilon = 1.5\%$ and $\epsilon = 1\%$ are quite comparable. To meet the more challenging optimality gap $\epsilon = 0.5\%$, a 40.4% CPU time increase over the case of $\epsilon = 1\%$ was necessary.

Finally, Table 6 shows the computational impact of adding different affinity considerations to the base instances. The instances were tested by the heuristic where $\tau = 4$, $\epsilon = 0.5\%$. In addition, for each data set, a time limit has been employed, which is comparable to the maximum solution time of the instances yielded by the heuristic. Specifically, 120, 180, 300, 480, 2400 CPU seconds time limits have been employed, respectively, for the Sets 1–5.

Results show that if some product category pairs have allocation disaffinity (in set $\mathcal{L}$), asymmetric assortment affinity (in set $\mathcal{H}_2$), and allocation affinity (in set $\mathcal{H}_3$), the computational tractability was not significantly affected. However, adding product category pairs into set $\mathcal{H}$, which have symmetric assortment affinity, and also adding pairs into all affinity sets, $\mathcal{L}$, $\mathcal{H}$, $\mathcal{H}_2$, $\mathcal{H}_3$ simultaneously, increased the final optimality gap. Specifically, the optimality gap increased from 0.45% to 0.53% at an average over all sets when the symmetric assortment affinity ($\mathcal{H}$) pairs were added, and the optimality gap increased to 0.58% at an average when all affinity considerations were added together. Note that for each type of affinity, five pairs were added to the instances. Hence, the ratio of the number of affinity pairs to the number of products is higher for smaller data sets. Consequently, the effect of the affinities on smaller data sets is larger. Particularly, the optimaliy gap increased from 0.45% to 0.61% at an average for Set 1, whereas it increased to 0.48% for Set 5 when the five symmetric assortment affinity ($\mathcal{H}$) pairs were added into the instances. Similarly, the optimality gap increased to 0.68% at an average for Set 1, whereas it increased to 0.49% for Set 5, when all affinity considerations were added together.

7. Conclusion

This paper rests on recent developments in shelf space management and introduces a joint assortment planning and store-wide shelf space allocation problem. The problem poses computational challenges for large-scale problem instances due to its combinatorial nature. The proposed solution strategy prescribes to the decision-maker assortment and shelf space allocation decisions that can strengthen in-store footprint, result in highly convenient configurations, and stimulate impulse purchases. It is our observation, based on a motivational case study, that the constructed solution strikes a good balance between fast-movers and high-impulse product categories. Prime shelf segments are reserved to fast-movers, which has the benefit of increasing in-store traffic and improving the shopping experience. The remaining attractive shelf segments are allocated to high-impulse product categories, thereby making them more visible to consumer and increasing the likelihood of unplanned purchases. It is also our observation that fast-movers and high-impulse product categories share several shelves. This can bring some high-impulse product categories to the consumer's attention, as they visit a shelf to buy a fast-mover.

Whereas our case study involved a mid-sized grocery store with about 30 shelves and 100 candidate categories, our computational study explores the usefulness of our methodology for larger instances, involving up to 100 shelves and 800 product categories. Our proposed heuristic consistently yielded near-optimal solutions within optimality gaps below 0.5% for the 50 instances in our testbed in manageable times. By comparison, it drastically outperforms CPLEX with a time limit both in terms of the quality of the solution and the computational time. Our sensitivity analysis reveals that the discovery of a near-optimal solution that meets a target optimality gap (e.g., 1.5%, 1%, and 0.5%) is achievable with the heuristic and requires an expansion of the size of the neighborhood (subset of shelves) that is explored by the re-optimization subproblem.

Our model could be altered using different objectives for other application areas such as warehouse allocation problems (e.g., [37]). Particularly, affinity considerations could be applied to such problems to allocate related SKU groups, which are frequently purchased together, to the same area in a warehouse in order to minimize the total number of visited areas or total time to assemble the incoming orders.

We recommend for future research the integration of assortment planning and shelf space allocation with the added consideration of store layout planning. In particular, it would be interesting to expand our mathematical model to account for shelf orientation and layout decisions with the aim of maximizing customer traffic and product category visibility throughout the store. In addition to integrating shelf allocation and configuration decisions, future extensions could also include (i) end-caps allocation, as retailers typically pay special attention to such allocation; and (ii) integrated vertical and horizontal space planning, 3D allocations, which integrate classic vertical planogram-like practices on vertical shelf planning with horizontal planning such as the one discussed in this paper.

