ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Bachelor Thesis (BSc^2 Econometrics/Economics)

An integrated mixed-integer programming and simulation-based approach to design healthy supermarkets

Name student: Rick Kessels

Student ID number: 445742

Supervisor: O. Kuryatnikova

Second assessor: F.J.L. van Maasakkers

Date final version: July 4th, 2021

Abstract: To investigate the role of supermarkets in encouraging their customers to make healthy purchases, the APSA model as introduced by Flamand et al. (2018) has been extended with additional constraints to simultaneously select more healthy products into the assortment and allocate these products to highly-visible shelf segments across the store. The extended model is compared for both the grid store layout and the herringbone store layout. The resulting shelf allocations of the extended models are used as an input to a simulation to find a reallocation of the shelves that maximizes impulse purchases of healthy product items. The results show a significant increase in the number of healthy impulse purchases in 1,000 random simulations for both store layouts. Store managers who aim to invest in their consumers' health can use the extended model and simulation to achieve a health-optimizing product assortment, shelf allocation and store layout.

Contents

1 Introduction **3**

2 Theoretical Framework **5**

2.1 Product assortment 5

2.2 Shelf allocation 6

2.3 Personal factors 6

3 Methodology **7**

3.1 The APSA-model 7

3.1.1 Sets 8

3.1.2 Parameters 8

3.1.3 Decision variables 9

3.1.4 The model 9

3.1.5 Explanation of the objective function (1a) and constraints (1b)-(1j) 10

3.1.6 Product affinity sets and their associated constraints (1k)-(1s) 11

3.1.7 Additional constraints 12

3.2 Heuristic 13

3.2.1 Algorithm 1 13

3.2.2 Algorithm 2 14

3.3 Additional health-related constraints 16

3.4 Implementation 17

4 Results **17**

4.1 Replication 17

4.1.1 Choice of parameters 17

4.2 Health-related constraints in original simulation 22

4.3 Store simulation 25

4.4 Comparison original APSA versus health-version of APSA 28

5 Discussion **30**

A Appendix **37**

A.1 SSP(i*) model 37

A.2 Results heuristic for varying values of τ . . . 38
A.3 Product information . . . 39
A.4 Store Layouts . . . 40
A.5 Illustration of path-determination . . . 42
A.6 Heatmaps . . . 44
A.7 The role of queues . . . 46
A.8 Optimizing queues . . . 46
A.9 Python code . . . 50
A.9.1 Mixed Integer Programming programs . . . 50
A.9.2 Simulation programs . . . 51

1 Introduction

A store manager needs to make crucial decisions: what products will the store offer, and how should these products be placed? The overarching goal is often to maintain customer satisfaction, while simultaneously maximizing profits. The latter can be achieved by choosing the product assortment and shelf layout in such a way as to stimulate the number of impulse, or unplanned, purchases.

Ultimately, profits are what keep the doors of a store open. However, these unplanned impulse purchases are often of hedonic nature. In the context of food and drinks, this is practically synonymous to unhealthy products (Crawford et al., 2007; Nederkoorn et al., 2009). More than 50% of adults in OECD countries are either overweight or obese (OECD, 2017), making the consumption of products on the unhealthy spectrum an expanding problem in many developed countries. Governments have made extensive effort to try to tackle this problem as to make their citizens adopt a healthier lifestyle and prevent future health care costs. Mass communication, adjustments to food labelling policies, and restrictions on food marketing campaigns, amongst others, have already been heavily investigated and implemented across many countries. For instance, the UK-based Change4Life initiative serves the purpose of promoting a healthy lifestyle by providing the consumer with food facts, recipes and activities to promote a healthy diet and limit sedentary behaviour (Levy, 2013). More recently, several West-European countries, including The Netherlands, have started cooperating to implement the nutritional label NutriScore designed by the French National Public Health Institute (RIVM, 2021). These labels not only help the customer in determining which products are healthy, but also encourage producers to increase their product's nutritional value.

