An optimization-based approach to integrated assortment planning and shelf-space allocation for retail stores including endogenous traffic drivers

David van Ommen

Erasmus University Rotterdam
Erasmus School of Economics
Rotterdam, the Netherlands
483808do@student.eur.nl
Supervisor Kuryatnikova, O.
Second assessor Maasakkers, F.J.L. van
4-7-2021

Abstract. In this thesis an existing integrated assortment planning and shelf-space allocation optimization-based heuristic is used for maximizing the impulse purchase profit for a retail store. We replicate the model and analyze the obtained results. An extra variable and constraints are added to the APSA model to include endogenous traffic drivers and this new model is called Model APSA^{+}. These endogenous traffic drivers give shelves a visibility boost if a fast-mover is placed on it, since more customers will walk past this shelf. The new model is computationally feasible and leads to different shelf space allocations where fast-movers are spread out more across the store, maximizing overall shelf visibility.

Table of Contents

1 Introduction

One of the most important and largest sectors worldwide is the retail industry. Because of the competitive nature of this sector, companies in it need to maximize their profits by all means possible. A concept that these companies try to exploit is *impulse* buying. This consists of purchases that consumers did not intend to do before going to the store and these make up a large part of the total purchases. The authors of [Kollat and Willett, 1967] even found that 50.5% of all supermarket purchases are unplanned. Because of this interest in maximizing the benefits of impulse buying the field of retail analytics has grown and new models are being introduced to help optimize the managerial choices of a company. While multiple factors influence impulse buying, this paper will focus on shelf-space allocation. This is important because it determines which products receive the best places in the store and therefore have a higher potential for being bought impulsively. The authors of [Inman et al., 2009] describe in their research that time spent in the store and the number of aisles visited have a significant impact on impulse purchases and these are the things that a shelf-space allocation can influence. The model in this thesis is based on an existing model by [Flamand et al., 2018] but adds an important variable that changes the way shelf-space allocations are computed.

2 Literature Review

For decades now, research has been done on assortment planning and shelf space allocation. For example, [Anderson, 1979] stated that shelf space allocation should not matter if consumers are completely brand loyal and all the products are always available. Later, [Carpenter and Lehmann, 1985] argued that consumer are willing to compromise and switch to other products because shelf displays influenced their behaviour. In the paper of [Yang, 2001] an approach is proposed that allocates products to shelves, one product at a time, according to sales profit per allocated space. With that approach, [Yang, 2001] showed that proficient shelf space management can lead to improved allocations and increased sales. Furthermore, the authors of [Chen et al., 2006] examined the effect of spatial relationships of products on unit sales. Placing certain products together, or not, can influence the profit of the store. Lots of models have been made for product assortment and shelf-space allocation, but most of them did not focus on impulse buying.

In the field of impulse purchases, substantial research has been done already. For example, [Kacen et al., 2012] show the importance of the effect of product characteristics on impulse buying rather than retailing factors. They conclude that products with a hedonic nature have higher impulse purchasing potential than non-hedonic products.

The writers of [Kök et al., 2008] concluded that most retailers base their choices on art and judgement while they could benefit from big data and mathematical models. The authors of [Ghoniem et al., 2014] propose such a model in the form of a 0-1 mixed-integer program (MIP) with an objective function that maximizes the impulse buying profit per customer. Furthermore, there are constraints regarding shelf-space capacity, for example, that products have a minimum and maximum shelf-space and that there are no products without shelf-space. There is also made sure that products with high affinity are placed close to each other. This model is an SSAP (Shelf Space Allocation Problem) and it has the complication that it is computationally difficult to perform in practice [Flamand et al., 2016].

This is why [Flamand et al., 2016] proposed a group-based model that first uses this SSAP to optimize the most efficient allocation of a specific product group for each shelf in the store and later uses this to find the best allocation for the whole store. This model is called the group assignment problem (GAP). Two years later, the same authors [Flamand et al., 2018] came up with a new model which is Model APSA that puts more focus on allocations that are convenient for the customers. It also adds to the existing model the optimization of the assortment, a topic that still has a lot of research potential. Furthermore, [Flamand et al., 2018] introduced Model APSA itself, as well as an optimization-based heuristic that outperforms the stand-alone Model APSA. Sect. 3 will further explain how this heuristic works and why it performs better than other models. However, even this improved model is not perfect as numerous assumptions are made.

For example, it assumed that *traffic densities* are exogenous and not endogenous. Traffic densities are values that indicate the attractiveness of shelf segments by their location in the store. These are fixed in the heuristic of [Flamand et al., 2018] but this means *traffic drivers* are ignored. Traffic drivers are products that are often bought such as bread and milk. For the rest of this paper, these will be referenced as *fast-movers*. By strategically placing these products, the average customer will spend more time in the store and see more products hence increasing the impulse buying potential. Another way to exploit traffic drivers is placing a fast-mover next to an impulse product to increase its sales. There is obvious potential in endogenous traffic drivers, yet there is almost no literature about this topic. The existing research of [Flamand et al., 2018] also does not incorporate this factor as each shelf has fixed visibility, and traffic drivers have no influence on this.