Appendix A. Performance of the valid inequalities

Table 7 reports and compares the performance of the proposed MIP model without any valid inequalities, with valid inequality (2) and with valid inequalities (2) and (3) together. Column 1 in Table 7 provides a reference to the data set and Column 2 provides the instance number within its data set. Columns 3–5 report the CPU time (in seconds), the number of branch&bound/cut nodes explored and the % solver optimality gap at termination, respectively for the MIP model without any valid inequalities. Further, Columns 6 and 7 report the CPU time (in seconds) and the % optimality gap for the MIP model along with the valid inequality (2). Further, Columns 8 and 9 report the CPU time (in seconds) and the % optimality gap for the MIP model along with the valid inequalities (2) and (3).

According to the results, the model along with the valid inequalities (2) and (3) provided better optimality gaps in one CPU hour time limit. Although the results are comparable for small size instances, as the size of the instances increases, using (2) and (3) together becomes more advantageous. Hence, both of the valid inequalities are included in the Model APSA.

Table 7
Performance of the valid inequalities.

Set	*Inst.*	*Base MIP*			*MIP+(2)*		*MIP+(2)+(3)*	
		CPU(s)	*B&B/C*	*Gap(%)*	*B&B/C*	*Gap(%)*	*B&B/C*	*Gap(%)*
Set 1	1	3,600	15,182	0.74	10,859	0.38	15,496	0.31
	2	3,600	79,568	0.23	13,196	0.58	14,778	0.18
	3	3,600	33,118	0.33	9,925	0.43	46,546	0.06
	4	3,600	10,843	0.24	7,158	0.42	16,619	0.12
	5	3,600	12,096	0.72	15,168	0.26	11,438	0.76
	6	3,600	34,080	0.31	9,084	0.88	8,495	0.47
	7	3,600	16,254	0.15	23,268	0.13	10,750	0.44
	8	3,600	6,478	0.51	5,672	0.88	10,930	0.26
	9	3,600	24,601	0.67	44,925	0.26	21,724	0.23
	10	3,600	13,225	0.65	10,366	0.64	9,738	0.65

(*continued on next page*)

Table 7 (*continued*)

Set	*Inst.*	*Base MIP*			*MIP+(2)*		*MIP+(2)+(3)*	
		CPU(s)	*B&B/C*	*Gap(%)*	*B&B/C*	*Gap(%)*	*B&B/C*	*Gap(%)*
Set 2	1	3,600	3,743	0.76	9,857	0.82	2,661	0.66
	2	3,600	19,181	0.53	17,974	0.55	7,747	0.55
	3	3,600	20,578	0.43	3,041	1.06	2,602	0.79
	4	3,600	4,484	1.61	10,610	0.70	9,048	0.41
	5	3,600	2,872	0.91	2,723	0.50	3,426	0.96
	6	3,600	4,121	0.76	2,465	0.88	4,499	0.78
	7	3,600	8,282	0.67	2,394	0.86	5,764	1.01
	8	3,600	4,319	1.69	4,099	0.70	17,686	0.42
	9	3,600	19,255	0.68	10,225	0.82	16,110	0.65
	10	3,600	12,262	0.51	14,695	0.49	3,285	0.74
Set 3	1	3,600	2,847	1.25	3,615	1.49	2,564	1.81
	2	3,600	5,207	1.06	4,268	1.23	5,410	1.39
	3	3,600	7,237	1.21	5,337	1.48	7,172	1.05
	4	3,600	1,674	1.95	1,765	1.25	1,654	1.61
	5	3,600	3,227	1.13	1,783	0.94	1,668	1.40
	6	3,600	4,014	1.09	3,136	1.47	3,021	1.17
	7	3,600	4,753	1.35	4,298	1.19	6,424	1.13
	8	3,600	4,182	0.92	3,651	1.28	3,910	1.07
	9	3,600	3,073	1.75	2,081	1.28	3,995	1.38
	10	3,600	5,326	0.96	6,124	1.52	5,431	1.38
Set 4	1	3,600	2,152	2.54	1,914	2.01	1,706	2.12
	2	3,600	1,667	1.70	1,672	1.63	2,266	2.06
	3	3,600	2,614	1.53	1,927	1.71	1,698	1.59
	4	3,600	1,684	2.02	2,133	1.96	1,641	1.87
	5	3,600	1,646	1.20	1,749	1.19	1,824	1.40
	6	3,600	1,803	2.46	1,688	2.51	1,649	2.36
	7	3,600	1,612	1.43	1,653	2.27	2,047	2.19
	8	3,600	1,701	1.11	1,670	1.40	2,107	1.78
	9	3,600	1,819	2.12	1,637	1.93	1,754	1.94
	10	3,600	1,803	1.56	1,743	2.52	1,854	2.38
Set 5	1	3,600	566	5.22	330	5.10	36	2.53
	2	3,600	53	6.07	3	5.96	12	2.56
	3	3,600	207	5.13	183	5.40	388	5.45
	4	3,600	10	2.67	8	2.85	18	2.76
	5	3,600	257	7.43	323	5.96	66	2.73
	6	3,600	121	2.74	413	5.22	123	2.62
	7	3,600	35	2.92	91	5.85	398	5.52
	8	3,600	12	6.07	42	2.19	0	5.73
	9	3,600	21	5.27	8	5.74	52	2.24
	10	3,600	155	4.70	36	2.29	24	1.89