What is currently lacking in the debate is concrete points of action to shape the role for supermarkets within this context as the facilitating medium between a producer's product and a customer's purchase. The conclusion of the "Superlijst Gezondheid 2020", a two-year research study to investigate the role Dutch supermarkets currently take on when it comes to encouraging their customers to make healthier decisions (Questionmark, 2020), was that the current role of Dutch supermarkets to stimulate a healthy lifestyle is marginal and should be expanded substantially over the next years. However, no action has been taken by the Dutch government to implement these recommendations.

Currently, supermarket layouts are heavily based in consumer psychology as to stimulate impulse purchases. For instance, a customer often enters the store by having to traverse through aisles with fresh vegetables and fruit, which makes the customer more likely to start-off their shopping trip with the rewarding feeling of putting a healthy item in their shopping cart. The customer will then move on to the next set of aisles, often containing rather unhealthy products. There, the customer will reward themselves for picking up a healthy item earlier, and will select an unhealthy item in this aisle (RSPH, 2019). Also, it is common practice to put high-demand products at the very back of the store, so customers who need one of these products have to traverse a large number of aisles, and are thus exposed to a large number of products and promotions (RSPH, 2019). From these examples, one sees that understanding how the customer's brain works can be a valuable asset.

Furthermore, it is crucial for a store's profit to place high-profit products in shelf segments that are highly visible. Shelf segments that are at the beginning or the end of an aisle are the segments with the highest visibility (Sorensen, 2003; Larson et al., 2005). A study by the Obesity Health Alliance (OHA) has shown that almost half of the products listed in highly-noticeable shelf segments are food and drink items that contain a high level of sugar (Alliance, 2018). Moreover, it has been shown that 70% of these products were products that UK children tend to consume in relatively high amounts. The OHA raises the question on whether the latter would have also been the case when these products were not presented in such high-visible shelf segments throughout the store.

Not only public health institutions care about this topic. A survey executed by the Royal Society of Public Health of the United Kingdom (RSPH) amongst 2,084 UK supermarket customers has revealed that 42% of the customers think supermarkets should take a more active role in tackling the ever-increasing problem of obesity (2019). Half of the surveyed customers have the impression that supermarkets offerings include more unhealthy products than healthy products. Besides this, 75% of the customers who were surveyed who have small children think that supermarkets should make serious effort to promote healthy food and drinks to children. From this survey, it becomes apparently clear that customers think it is at least partly the store's responsibility to contribute towards encouraging a healthier purchasing pattern.

Product assortment and shelf allocation in the context of promoting a healthy lifestyle has only recently started gaining attention. The report by the RSPH makes several recommendations concerning how to change the layout of a store to encourage customers to buy healthier alternatives (2019). First of all, there should be a larger shelf allocation for healthy food product categories to increase visibility. These categories should also be placed at several locations throughout the store. Furthermore, the visibility of unhealthy food product categories should be kept to a minimum. Finally, it is recommended to put healthy products at the checkout of the store so that if customers have to wait in the queue, they are again exposed to healthy products instead of the usual unhealthy products that lie in these segments.

Using the theory behind shelf-allocation to the benefit of the consumer's health, the research question of the current paper is: *"What are the consequences of employing behavioural insights as such to nudge customers into buying healthy products?"*.

To investigate this research question, several models will be compared. The original APSA (Assortment Planning / Space Allocation) model with horizontal product placement, as introduced by Flamand et al. (2018), will be used as the benchmark model. Here, different store layouts, each with their own shelf segment attraction levels, will be considered. Several additional health-related restrictions will be introduced to the model. The results of this model will be directly compared with the store-layout specific APSA model. 1,000 simulations will be employed to give insight into the most effective shelf permutation to encourage healthy purchasing behaviour by the customer. Here, we can formulate a subquestion: *"How can a store's layout contribute to healthy purchasing behaviour by the consumer?"* The contribution of the current paper is to provide a flexible two-step approach to provide store managers with the tools to create healthy supermarkets.

2 Theoretical Framework

2.1 Product assortment

Product assortment is defined as the number of distinct items within a product category (Levy and Weitz, 1995), and it has found to be a critical factor in the potential success of a store. Namely, a consumer's expectations about the store's product assortment, together with store location and product price, are the three main determinants of a store's profitability (Broniarczyk and Hoyer, 2006). The relevance of product assortment was once more supported by the findings of the meta-analytical study conducted by Pan and Zinkhan (2006).