The authors of [Flamand et al., 2016] even concluded that there does not exist a model that includes endogenous traffic drivers. They suggested as an extension to their GAP model, to incorporate endogenous synergies such as cross-selling or traffic drivers. This thesis will add this factor by giving shelves with a traffic driver on them a higher visibility level. The new model with these endogenous traffic drivers is Model APSA$^+$ and it also uses the heuristic approach of [Flamand et al., 2018]. The research question of this paper will then be:

How will the performance of the state-of-the-art mixed-integer programming heuristic for optimizing shelf-space allocation in a retail store change by adding endogenous traffic drivers?

Since the field of shelf-space optimization utilizing mixed-integer problems is relatively unexplored, there is literature missing. Adding these traffic drivers to the model is something that has not been done before and therefore it is very interesting to see how the results of the proposed state-of-the-art model by [Flamand et al., 2018] will change with this extension.

3 Methodology

The model that is used in this thesis is the heuristic approach of [Flamand et al., 2018]. As explained in Sect. 2 it consists of two mixed-integer problems and the following sets are used for both:

3.1 APSA

- $B \equiv 1, ..., m$: Set of shelves indexed by i.
- K_i : Set of consecutive shelf segments along shelf $i (i \in B)$, indexed by k.
- $K \equiv \cup_{i \in B} K_i$: Set of all shelf segments.
- $N = 1, ..., n$: Set of all product categories indexed by j.
- $F \subset N$: Set of fast-movers, i.e., the 20% of all product categories that contribute almost 80% of the expected sales. These products have high sales but low profit.
- $I \equiv N \backslash F$: Set of slow-movers that have high-profit margins, low sales, and strong impulse buying potential.
- L : Set of product pairs $(j, j') \in N^2$ that have *allocation dissaffinity*. They should not be placed on the same shelf.
- H_1 : Set of product pairs $(j, j') \in P^2$ that have *symmetric assortment affinity*. These pairs can only be selected together in the same shelf or neither of these products can be selected.
- H_2 : Set of product pairs $(j, j') \in P^2$ that have *asymmetric assortment affinity*. If product j is selected, product j' must be selected and both be located on the same shelf. However, product j' can be selected without product j.
- H_3 : Set of product pairs $(j, j') \in P^2$ that have *allocation affinity*. If both products are selected they should be placed on the same shelf.

The last four sets are not used for the main research but they are included for further research in Tables 4 and 5. Furthermore, these are the input parameters used:

- ρ_j : Per-unit profit for product category $j, \forall j \in N$. This parameter is different in this thesis since [Flamand et al., 2018] uses profit margins.
- v_j : Expected demand for product category j.

- $\gamma_p \in [0,1]$: Impulse purchase rate for each product category $j, \forall j \in N$.
- $f_k \in [0,1]$: Customer traffic density for segment $k, \forall k \in K$. The traffic density is based on the prominent walking routes in a store. For example, an aisle that is close to the counters or entrance will have a more prominent spot than an aisle in the back of the store.
- $\Phi_j \equiv \gamma_j \times \rho_j \times v_j$: Largest profit possible for product j that would be realized in an attractive spot that has a high traffic density of 1.
- l_j, u_j : Minimum and maximum shelf-space requirement for product category $j, \forall j \in N$. These are set by the retailer to reduce the risk of stock-outs.
- ψ_j : Smallest product facing length for each product category $j, \forall j \in N$.
- α_i/β_i : Smallest/largest index of the segment belonging to shelf $i, i \in B$.
- c_k : Maximum shelf-space available for segment $k, \forall k \in K$.
- c^{max} : Maximum shelf capacity among all shelf segments, $c^{max} = max_{k \in K} c_k$.
- $C_i \equiv \sum_{k \in K_i} c_k$: Capacity of shelf $i, i \in B$.

All the decision variables of the model are:

- $x_{ij} \in \{0,1\}$: $x_{ij} = 1$, if and only if product category j is assigned to shelf $i, \forall j \in N, i \in B$.
- $y_{kj} \in \{0,1\}$: $y_{kj} = 1$, if and only if product category j is assigned to segment $k, \forall j \in N, k \in K$
- s_{kj} : Allocated shelf-space for product category j along segment $k, \forall j \in N, k \in K$.
- $q_{kj} \in \{0,1\}$: $q_{kj} = 1$ if and only if product j is assigned to shelf segments k and $k+1, \forall k \in K \setminus \{B_i : i \in B\}, j \in N$.

For the rest of this heuristic approach Model APSA [Flamand et al., 2018] is used multiple times, which takes as input a set of shelves and a set of products. The constraints of Model APSA are the following:

$$\text{APSA}(N,B) : \text{Maximize} \quad \sum_{k \in K_i} \sum_{j \in N} \Phi_j \frac{f_k s_{kj}}{c_k} \tag{3.1}$$

$$\text{subject to} \quad \sum_{i \in B} x_{ij} \leq 1, \quad \forall j \in N \tag{3.2}$$

$$\sum_{j \in N} s_{kj} \leq c_k, \quad \forall k \in K \tag{3.3}$$

$$l_j \sum_{i \in B} x_{ij} \leq \sum_{k \in K} s_{kj} \leq u_j \sum_{i \in B} x_{ij}, \quad \forall j \in N \tag{3.4}$$

$$\psi_j y_{kj} \leq s_{kj} \leq min\{c_k, u_j\} y_{kj}, \quad \forall j \in N, k \in K \tag{3.5}$$