Appendix B. Group-based model

This section extends the group-based procedure that was proposed by [13]. In this section, we first present single-group, single-shelf version of the Model APSA. This model is similar to the Model SSP proposed in Section 4.1. Let $\mathcal{Z}_g$ be a set of product categories in group g. For this problem, we define 31 groups as shown in Table 1. The Model APSA(g, i) determines the optimal assortment among product categories of group g and optimally allocates them to shelf i.

$$\mathbf{APSA}(\mathbf{g},\mathbf{i}): \text{Maximize} \sum_{k\in\mathcal{K}_i}\sum_{j\in\mathcal{Z}_g}\Phi_j\frac{f_k s_{kj}}{c_k} \tag{5a}$$

$$\sum_{j\in\mathcal{Z}_g} s_{kj} \le c_k, \quad \forall k \in \mathcal{K}_i \tag{5b}$$

$$\ell_j w_j \le \sum_{k\in\mathcal{K}_i} s_{kj} \le u_j w_j, \quad \forall j \in \mathcal{Z}_g \tag{5c}$$

$$\varphi_j y_{kj} \le s_{kj} \le \min\{c_k, u_j\} y_{kj}, \quad \forall j \in \mathcal{Z}_g, k \in \mathcal{K}_i \tag{5d}$$

$$s_{k_2} \ge c_{k_2}(y_{k_1,j} + y_{k_3,j} - 1), \quad \forall j \in \mathcal{Z}_g, k_1, k_2, k_3 \in \mathcal{K}_i | k_1 < k_2 < k_3, (k_1, k_3, j) \notin \mathcal{R} \tag{5e}$$

$$y_{kj} \le w_j, \quad \forall j \in \mathcal{Z}_g, k \in \mathcal{K}_i \tag{5f}$$

$$w_j \le \sum_{k\in\mathcal{K}_i} y_{kj}, \quad \forall j \in \mathcal{Z}_g \tag{5g}$$

$$q_{kj} \ge y_{kj} + y_{k+1,j} - 1, \quad \forall j \in \mathcal{Z}_g, \ k \in \mathcal{K}_i \setminus \{\beta_i\} \tag{5h}$$

$$\sum_{j\in\mathcal{Z}_g} q_{kj} \le 1, \quad \forall k \in \mathcal{K}_i \setminus \{\beta_i\} \tag{5i}$$

$$y_{k_1 j} + y_{k_2 j} \le 1, \quad \forall (k_1, k_2, j) \in \mathcal{R} | k_1, k_2 \in \mathcal{K}_i \tag{5j}$$

$$\mathbf{w}, \mathbf{y}, \mathbf{q} \text{ binary}, \ \mathbf{s} \ge 0. \tag{5k}$$

(5a)-(5k) can be interpreted in a same manner with the Model APSA proposed in Section 3. To introduce the group-based model, consider the following notation:

- $\mathcal{J}$: Set of product category groups.
- μ_{gi}: The optimal objective value of Model APSA(g, i) by optimally selecting the assortment of group g and allocating them to shelf i, $\forall g \in \mathcal{J}, i \in \mathcal{B}$.

- $\mathcal{M}$: Set of product group pairs (g, g') that must be allocated to the same aisle due to high affinity in sales (e.g., Dairy 1 and Dairy 2 in Table 1).
- $\mathcal{A}$: Set of aisles.
- $\mathcal{N}_a$: Set of shelves located in the same aisle, $a \in \mathcal{A}$.

We define the following binary variable for the group-based model:

- $\theta_{gi} \in \{0, 1\}$: $\theta_{gi}=1$ if and only if group g is assigned to a shelf i, $\forall g \in \mathcal{J}, i \in \mathcal{B}$.