This definition of product assortment includes both the breadth (the variety in products offered), and the depth (and the number of stock keeping units, "SKUs") of the assortment. Offering a large product assortment can aid a store in satisfying the heterogeneous preferences of their customers (Dhar et al., 2001). Not only can a greater product assortment possibly attract more customers, it can also induce more in-store impulse purchases. Furthermore, customers take into account the total amount of travel costs and effort when making decisions about which store to visit. A larger product assortment will allow them to shop a greater part of their shopping list in the current store, decreasing the probability they have to visit other stores, which in turn reduces the total cost of their shopping trip and increases the level of overall convenience (Pan and Zinkhan, 2006).

2.2 Shelf allocation

The mix of environmental and customer-specific variables influence a customer's food purchases, and consequently, their eating behaviour (Brug, 2008). Changes in an individual's diet can be affected by the adaptation of the environment in which the individual needs to make their dietary decisions (Wilson et al., 2016). This means that being able to follow through with one's health intentions depends significantly on the environment in which this perseverance has to occur. Therefore, it seems effective to alter the temptation-inducing environment to one that rather stimulates healthy alternatives.

One way to achieve this is through product placement. Namely, it has been shown in several studies that in-store product placement strategies play a significant role in the purchasing choices of the customer. Caspi et al. (2017) found a positive relationship between a greater shelf allocation to healthier products and healthier in-store purchases. Furthermore, several observational studies conducted by Nakamura et al. (2014) and Caspi et al. (2017) have provided evidence that strategic positioning of healthy product items in the store significantly increases the amount of healthy products purchased by the customer. Furthermore, it was shown by Cohen et al. (2015) and Nakamura et al. (2014) that the positioning of unhealthy items can also critically influence the amount of unhealthy products purchased by the customer, especially when these are placed in high-attraction shelf segments. Kerr et al. (2012), Cohen et al. (2015) and Nakamura et al. (2014) have all shown that the strategic positioning of healthy and unhealthy food items in highly visible shelf segments throughout the store is positively associated with the sales of these products. This latter finding is crucial to the current research.

2.3 Personal factors

The act of purchasing unhealthy food items has been related to the individual's unconscious way of dealing with a negative psychological headspace, often induced by a low self-esteem or a high stress level. Verplanken et al. (2005) states that the urge of an in-store impulse purchase might be related with the negative emotional state of an individual. Kacen et al. (2012) note that the impulse purchase potential depends partially on the hedonic nature of the product. An upset customer therefore might try to cope with their unhappy feelings by resorting to food items that give them hedonic value, which in most cases are unhealthy.

Besides the psychological aspects, there also seems to be a significant role of education in explaining the different lifestyles pursued by customers. Handbury et al. (2015) found that education levels are a critical determinant when it comes to food preferences. People that enjoyed a higher level of education or have a higher income tend to exhibit healthier food purchasing patterns than people who have a lower education or a low income. The difference is significant: people in the former group tend to consume around 40% closer to the recommendations put in place by the U.S. Department of Agriculture than people in the latter group. Here, the impact of education was found to be much more significant than the impact of income. Similarly, Finnegan Jr et al. (1990) found that people with a tertiary education level tend to be more knowledgeable about the nutritional value of their food, which leads to better-informed choices. The Food and Nutrition Service by the Agriculture Department of the United States found that education about nutrition can lead to the pursuit of healthier diets 2013. The conclusions of these studies underline the importance of educating customers about healthy nutrition.

3 Methodology

3.1 The APSA-model

Flamand et al. (2018) introduce the APSA-model, which offers an integrated, tactical approach to product category level assortment planning and shelf space allocation. On the one hand, this model selects the necessary "fast-moving" product categories with a relatively low profit-margin into the assortment to attract customer traffic. On the other hand, product categories with a high-impulse purchase potential are also selected into the assortment and are specifically allocated to high-attraction shelf segments to stimulate impulse purchases. The model maximizes overall store profitability, which is a function of each product category's associated profit, the attraction level of the shelf segment this product category is located on, and the total amount of space allocated to the product category along this shelf segment.