$$s_{k_2,j} \geq c_{k_2}(y_{k_1,j} + y_{k_3,j} - 1), \quad \forall j \in N, k_1, k_2, k_3 \in K_i | k_1 < k_2 < k_3 \tag{3.6}$$

$$y_{kj} \leq x_{ij}, \quad \forall i \in B, j \in N, k \in K_i \tag{3.7}$$

$$x_{ij} \leq \sum_{k \in K_i} y_{kj}, \quad \forall i \in B, j \in N \tag{3.8}$$

$$q_{kj} \geq y_{kj} + y_{k+1,j} - 1, \quad \forall i \in B, j \in N, k \in K_i \backslash \{B_i\} \tag{3.9}$$

$$\sum_{j \in N} q_{kj} \leq 1, \quad \forall i \in B, k \in K_i \backslash \{B_i\} \tag{3.10}$$

$$x_{ij} + x_{ij'} \leq 1, \quad \forall (j, j') \in L, i \in B \tag{3.11}$$

$$x_{ij} - x_{ij'} = 0, \quad \forall (j, j') \in H_1, i \in B \tag{3.12}$$

$$x_{ij} \leq x_{ij'}, \quad \forall (j, j') \in H_2, i \in B \tag{3.13}$$

$$x_{ij} - x_{ij'} \leq 1 - z_{jj'}, \quad \forall (j, j') \in H_3, i \in B \tag{3.14}$$

$$x_{ij} - x_{ij'} \geq -1 + z_{jj'}, \quad \forall (j, j') \in H_3, i \in B \tag{3.15}$$

$$z_{jj'} \leq \sum_{i \in B} x_{ij}, \quad \forall j, j' \in H_3 \tag{3.16}$$

$$z_{jj'} \leq \sum_{i \in B} x_{ij'}, \quad \forall j, j' \in H_3 \tag{3.17}$$

$$z_{jj'} \geq \sum_{i \in B} x_{ij} + \sum_{i \in B} x_{ij'} - 1, \quad \forall j, j' \in H_3 \tag{3.18}$$

$$x, y, z \text{ binary}, s, q \geq 0. \tag{3.19}$$

The objective function 3.1 iterates over all shelf segments and products and maximizes the store profit weighted by impulse purchase potential. When a product j is selected on shelf segment k the product parameters that contribute to this value are the per-unit profit ρ_j, the impulse potential γ_j, and demand v_j. The parameters corresponding to the shelf segment that contribute are the traffic density f_k and the capacity c_k where the latter harms the objective value.

Constraint 3.2 makes sure that each product can not be assigned to more than one shelf. Constraint 3.3 restricts the shelf-space allocated to a shelf segment to its capacity. Constraint 3.4 ensures that the shelf-space allocated to a product, if selected, is between its minimum and maximum requirements. Constraint 3.5 creates a conditional lower and upper bound for the amount of space a product can be given along any shelf. Constraint 3.6 makes sure that if a product is allocated to a pair of shelf segments, it will fill the intermediary shelf segment if there is one. Constraint 3.7 ensures that a product can only be allocated to a shelf segment if it is allocated to that corresponding shelf. Constraint 3.8 is for making sure that if a product is allocated to a shelf, it has to be allocated to one of its shelf segments. Constraints 3.9 and 3.10 ensure that not more than one product category can be placed along with contiguous shelf segments. Constraint

3.11 prevents that product pairs of the set L are placed on the same shelf. This is because of their allocation dissaffinity. Similarly, constraint 3.12 makes sure that product pairs in H_1 are assigned to the same shelf, or not selected at all. This needs to be done since these products have symmetric assortment affinity. Constraint 3.13 makes sure for the product pairs in H_2 that if product j is selected, product j' should be selected too and allocated to the same shelf. These products have asymmetric assortment affinity. Constraints 3.14 and 3.15 deal with the H_3 set. This means that products j and j' should be placed on the same shelf if they are both selected because of allocation affinity. Constraints 3.16-3.18 are linearization constraints that are used to avoid quadratic constraints. Lastly, constraint 3.19 confirms that the binary variables are binary and the non-negative variables are non-negative.

Model APSA can be solved by optimization software CPLEX fully, but [Flamand et al., 2018] suggest to solve it using their heuristic, which will be done in this thesis. Model APSA is the main ingredient for tis heuristic, since it is used in both algorithms that form the heuristic. Firstly, Algorithm 1 creates a feasible product allocation and then Algorithm 2 further optimizes this solution. To create the initial solution Algorithm 1 firstly ranks all the shelves according to their attractiveness. This is done by using the following measure: $\lambda_i \equiv \frac{\sum_{k \in K_i} f_k c_k}{\sum_{k \in K_i} c_k}, \forall i \in B$. Let $\sigma = (\sigma_1, ..., \sigma_m)$ be the shelves ordered based on relative attractiveness. Algorithm 1 then uses Model APSA for each shelf separately in the order of σ. The first shelf has all the products available, but if a product is allocated to a shelf, it will not be available for less attractive shelves. This process continues until all the shelves have products allocated to them. A pseudocode of Algorithm 1 is given below.