The group assignment problem, which optimally assigns each group to a shelf, is stated as follows:

$$\textbf{GAP} : \textit{Maximize} \sum_{i \in \mathcal{B}} \sum_{g \in \mathcal{J}} \mu_{gi} \theta_{gi} \tag{6a}$$

$$\text{subject to} \sum_{i \in \mathcal{B}} \theta_{gi} = 1, \quad \forall g \in \mathcal{J} \tag{6b}$$

$$\sum_{g \in \mathcal{J}} \theta_{gi} = 1, \quad \forall i \in \mathcal{B} \tag{6c}$$

$$\theta_{gi} - \theta_{g'i'} = 0, \quad \forall (g, g') \in \mathcal{M}, a \in \mathcal{A}, i, i' \in \mathcal{N}_a \tag{6d}$$

$$\theta \text{ binary.} \tag{6e}$$

The objective function (6a) maximizes a measure of the overall store profitability. Constraint (6b) guarantees that every group is assigned to one shelf, while Constraint (6c) guarantees that every shelf accommodates one group. Constraint (6d) requires group pairs having high affinity to be assigned to the sames aisle. Finally, Constraint (6e) enforces logical binary restrictions on the decision variables.

References

[1] Amrouche N, Zaccour G. Shelf-space allocation of national and private brands. Eur J Oper Res 2007;180:648–63.
[2] Anderson EE, Amato HN. A mathematical model for simultaneously determining the optimal brand-collection and display-area allocation. Oper Res 1974;22(1):13–21.
[3] Bianchi-Aguiar T, Silva E, Guimaraes L, Carravilla MA, Oliveira JF, Amaral JG, et al. Using analytics to enhance a food retailer's shelf-space management. Interfaces 2016;46(5):424–44.
[4] Borin N, Farris PW, Freeland JR. A model for determining retail product category assortment and shelf space allocation. Decis Sci 1994;25(3):359–84.
[5] Botsali AR. Retail facility layout design. Texas A&M University; 2007. Ph.d. dissertation.
[6] Cachon GP, Kök AG. Category management and coordination in retail assortment planning in the presence of basket shopping consumers. Manage Sci 2007;53(6):934–51.
[7] Cox KK. The effect of shelf space upon sales of branded products. J Market Res 1970;7(1):55–8.
[8] Corstjens M, Doyle P. A model for optimizing retail space allocations. Manage Sci 1981;27(7):822–33.
[9] Curhan RC. The relationship between shelf space and unit sales in supermarkets. J Market Res 1972;9(4):406–12.
[10] Daruka GP, Palekar US. Application of column generation techniques to retail assortment planning. In: Karlof JK, editor. Integer programming theory and practice. Taylor & Francis Group; 2006. p. 221–51.
[11] Desmet P, Renaudin V. Estimation of product category sales responsiveness to allocated shelf space. Int J Res Market 1998;15(5):443–57.
[12] Dreze X, Hoch SJ, Purk ME. Shelf management and space elasticity. J Retail 1994;70(4):301–26.
[13] Flamand T, Ghoniem A, Maddah B. Promoting impulse buying by allocating retail shelf space to grouped products. J Oper Res Soc 2016;67(7):953–69.
[14] Frontoni E, Marinelli F, Rosetti R, Zingaretti P. Shelf space re-allocation for out of stock reduction. Comput Ind Eng 2017;106:32–40.
[15] Gajjar HK, Adil GK. A piecewise linearization for retail shelf space allocation problem and a local search heuristic. Ann Oper Res 2010;179(1):149–67.
[16] Ghoniem A, Maddah B. Integrated assortment, pricing, and inventory decisions: optimization and insights. Omega 2015;55:38–52.
[17] Ghoniem A, Maddah B, Ibrahim A. Optimizing assortment and pricing of multiple retail categories with cross-selling. J Global Optim 2016;66(2):291–309.
[18] Ghoniem A, Flamand T, Haouari M. Exact solution methods for a generalized assignment problem with location/allocation considerations. INFORMS J Comput 2016;28(3):589–602.
[19] Ghoniem A, Flamand T, Haouari M. Optimization-based very large-scale neighborhood search for generalized assignment problems with location/allocation considerations. INFORMS J Comput 2016;28(3):575–88.
[20] Hansen P, Heinsbroek H. Product selection and space allocation in supermarkets. Eur J Oper Res 1979;3(6):474–84.
[21] Hansen JM, Raut S, Swami S. Retail shelf allocation: a comparative analysis of heuristic and meta-heuristic approaches. J Retail 2010;86(1):94–105.
[22] Hariga MA, Al-Ahmari A, Mohamed AA. A joint optimisation model for inventory replenishment, product assortment, shelf space and display area allocation decisions. Eur J Oper Res 2007;181(1):239–51.