It is important to note that in the model, Flamand et al. (2018) consider product categories, containing several similar products, instead of individual products (so-called "SKUs"). This means that the trade-off between a high number of product categories and a high number of in-category product alternatives is decided in favour of the latter. However, one could wonder whether a consumer will actually benefit from a wide variety of options in product categories like sugar, salt or water, since the added value of one brand in this category over another within these categories is less clearly defined. Besides this, individual products with a high profit margin and a high impulse purchase potential which would have been selected in a product-specific model, can now be left out due to weaker average parameters of its associated product category (e.g. a lower average profit margin or smaller impulse purchase potential).

The original APSA model, as introduced by Flamand et al. (2018), is explained next.

3.1.1 Sets

Let us first introduce the sets employed in this model:

$\mathcal{N}$ denotes the set of possible product categories $\{1, ..., n\}$, with index j.

$\mathcal{F} \subset \mathcal{N}$ denotes the set of so-called fast-movers, which are products that constitute large portions of the overall revenue of the store. Flamand et al. (2018) use the Pareto principle to determine the product categories in this set.

$\mathcal{I} \equiv \mathcal{N} \backslash \mathcal{F}$ denotes the set of slow-movers, which is the complement of $\mathcal{F}$.

$\mathcal{L}$ denotes the set of pairs of product categories, $(j, j') \in \mathcal{N}$, that have allocation disaffinity, meaning that these product category pairs should not be located in the same shelf if both of them were to be selected in the assortment.

$\mathcal{H}_1$ denotes the set of pairs of product categories, $(j, j') \in \mathcal{N}$, that have symmetric assortment affinity, meaning that these product category pairs should either both be selected into the assortment, or not at all. In case both product categories get selected, they should also be allocated to the same shelf.

$\mathcal{H}_2$ denotes the set of pairs of product categories, $(j, j') \in \mathcal{N}$, that have asymmetric assortment affinity, meaning that if product category j is selected, the category j' also needs to be selected, and they should be allocated to the same shelf. Note that j' can also be selected individually without necessarily including j.

$\mathcal{H}_3$ denotes the set of pairs of product categories, $(j, j') \in \mathcal{N}$, that have allocation affinity, meaning that these product category pairs should be located in the same shelf if both of them were to be selected in the assortment. The product categories can also be individually selected into the assortment.

$\mathcal{B}$ denotes the set of available shelves in the store $\{1, ..., m\}$, with index i.

$\mathcal{K}_i$ denotes the set of available shelf segments along shelf $i \in \mathcal{B}$ with index k.

$\mathcal{K} \equiv \bigcup_{i \in \mathcal{B}} \mathcal{K}_i$ denotes the set of available shelf segments in the entire store.

3.1.2 Parameters

The parameters of the model are specified as follows:

ρ_j denotes the average profit margin for products in product category $j \in \mathcal{N}$.

ν_j denotes the expected demand for products of product category $j \in \mathcal{N}$.

$\gamma_j \in (0, 1]$ denotes a measure for the impulse purchase potential for product category $j \in \mathcal{N}$.

f_k denotes the attractiveness of shelf segment $k \in \mathcal{K}$, where $0 \leq f_k \leq 1$.

$\Phi_j \equiv \gamma_j \rho_j \nu_j$ denotes the largest possible profit for product category $j \in \mathcal{N}$.

ℓ_j denotes the minimum shelf space requirement for product category $j \in \mathcal{N}$, if j is selected in the product assortment.

u_j denotes the maximum shelf space requirement for product category $j \in \mathcal{N}$, if j is selected in the product assortment.

ϕ_j denotes the minimum space requirement in a segment that product category $j \in \mathcal{N}$ is allocated to, if j is selected in the product assortment.

β_i denotes the largest index of a shelf segment that is in shelf $i \in \mathcal{B}$.

c_k denotes the capacity of segment $k \in \mathcal{K}$.

$C_i \equiv \sum_{k \in \mathcal{K}_i} c_k$ denotes the capacity of shelf $i \in \mathcal{B}$.