Algorithm 1 Initialization procedure. [Flamand et al., 2018]

1: Input σ. Set $i = 1, S = \emptyset$ (S refers to the set of selected product categories)
2: Set $i^* \leftarrow \sigma_i$.
3: Solve the SSP(i^*) model which results in an optimal allocation for shelf i^*, regarding the product categories in $N \backslash S$. Let N_{i^*} be the set of products that are allocated to σ_i.
4: Set $S \leftarrow S \cup N_{i^*}$
5: **for all** $(j_1, j_2) \in H_3$ **do**
6:     **if** $(j_1 \in N_{i^*})$ **then**
7:         Set $S \leftarrow S \cup \{j_2\}$
8:     **end if**
9:     **if** $(j_2 \in N_{i^*})$ **then**
10:         Set $S \leftarrow S \cup \{j_1\}$
11:     **end if**
12: **end for**
13: **if** $(i = m)$ or $(S = N)$ **then**
14:     Stop.
15: **end if**
16: Set $i \leftarrow i + 1$, Go to Step 3.

Note that this algorithm deals with the H_3 by using if-statements. Therefore, the constraints 3.14-3.18 are removed from APSA when Algorithm 1 is used. The solution that is now created is not optimal but it is feasible. The second part of the heuristic uses this feasible base and further optimizes it. First, an upper bound is created by letting CPLEX solve the continuous relaxation of APSA. Then, the Model APSA is invoked over subsets of shelves. Algorithm 2 iteratively picks out τ shelves that have different visibility levels and uses Model APSA to re-optimize the allocation on these shelves. All products already on these shelves and products that have not been chosen in Algorithm 1 are available for re-optimization. This process continues until a stopping condition is met. The same sets and input variables as before are used.

Algorithm 2 MIP-based re-optimization procedure.

Create upper bound by solving the continuous relaxation of the problem.
Use algorithm 1 to find a feasible solution with reward $r^* = \sum_{i \in B} r_i$
incumbent: $\leftarrow r^*$
$\hat{k} \leftarrow (x, y, x)$
$\tau \leftarrow 3$
repeat
    while (|tempset| $\geq |B|(\text{mod})\tau$) **do**
        Let $\Delta = (\Delta_1, ..., \Delta_m)$ be the shelves in a non-increasing order based on their current objective value contribution.
        Ω =round($|\Delta|/\tau$)
        for all $k = 1, ..., \tau$ **do**
            Randomly select 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 new subset of shelves by considering all products
        incumbent $\leftarrow r^*_{new}$
        $\hat{k}^{new} \leftarrow (x^{new}, y^{new}, s^{new})$
        $\hat{k} \leftarrow \hat{k}^{new}$
    end while
until stopping condition is met

The stopping condition can be met in three possible ways. Mainly the model terminates when the found solution is closer than ϵ% from the continuous relaxation of the problem. This is called the Optimality Gap and it is calculated using the best incumbent solution and the upper bound (BUB). The upper bound or continuous relaxation is calculated by CPLEX by running APSA and relaxing the binary variables to the interval [0, 1]. The % Optimality Gap is then: $\frac{BUB - Incumbent}{BUB} * 100$. This has to be smaller than ϵ to terminate. Also when the algorithm can not find an improvement in objective value 10 times in a row it will terminate and finally, there is a time limit of 1 CPU hour. In this thesis, an ϵ of 5 is used. This is different than [Flamand et al., 2018] who used 0.5 as ϵ but as for this thesis less computational power and time are available, an ϵ of 5 is more appropriate. Furthermore, in this algorithm, a τ of 3 is used whereas in [Flamand et al., 2018] the writers use a τ of 4. As can be seen in Table 2 in Sect. 5 the computation times are significantly less for a τ of 3 and since it has no impact on the Optimality Gap of the model this is a better fit for this thesis. A τ of 2 also leads to reduced running times but this decreases the accuracy of the model.

3.2 APSA$^+$

We have now seen the complete heuristic approach of [Flamand et al., 2018] but it does not account for endogenous traffic drivers. To incorporate this, an extra variable, as well as a new constraint, needs to be introduced. This new model is Model APSA$^+$. Some changes have to be made and the extra variable that will be added to the model is the following:

- $\chi_i \in \{0, 1\} : \chi_i = 1$, if and only if shelf i contains a product $j \in F, \forall i \in B$.

χ_i is binary variable that is 1 if shelf i contains a fast-mover product and 0 otherwise. It is important since it is used to help increase the visibility of a shelf when a fast-mover is on that shelf.

The following two constraints are added to give the variables the desired properties:

$$\chi_i \leq \sum_{k \in K_i} \sum_{j \in F} y_{kj} \quad \forall i \in B, k \in K \tag{3.20}$$

$$\chi \text{ binary}, \mu \geq 0. \tag{3.21}$$

Constraint 3.20 makes sure that χ_i is equal to 0 if no fast-movers are on shelf i and equal to 1 if there is a fast-mover on that shelf. This is the case since χ_i is binary and the objective function benefits of a χ_i as high as possible.