[23] Hübner A, Kuhn H. Retail category management: state-of-the-art review of quantitative research and software applications in assortment and shelf space management. Omega 2012;40(2):199–209.
[24] Hübner A, Kuhn H, Kühn S. An efficient algorithm for capacitated assortment planning with stochastic demand and substitution. Eur J Oper Res 2016;250:505–20.
[25] Hübner A, Schaal K. An integrated assortment and shelf-space optimization model with demand substitution and space-elasticity effects. Eur J Oper Res 2017;261:302–16.
[26] Hübner A, Schaal K. A shelf-space optimization model when demand is stochastic and space-elastic. Omega 2017;68:139–54.
[27] Hübner A, Schaal K. Effect of replenishment and backroom on retail shelf-space planning. Bus Res 2017;10:123–56.
[28] Inman JJ, Winer RS, Ferraro R. The interplay among category characteristics, customer characteristics, and customer activities on in-store decision making. Am Market Assoc 2009;73(5):19–29.
[29] Irion J, Lu JC, Al-Khayyal FA. Tsao y-c. a piecewise linearization framework for retail shelf space management models. Eur J Oper Res 2012;222(1):122–36.
[30] Kacen JJ, Hess JD, Walker D. Spontaneous selection: the influence of product and retailing factors on consumer impulse purchases. J Retail Consum Serv 2012;19(6):578–88.
[31] Kahn BE, Lehmann DR. Modeling choice among assortments. J Retail 1991;67(3):274–99.
[32] Karampatsa M, Grigoroudis E, Matsatsinis NF. Retail category management: a review on assortment and shelf-space planning models. In: Grigoroudis E, Doumpos M, editors. Operational research in business and economics. Springer Proceedings in Business and Economics; 2017. p. 35–67.
[33] Katsifou A, Seifert RW, Tancrez JS. Joint product assortment, inventory and price optimization to attract loyal and non-loyal customers. Omega 2014;46:36–50.
[34] Ke W, Van Ryzin G. Optimization of product placement in a retail environment. Columbia University; 2011. Graduate SSchool of Business. Manuscript.
[35] Kollat DT, Willett RP. Customer impulse purchasing behaviour. J Market Res 1967;4:21–31.
[36] Kök AG, Fisher ML, Vaidyanathan R. Assortment planning: review of literature and industry practice. Retail supply chain management. Agrawal N, Smith SA, editors. Kluwer Publishers; 2008.
[37] Kress D, Boysen N, Pesch E. Which items should be stored together? a basic partition problem to assign storage space in group-based storage systems. IISE Trans 2017;49(1):13–30.
[38] Lam SY, Vandenbosch M, Hulland J, Pearce M. Evaluating promotions in shopping environments: decomposing sales response into attraction, conversion, and spending effects. Market Sci 2001;20(2):194–215.
[39] Maddah B, Bish EK, Munreo B. Pricing, variety, and inventory decisions for categories of substitutable items. Planning production and inventories in the extended enterprise: a state of the art handbook. Kemph K, Keskinocak P, Uzsoy R, editors. Dordrecht: Kluwer Academic Publishers; 2011.
[40] Maddah B, Bish EK, Tarhini H. Newsvendor pricing and assortment under poisson decomposition. IIE Trans 2014;46:567–84.
[41] Murray CC, Talukdar D, Gosavi A. Joint optimization of product price, display orientation and shelf-space allocation in retail category management. J Retail 2010;86(2):125–36.
[42] Perdikaki O, Kesavan S, Swaminathan JM. Effect of traffic on sales and conversion rates of retail stores. Manufact Serv Oper Manage 2011;14(1):145–62.
[43] Russell RA, Urban TL. The location and allocation of products and product families on retail shelves. Ann Oper Res 2010;179:131–47.
[44] Smith-Miles K, Lopes L. Measuring instance difficulty for combinatorial optimization problems. Comput Oper Res 2012;39(5):875–89.
[45] Stassen RE, Mittelstaedt JD, Mittelstaedt RA. Assortment overlap: its effect on shopping patterns in a retail market when the distributions of prices and goods are known. J Retail 1999;75(3):371–86.
[46] Yang MH, Chen WC. A study on shelf space allocation and management. Int J Prod Econ 1999;60–61:309–17.
[47] Yang MH. An efficient algorithm to allocate shelf space. Eur J Oper Res 2001;131:107–18.
[48] Zufryden FS. A dynamic programming approach for product selection and supermarket shelf-space allocation. J Oper Res Soc 1986;37(4):413–22.