3.1.3 Decision variables

The model contains five decision variables. x_{ij} is a binary variable with value 1 if product category $j \in \mathcal{N}$ is allocated to shelf $i \in \mathcal{B}$, and 0 otherwise. y_{kj} is a binary variable with value 1 if product category $j \in \mathcal{N}$ is allocated to shelf segment $k \in \mathcal{K}$, and 0 otherwise. q_{kj} is a binary variable with value 1 if product category $j \in \mathcal{N}$ is allocated to both shelf segment k and $k+1$ for $k \in \mathcal{K}_i \backslash \{\beta_i\}, i \in \mathcal{B}$, and 0 otherwise. $z_{jj'}$ is a binary variable with value 1 if product category j and product category j' are both selected into the product assortment, and 0 otherwise. Finally, $s_{kj} \geq 0$ denotes the amount of space in shelf segment $k \in \mathcal{K}$ assigned to product category $j \in \mathcal{N}$.

3.1.4 The model

Then, the model can be formulated as follows:

Maximize

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

s.t.

$$\textstyle\sum_{i\in B} x_{ij} \leq 1 \qquad \forall j \in \mathcal{N} \tag{1b}$$

$$\textstyle\sum_{j\in\mathcal{N}} s_{kj} \leq c_k \qquad \forall k \in \mathcal{K} \tag{1c}$$

$$\textstyle\ell_j \sum_{i\in\mathcal{B}} x_{ij} \leq \sum_{k\in\mathcal{K}} s_{kj} \leq u_j \sum_{i\in\mathcal{B}} x_{ij} \qquad \forall j \in \mathcal{N} \tag{1d}$$

$$\phi_j y_{kj} \leq s_{kj} \leq \min\{c_k, u_j\} y_{kj} \qquad \forall j \in \mathcal{N}, k \in \mathcal{K} \tag{1e}$$

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

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

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

$$q_{kj} \geq y_{kj} + y_{k+1,j} - 1 \qquad \forall i \in \mathcal{B}, j \in \mathcal{N}, k \in \mathcal{K}_i \backslash \{\beta_i\} \tag{1i}$$

$$\textstyle\sum_{j\in\mathcal{N}} q_{kj} \leq 1 \qquad \forall i \in \mathcal{B}, k \in \mathcal{K}_i \backslash \{\beta_i\} \tag{1j}$$

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

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

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

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

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

$$\textstyle z_{jj'} \leq \sum_{i\in\mathcal{B}} x_{ij} \qquad \forall (j, j') \in \mathcal{H}_3 \tag{1p}$$

$$\textstyle z_{jj'} \leq \sum_{i\in\mathcal{B}} x_{ij'} \qquad \forall (j, j') \in \mathcal{H}_3 \tag{1q}$$

$$\textstyle z_{jj'} \geq \sum_{i\in\mathcal{B}} x_{ij} + \sum_{i\in\mathcal{B}} x_{ij'} - 1 \qquad \forall (j, j') \in \mathcal{H}_3 \tag{1r}$$

$$x, y, q, z \in \{0, 1\}, s \geq 0 \tag{1s}$$

3.1.5 Explanation of the objective function (1a) and constraints (1b)-(1j)

The model maximizes the objective function (1a), which represents the total profit potential of all product categories $j \in \mathcal{N}$ being allocated to a shelf segment $k \in \mathcal{K}$ in the store. This objective function can be split into a part representing the fast-mover product categories $j \in \mathcal{F}$ that exhibit high overall sales, and a part representing slow-mover product categories $j \in \mathcal{I}$, characterized by lower sales but instead a higher impulse purchase potential and profit margin. This yields the following: $\sum_{j\in\mathcal{F}} \sum_{k\in\mathcal{K}} \rho_j \nu_j \frac{f_k s_{kj}}{c_k} + \sum_{j\in\mathcal{I}} \sum_{k\in\mathcal{K}} \rho_j \nu_j \gamma_j \frac{f_k s_{kj}}{c_k}$, where the fraction $\frac{f_k s_{kj}}{c_k}$ represents the probability of a product category $j \in \mathcal{N}$ allocated to a shelf segment $k \in \mathcal{K}$ to be observed by a customer. Note that the first part of the split-up objective function does not contain the variable γ_j since this part is concerned with maximizing the number of fast-movers in the product assortment to boost customer traffic to the store. In turn, the slow-moving part maximizes the potential for impulse purchases by allocating product categories $j \in \mathcal{N}$ with a high profit margin to high-attraction shelf segments $k \in \mathcal{K}$.