Namely, the adjusted objective function looks like this:

$$\text{APSA}^+(N, B) : \text{Maximize} \quad \sum_{i \in B} \sum_{k \in K_i} \sum_{j \in N} \Phi_j \frac{f_k s_{kj}}{c_k} \frac{10 + \frac{\chi_i}{f_k^\delta}}{10} \tag{3.22}$$

In this objective a bonus is given to shelves that have a fast-mover on it. The f_k is included to make the model more realistic because without the f_k the bonus visibility gained from including a fast-mover would be higher for shelves with already high visibility. We can see that when we calculate it like this:

$$f_k * \frac{10 + \frac{\chi_i}{f_k^\delta}}{10} = f_k + \frac{\chi_i}{10 f_k^{\delta - 1}} \tag{3.23}$$

In this thesis a δ of 1 is used which means that all shelves will gain a 10% bonus if a fast-mover is allocated on it. For the bonus of 10%, we are interested in 10% of χ_i, and not of f_k. In this manner, the obtained bonus has a relatively stronger effect on shelves with low visibility, which seems more realistic. In the sensitivity analysis of δ in Table 6 values of 2 and 3 are used. This means that the bonus decreases with the current visibility of a shelf. In this way, shelves with low visibility gain a larger bonus than shelves with already high visibility levels.

As can be seen in this objective function two variables are included, namely s_{kj} and χ_i. Although this looks like a quadratic objective function, this can be avoided by doing the following: Let $s_{kj} * \chi_i$ be replaced with ν_{kij} and let M

be a large number where $M > s_{kj}$, for all feasible values of s_{kj}. If we then add the following constraints to the model the problem will still be linear and it can easily be solved. This procedure is called linearization and since it is automatically done by CPLEX, it is not in the formulation.

$$0 \leq \nu_{kij} \leq s_{kj} \quad \forall k \in K, i \in B, j \in N \tag{3.24}$$

$$\nu_{kij} \leq M * \chi_i \quad \forall k \in K, i \in B, j \in N \tag{3.25}$$

With this new objective function, the objective value of the problem becomes incomparable to the original model. This is not a problem for this thesis because it is the shelf-space allocation that matters, not the objective value.

4 Data

To be able to draw any conclusions on the research question, relevant data should be used. In this thesis we look at two types of simulations: one where the store is simulated but the products are real and one where the store and products are both simulated. Both are used for the evaluation.

4.1 Real products

In [Flamand et al., 2018] a retailer is considered who owns multiple stores and is about to open a new one. This thesis will use the same products. Product names and their demands based on historical data are given, as well as the fast-mover set. These fast-movers are Bread(9), Canned Vegetables (17), Cigarettes (23), Juice (26), Soda (27), Cheese (32), Milk (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 figure 1 all the product categories are given. For not fast-movers, the impulse purchase rate γ is determined into three categories: low [0, 0.1), medium [0.1, 0.4) and high [0.4, 0.5], where a fast-mover automatically obtains a 1, since their probability of being bought only depends on the location and visibility of its shelf. The only parameter left to complete the largest possible profit $\Phi_j \equiv \gamma_j \times \rho_j \times \upsilon_j$, is the profitability and this is simulated as followed. Since per-unit profits for fast-mover products are generally low, profits for these products are simulated between 0.2 and 0.8 (dollars) for each unit sold. For other products, profits are higher if they are sensitive to impulse buying. In this thesis the per-unit profits are correlated with the impulse rate and calculated like this:

$$\rho_j = \gamma_j * \text{rand}, \quad \forall j \in N, \text{rand} \in [3, 6] \tag{4.1}$$

Here rand is randomly generated. In this manner, the maximum per-unit profit for a product is 3 (dollars), which seems reasonable since profit margins are not high in supermarkets.

Fig. 1. All product categories within their groups, [Flamand et al., 2018]

#	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)

The minimum and maximum requirements for products when selected are simulated in the intervals [1, 3] and [min, 6] respectively since the capacity of one shelf segment is 6 (feet). Lastly, the minimum allocated space ϕ_j is set to 0.1 for all products.

The shelves in the store are similar as they all contain three segments with a capacity of 6 each, adding up to a total of 18. However, the attractiveness for the shelves is done in such a way that every 20% of shelves have the same level of attractiveness t, where $t = 5\%, 25\%, 45\%, 65\%$, and 85%. Furthermore the attractiveness or traffic density for each middle shelf segment $[t, t,\ 0.05]$ is lower than the outside segments $[t + 0.06,\ t + 0.1]$.

The sets L, H_1, H_2, H_3 are only used for Tables 4 and 5 and they are simulated as well. For each of the four sets, there is a 10% chance that a simulated product is added to one of the sets. If a product is selected, it will be randomly appointed a product that was selected as well, to form a pair. If an uneven number of products is chosen the last chosen product will drop out.

4.2 Simulated products

The previous test method that uses a real store works well for seeing how the endogenous traffic drivers work in practice but for testing the performance of the model in terms of running time and accuracy random product instances are used. Shelves are generated in the same way but the product categories are simulated as well now. Product categories will not have a name and every product characteristic is simulated, where a product is added to the fast-movers with a 20% chance. The demand is then randomly generated for fast-movers within a range from [8, 1500] for slow-movers and [2100, 9500] for fast-movers. The rest of the characteristics is done in the same way as before. The random instances are divided into 4 sets that have 30, 40, 50, and 60 shelves in the model with 240, 320, 400, and 480 product categories respectively. Then 2 random instances are included for each of the 4 sets, adding up to 8 different simulated stores.

5 Evaluation of APSA$^+$

To evaluate the impact of the endogenous traffic drivers, the new model is tested for its practicality as well as its computation speed, using the heuristic approach.