The total set of constraints ensure that the requirements on the products, shelves and segments are satisfied. Constraint (1b) ensures that a product category $j \in \mathcal{N}$ cannot be allocated to more than one shelf. Constraint (1c) makes sure that the shelf segment capacity c_k is not exceeded for a shelf segment $k \in \mathcal{K}$. The minimum and maximum space requirements for a product category $j \in \mathcal{N}$ that is selected into the assortment is administered by constraint (1d). Constraint (1e) puts lower and upperbounds in place on the space allocated to a product category $j \in \mathcal{N}$ across shelf segment $k \in \mathcal{K}$. Constraint (1f) ensures that if a product category $j \in \mathcal{N}$ is allocated to shelf segments k_1 and k_3, where $k_1 < k_2 < k_3 | k_1, k_2, k_3 \in \mathcal{K}_i$, $i \in \mathcal{B}$, product category j is also allocated to the the intermediate shelf segment k_2. Constraint (1g) manages the fact that a product category $j \in \mathcal{N}$ can only be allocated to a shelf segment $k \in \mathcal{K}_i$ in case it is also allocated to shelf $i \in \mathcal{B}$. Constraint (1h) is closely related to constraint (1g): If a product category $j \in \mathcal{N}$ is allocated to a shelf $i \in \mathcal{B}$, j should be allocated to at least one of shelf i's segments $k \in \mathcal{K}_i$. Constraints (1i) and (1j) together make sure that there can only be one product category $j \in \mathcal{N}$ that runs over two adjacent shelf segments $k \in \mathcal{K}_i$, $i \in \mathcal{B}$.

3.1.6 Product affinity sets and their associated constraints (1k)-(1s)

The model allows for four types of product category relationships to exist between two product categories: (1) symmetric assortment affinity, (2) asymmetric assortment affinity, (3) allocation disaffinity, and (4) allocation affinity. Symmetric assortment affinity means that these product category pairs should either both be selected into the assortment or both should be left out, which is ensured by constraint (1l) in the model. This could be the case when the product category pair has a negative cross elasticity of demand $\epsilon \in (-\infty, 0)$, implying that these product categories are complementary. It is important to emphasize that products from these product categories cannot be properly used without each other. An example of product categories in this affinity set are razors and razor blades. The product category pairs (j, j'), $j \in \mathcal{N}$, $j' \in \mathcal{N}$, that exhibit symmetric assortment affinity are stored in the set $\mathcal{H}_1$ in the APSA-model.

The product category pairs in the set $\mathcal{H}_2$, representing the asymmetric assortment affinity pairs, have a less straight-forward relationship. Let us take $(j, j') \in \mathcal{H}_2$, where $j \in \mathcal{N}$ and $j' \in \mathcal{N}$. Then, product category j' can be selected individually into the assortment, but if category j is selected, category j' also needs to be selected. There also is the case where neither of the product categories are selected. Again, these product categories are very likely to have a negative cross elasticity of demand, but the distinction with the pairs in set $\mathcal{H}_1$ is that product category j' can be used without product category j. An example, as mentioned by Flamand et al. (2018), is cake icing and cake. Cake can be consumed by itself and can therefore enter the assortment as a standalone category, whereas having cake icing in the assortment without having cake in the assortment would first of all be a waste of shelf space due to the fact that the expected demand is (close to) zero, and secondly lead to confused customers looking for a cake to put the icing on. Constraint (1m) in the model takes care of this asymmetric assortment affinity pair relationship.

Not all product category pairs necessarily have any interdependent restrictions when it comes to assortment planning. Yet, they might still have restrictions when it comes to their shelf allocation. The product category pairs in set $\mathcal{L}$, representing allocation disaffinity, cannot be allocated to the same shelf in case they are both selected into the assortment. For instance, a customer could be confused to see household appliances in between fresh vegetables and fruit. This affinity relationship is administered by constraint (1k).

We can also inverse this relationship by introducing set $\mathcal{H}_3$, where the product category pairs, in case both selected, need to be allocated to the same shelf. This would make most sense for products that are closely related to each other, e.g. white wine and red wine. Note that these product categories can also be selected into the assortment individually. Constraints (1n) and (1o) require the product category pairs in $\mathcal{H}_3$ to be allocated to the same shelf $i \in \mathcal{B}$ in case both product categories are selected. Constraints (1p), (1q) and (1r) take care of determining the value of the variable $z_{jj'}$ for product category pairs $(j, j') \in \mathcal{H}_3$ by linearizing the summations $\sum_{i \in \mathcal{B}} x_{ij}$ and $\sum_{i \in \mathcal{B}} x_{ij'}$. Namely, $z_{jj'}$ takes the value 1 in case both summations have value 1, and takes value 0 otherwise.

3.1.7 Additional constraints

In addition to the APSA model presented above, Flamand et al. (2018) introduce an additional set, from which additional constraints can be formed. The set $\mathcal{R} \equiv \{(k_1, k_2, j) \in \mathcal{K} \times \mathcal{K} \times \mathcal{N} : k_1 < k_2 \text{ and } \sum_{h=k_1+1}^{k_2-1} c_h > u_j - 2\phi_j\}$ can be used to limit a product category's ability to be in two separate segments if the maximum space requirement of that product category j, as denoted by u_j, is not large enough to cover all the segments in between k_1 and k_2. We can prevent the assignment of a product category j to such shelf segments k_1 and k_2 by introducing the following inequality to the model:

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

Furthermore, the set of constraints (1f) in the original APSA model can be adjusted into a smaller set of constraints by using the set $\mathcal{R}$ as follows:

$$s_{k_2,j} \geq c_{k_2}(y_{k_1,j} + y_{k_3,j} - 1), \quad \forall j \in \mathcal{N}, i \in \mathcal{B}, k_1, k_2, k_3 \in \mathcal{K}_i | k_1 < k_2 < k_3, (k_1, k_3, j) \notin \mathcal{R}. \quad \text{(1f*)}$$

By replacing constraint (1f) by this set of constraints, we saved the solver from having to go through all combinations of $\mathcal{N}, \mathcal{B}$, and $k_1, k_2, k_3 \in \mathcal{K}_i$, which can result in a considerable gain in efficiency for increasing cardinalities of the aforementioned sets. Whenever the APSA model is applied in the current paper, these two sets of constraints are included in the model.

3.2 Heuristic

Flamand et al. (2018) present a heuristic to solve their APSA model in a significantly faster fashion. This heuristic employs two algorithms: Algorithm 1 and Algorithm 2. Whereas Algorithm 1 focuses on finding an initial feasible solution to the problem, Algorithm 2 tries to iteratively improve the current solution by optimizing a subproblem with a smaller number of products and shelves. For this heuristic, we first need to introduce a single-shelf variant of the original APSA model, the SSP(i^*) model, which can be found in Appendix A.1. Note that the decision variable x_{ij} in the APSA model is replaced by w_j, denoting the binary choice of a product category j being selected for the current shelf under consideration ($i^* \in \mathcal{B}$). The main difference between the two models is the fact that the SSP(i^*) model considers only the shelf segments $\mathcal{K}_{i^*}$ associated to shelf $i^* \in \mathcal{B}$. Furthermore, constraint (1b) as formulated in the original model is redundant, since this model only considers one shelf at a time.