All the models are coded in Java with Eclipse as IDE and solved using CPLEX 20.1. See the Appendix for a code description. All computations are done on a Dell XPS 13 7390 workstation with an Intel(R) Core(TM) i3-10110U CPU 2.10GHz processor and 8 GB of RAM.

5.1 Real products

Since for now the L, H_1, H_2, and H_3 sets are not incorporated in the model products can be placed randomly across the shelves since detergent can now be placed alongside bread for example. The interesting aspect to examine is the distribution of fast-mover products since these products might be placed differently because they increase traffic densities for shelves. When running the optimization-heuristic of [Flamand et al., 2018], fast-mover products are placed on the most prominent shelves together, as well as some of the slow-movers with the highest impulse purchase rates such as ice cream and chocolate. When we compare the two heuristics using APSA and APSA, we can see a clear difference between the two. We see that with the endogenous traffic drivers, the fast-movers are distributed more equally across the store. Because of the adjusted objective function we see that for every simulation the model with the endogenous traffic drivers has more shelves with fast-movers than the model without.

5.2 Simulated products

To give a better image of how the implementation of endogenous traffic drivers affects the model, the simulated stores are used. For each of the 8 instances, the model is run 12 times, adding up to a total of 96 test runs. The computation times (CPU(s)), Optimality Gap (Gap(%)), and the number of shelves with fast-movers on them (Shelves) are compared. These three measurement values are used for all results. The computation time is given in CPU seconds and the Optimality Gap in %. The Optimality Gap was calculated using the best incumbent solution and the upper bound (BUB), so: $\frac{BUB - Incumbent}{BUB} * 100$. All results are computed using the heuristic approach with Model APSA$^+$, except in Table 1, where the heuristic approach with Model APSA is used as well.

APSA$^+$. The model is run twice for each instance, once using the heuristic approach with Model APSA$^+$ and once using the original Model APSA [Flamand et al., 2018].

Table 1. Results for APSA$^+$

Set	$(\lvert B\rvert, \lvert N\rvert)$	Inst.	APSA$^+$			APSA		
			CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves
Set 1	(30, 240)	1	127	4.65	25/30	53	4.71	18/30
		2	139	4.72	23/30	32	4.64	14/30
Set 2	(40,320)	1	311	4.76	29/40	129	4.67	23/40
		2	298	4.37	32/40	174	4.94	25/40
Set 3	(50,400)	1	574	4.86	40/50	345	4.94	35/50
		2	1262	4.81	33/50	384	4.95	24/50
Set 4	(60,480)	1	2115	4.82	43/60	1081	4.73	32/60
		2	1859	4.96	41/60	660	4.82	33/60

When we take a look at this table, it is clear that for these random instances the models with endogenous traffic drivers have more shelves with fast-movers than the models without these traffic drivers. Also, computation times increase in Model APSA$^+$.

In the following part of this thesis, a sensitivity analysis is done for the heuristic approach with APSA$^+$, examining the effect of some input parameters in the model.

Sensitivity analysis of τ. The first parameter to analyze is τ which is the neighborhood size used in Algorithm 2. τ values of 2 and 4 are tested while ϵ is fixed at 5.

Table 2. Sensitivity analysis of τ

Set	$(\|B\|, \|N\|)$	Inst.	$\tau = 2,\ \epsilon = 5\%$			$\tau = 3,\ \epsilon = 5\%$			$\tau = 4,\ \epsilon = 5\%$		
			CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves
Set 1	(30, 240)	1	82	4.91	21/30	127	4.65	25/30	334	4.50	23/30
		2	95	4.70	21/30	139	4.72	23/30	547	4.99	19/30
Set 2	(40, 320)	1	261	4.97	30/40	311	4.76	29/40	489	4.81	29/40
		2	212	4.56	33/40	298	4.37	32/40	501	4.45	32/40
Set 3	(50, 400)	1	449	4.95	38/50	574	4.86	40/50	995	4.93	38/50
		2	1182	4.50	32/50	1262	4.81	33/50	1856	4.80	31/50
Set 4	(60, 480)	1	1829	5.39	40/60	2115	4.82	43/60	2289	4.80	38/60
		2	929	4.97	42/60	1859	4.96	41/60	2541	3.97	44/60

As can be seen from Table 2, computation times increase as τ increases. This is contrary to the findings of [Flamand et al., 2018] but as addressed earlier, this is because of the higher ϵ value used in this thesis. We also see that the accuracy of the model increases with a low τ. This will be further elaborated in Sect. 6.

Sensitivity analysis of ϵ. Now a sensitivity analysis of ϵ is done. Values of 2 and 10 are tested while τ is fixed at 3.

Table 3. Sensitivity analysis of ϵ

Set	$(\|B\|, \|N\|)$	Inst.	$\tau = 3,\ \epsilon = 2\%$			$\tau = 3,\ \epsilon = 5\%$			$\tau = 3,\ \epsilon = 10\%$		
			CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves
Set 1	(30, 240)	1	461	1.99	23/30	127	4.65	25/30	60	9.53	17/30
		2	435	1.97	22/30	139	4.72	23/30	79	7.37	15/30
Set 2	(40,320)	1	1118	1.63	30/40	311	4.76	29/40	256	9.16	20/40
		2	1019	2.27	33/40	298	4.37	32/40	143	9.21	25/40
Set 3	(50, 400)	1	2673	1.99	38/50	574	4.86	40/50	394	9.71	31/50
		2	3600	9.36	21/50	1262	4.81	33/50	1009	9.62	21/50
Set 4	(60, 480)	1	3600	2.58	41/60	2115	4.82	43/60	1838	9.92	30/60
		2	3600	2.53	40/60	1859	4.96	41/60	1836	9.83	30/60

For the sensitivity analysis of ϵ we find what we expect. Namely, a higher ϵ leads to lower accuracy and less computation time, where a slightly lower ϵ gives much longer computation times.

Extra sets When the L, H_1, H_2, and H_3 sets are included individually and all together, the outcomes of the heuristic approach are as below in Tables 4 and 5. The results are split into two tables due to space limitations.

Table 4. Extra sets (1/2)

Set	$(\|B\|, \|N\|)$	Inst.	APSA$^+$			L			H_1		
			CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves
Set 1	(30,240)	1	127	4.65	25/30	103	4.54	23/30	103	4.20	25/30
		2	139	4.72	23/30	113	4.95	21/30	111	4.84	23/30
Set 2	(40,320)	1	311	4.76	29/40	307	4.85	31/40	400	4.97	30/40
		2	298	4.37	32/40	307	4.51	31/40	305	4.94	30/40
Set 3	(50,400)	1	574	4.86	40/50	529	4.76	39/50	639	4.97	38/40
		2	1262	4.81	33/50	1254	4.99	33/50	1380	4.50	33/50
Set 4	(60,480)	1	2115	4.82	43/60	3487	4.99	42/60	2715	4.95	43/60
		2	1859	4.96	41/60	3463	4.83	43/60	2795	4.99	44/60

Table 5. Extra sets (2/2)

Set	$(\|B\|, \|N\|)$	Inst.	APSA$^+$			H_2			H_3			L, H_1, H_2 and H_3		
			CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves
Set 1	(30,240)	1	127	4.65	25/30	125	4.65	24/30	114	4.97	24/30	161	4.85	24/30
		2	139	4.72	23/30	147	4.99	22/30	136	4.60	21/30	175	4.99	20/30
Set 2	(40,320)	1	311	4.76	29/40	289	4.94	30/40	358	4.84	29/40	362	4.91	30/40
		2	298	4.37	32/40	327	4.78	34/40	292	4.68	31/40	313	4.78	32/40
Set 3	(50,400)	1	574	4.86	40/50	547	4.91	39/50	606	4.48	41/50	1514	4.99	40/50
		2	1262	4.81	33/50	1386	4.96	30/50	1220	4.87	34/50	1657	4.82	33/50
Set 4	(60,480)	1	2115	4.82	43/60	3028	4.76	45/60	2237	4.97	42/60	3600	9.12	34/60
		2	1859	4.96	41/60	2927	4.99	46/60	1968	4.96	40/60	3600	11.54	29/60

When the simulated sets are included we can find some interesting points. For the smaller problems, adding a set does not seem to differ much in terms of computation times. Sometimes the model with a set is quicker and sometimes not. However, the computation times of the models including all the sets are higher for all cases. For the largest problem instances of set 4, it holds that including a set always leads to higher computation times.

Sensitivity analysis of $\boldsymbol{\delta}$. For the sensitivity analysis of δ values of 2 and 3 are tested. τ and ϵ stay fixed at 3 and 5, respectively.

Table 6. Sensitivity analysis of δ

Set	$(\|B\|, \|N\|)$	Inst.	$\delta = 1$			$\delta = 2$			$\delta = 3$		
			CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves	CPU(s)	Gap(%)	Shelves
Set 1	(30, 240)	1	127	4.65	25/30	245	4.57	28/30	1443	4.93	24/30
		2	139	4.72	23/30	532	4.72	27/30	2026	7.75	23/30
Set 2	(40, 320)	1	311	4.76	29/40	1246	4.97	33/40	3600	17.54	32/40
		2	298	4.37	32/40	909	4.98	38/40	1147	31.83	35/40
Set 3	(50, 400)	1	574	4.86	40/50	2121	4.99	46/50	3600	29.68	42/50
		2	1262	4.81	33/50	2936	4.91	42/50	3600	23.92	32/50
Set 4	(60, 480)	1	2115	4.82	43/60	3600	6.30	49/60	3600	33.97	45/60
		2	1859	4.96	41/60	3600	8.18	51/60	3600	28.90	43/60

In this sensitivity analysis of δ, the objective function of the extension is adjusted. With a δ of 1, all shelves gain a 10% increase in visibility if a fast-mover is allocated to it but with a δ of 2, this bonus increases with a factor of $1/f_k$ which means a bigger bonus for shelves with low original visibility level. With a δ of 3, this effect will increase to the power of 2, namely $1/f_k^2$, increasing the effect that traffic drivers have on the shelf-space allocation. As can be seen from the results a higher δ leads to a much higher computation time. Also, the number of shelves with a fast-mover is higher for a δ of 2 but it seems lower for a δ of 3. For the instances with a high number of shelves and products, the time limit is even reached for the high δ's leading to very high Optimality Gaps.

6 Conclusion APSA$^+$

In this thesis, a new Model APSA$^+$ is proposed that replaces the existing Model APSA in the heuristic approach of [Flamand et al., 2018]. Endogenous traffic drivers are introduced and this affects the shelf-space allocation. When included in the model the logical and apparent outcome is that fast-movers are more evenly spread across the store to increase the overall visibility of all the shelves. However, the model becomes a little less practical since the running time for the model increases. This makes sense since an extra variable is added to the model which makes the problem more complicated.

In the sensitivity analysis of τ and ϵ we find that a lower neighborhood size leads to lower computation times but when the problem becomes big, it has trouble finding a high accuracy solution. A lower τ finds a solution more quickly with a relatively high ϵ but when there is need for a very optimized solution, a larger τ is preferred, even though it takes more time.

When the extra L, H_1, H_2, and H_3 sets are introduced it does not seem to have a big influence unless they are all put together. However, it will still find a solution so this is not an obstacle if you take some time.

For the δ parameter, we obtain some interesting findings. Computation times increase drastically whit a higher δ because of Algorithm 2. In Algorithm 1 a base solution is created where the most profitable products and fast-movers are placed on the best shelves. Then in Algorithm 2 numerous shuffling needs to be done to optimize the overall visibility bonuses. This is why the computation times are so much higher for a higher δ, often reaching the time limit of 3600 CPU seconds. Another remarkable finding is that for a δ of 3, fewer shelves are appointed a fast-mover than with a δ of 2. We can not conclude this for the bigger sets, because of their low accuracy's but in Set 1 this is the case. The interesting thing is that in the model where δ is 3, the shelves with original low visibility all have fast-movers and the original highly visible shelves have not. With such a high δ it is more profitable to include fast-movers on the highly boosted low visibility shelves than to place them on a high visibility shelf with a lower boost. A δ this high seems a bit too much but it is still an interesting result. However, a δ of 1, as well as δ of 2 or somewhere in, between could be the right fit for a specific implementation of Model APSA$^+$.

In the end, it depends on the trade-off between consumer-friendliness and overall shelf visibility. Consumer-friendliness suffers when fast-movers are spread more across the store, since consumers will have to search through the store for these products. With a lower δ of 1, fast-movers are spread somewhat more than in the original Model APSA, but they are still mostly placed on the most visible shelves. This means that consumer-friendliness does not suffer that much from the new shelf-space allocation. With a higher δ of 2, the products are spread more which means a reduction in consumer-friendliness but an increase in overall shelf visibility. When δ is 3, the fast-movers are placed on the least visible shelves and are also not spread that much. This is disadvantageous for consumer-friendliness as well as overall visibility so this does not seem like a strong option to use. However, it is to the user of Model APSA$^+$ how this trade-off is to be made.

References

Anderson, 1979. Anderson, E. E. (1979). An analysis of retail display space: Theory and methods. *The Journal of Business*, 52(1):103–118.

Carpenter and Lehmann, 1985. Carpenter, G. S. and Lehmann, D. R. (1985). A model of marketing mix, brand switching, and competition. *Journal of marketing research*, 22(3):318–329.

Chen et al., 2006. Chen, Y.-L., Chen, J.-M., and Tung, C.-W. (2006). A data mining approach for retail knowledge discovery with consideration of the effect of shelf-space adjacency on sales. *Decision support systems*, 42(3):1503–1520.

Flamand et al., 2018. Flamand, T., Ghoniem, A., Haouari, M., and Maddah, B. (2018). Integrated assortment planning and store-wide shelf space allocation: An optimization-based approach. *Omega*, 81:134–149.

Flamand et al., 2016. Flamand, T., Ghoniem, A., and Maddah, B. (2016). Promoting impulse buying by allocating retail shelf space to grouped product categories. *Journal of the Operational Research Society*, 67(7):953–969.

Ghoniem et al., 2014. Ghoniem, A., Flamand, T., and Maddah, B. (2014). Maximizing impulse buying via store-wide shelf space allocation of high-impulse products and traffic drivers. *Isenberg School of Management, University of Massachusetts Amherst.*

Inman et al., 2009. Inman, J. J., Winer, R. S., and Ferraro, R. (2009). The interplay among category characteristics, customer characteristics, and customer activities on in-store decision making. *Journal of Marketing*, 73(5):19–29.

Kacen et al., 2012. Kacen, J. J., Hess, J. D., and Walker, D. (2012). Spontaneous selection: The influence of product and retailing factors on consumer impulse purchases. *Journal of Retailing and Consumer Services*, 19(6):578–588.

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

Kollat and Willett, 1967. Kollat, D. T. and Willett, R. P. (1967). Customer impulse purchasing behavior. *Journal of marketing research*, 4(1):21–31.

Yang, 2001. Yang, M.-H. (2001). An efficient algorithm to allocate shelf space. *European journal of operational research*, 131(1):107–118.

Appendix

In the attached zip-file the java code is given that is used for this thesis. It consists of six classes of which four are used to create the store environment. These are the classes 'Store', 'Shelf', 'Edge' and 'Product'. The remaining classes are then 'MainModel' and 'MIP'. 'MIP' contains Model APSA that can turn into Model APSA$^+$ with a boolean. It takes as input a set of products and a set of shelves. Here all the CPLEX computations take place. In 'MainModel' the parameters can be tuned to your liking and a store will be created from here. Then Algorithms 1 and 2 are called and the right output is saved and printed.