3.2.1 Algorithm 1

This heuristic is based on the categorization of a shelf's relative attractiveness, as measured by $\lambda_i = \frac{\sum_{k \in \mathcal{K}_i} f_k c_k}{\sum_{k \in \mathcal{K}_i} c_k}$. The SSP procedure is applied to the ordered list of shelves σ_i based on λ_i, where the shelves with the highest value of λ_i are considered first. This means that the shelves with the highest attractiveness will get "first pick" when it comes to the product assortment, leaving only a subset of the original product set (namely, $\mathcal{N}\backslash\mathcal{S}$, where $\mathcal{N}$ denotes the total set of products available for the assortment, and $\mathcal{S}$ the selected products in previous iterations of the algorithm) to the next shelf in line. Note that the SSP-model only includes constraints on the affinity sets $\mathcal{L}, \mathcal{H}_1,$ and $\mathcal{H}_2$. Namely, if $(j, j') \in \mathcal{H}_3$ is an allocation affinity pair, and in case either j or j' is selected onto the current shelf in the algorithm, the other product is added to the current shelf as well (see lines 4-6 in Algorithm 1). This means both products are taken out of the set of product categories up for consideration from the next iteration onwards. The SSP(i^*) model is not able to account for the definition of set $\mathcal{H}_3$, since j and j' are allowed to individually be selected for a shelf. However, if these products are both individually selected to different shelves in subsequent iterations, this would lead to an overall infeasible solution, rendering the purpose of the algorithm useless.

Algorithm 1: Construction of an initial feasible solution (Flamand et al., 2018)

```
1 Input an array of the shelves σ, sorted in non-decreasing order according to their relative
    attractiveness. Set i = 1, i* = σ_i, and S = ∅
2 while (i ≠ m + 1) ∧ (S ≠ N) do
3     Apply SSP(i*) with products N\S with resulting product allocation N_{i*} Set S ← S ∪ N_{i*}
4     if (j ∈ S) ∨ (j' ∈ S), ∀(j, j') ∈ H_3 then
5         S ← S ∪ {j, j'}
6     end
      i ← i + 1
7 end
```

3.2.2 Algorithm 2

Algorithm 2 builds on the results of Algorithm 1 by categorizing the shelves into τ groups based on the shelf's current contribution towards the objective value. One shelf from each of these τ groups will randomly be selected (without replacement), and the original APSA model will be used to jointly optimize these τ selected shelves by considering the products currently allocated to these shelves, as well as the products that have not been allocated to any shelf. After optimization, the shelf's contribution towards the overall objective value will be updated. Note here that an individual shelf's objective value can decrease after an iteration, but the summation of the objective values of the shelves considered in this iteration is at least equal to the summation of objective values before the iteration. τ shelves will continue to be selected up until the point that the number of shelves not considered in this iteration of the algorithm is smaller than τ. In case one of the following three stopping criteria holds true, the algorithm is terminated: (1) the relative gap between the current objective value and the upper bound as given by the linear relaxation of the entire APSA model is smaller than ϵ%, (2) the objective value has not improved for w_{max} iterations, where w_{max} is left unspecified by Flamand et al. (2018), or (3) if the algorithm reaches a user-specified time limit. In the current paper, the parameter values listed in Table 1 were chosen. Note that the time limits employed in the current paper are around 2.5 times higher than the ones used by Flamand et al. (2018), to possibly account for higher processing times due to different computational environments.

Algorithm 2: Repeated re-optimization (Flamand et al., 2018)

Input: τ, ϵ

Obtain an upperbound by relaxing the integrality constraints of the original APSA model
Use Algorithm 1 to obtain an initial solution to the model, and store each shelf's individual contribution r_i to the objective value $r^* = \sum_{i \in \mathcal{B}} r_i$
Store the values of x, y and s
while no stopping criterium is met **do**
    while $|\text{tempset}| \geq |\mathcal{B}| \pmod{\tau}$ **do**
        Sort shelves in non-decreasing order based on their current value of r_i, and store these in Δ
        Split Δ into τ categories
        Randomly select a shelf i from each of the τ categories and remove this shelf i from tempset
        Solve the APSA model for the selected shelves and associated segments by considering the products currently allocated to the selected shelves, as well as products that have not been selected into the assortment.
        Update the shelf- and segment-specific values for x, y and s.
    end
end

Parameter	Set Instance	Value
w_{max}	all	50
$t_{\text{max,non-affinity}}$	all	1000.0 s
$t_{\text{max,affinity}}$	$\mathcal{N} = 240, \mathcal{B} = 30$	300.0 s
$t_{\text{max,affinity}}$	$\mathcal{N} = 320, \mathcal{B} = 40$	450.0 s
$t_{\text{max,affinity}}$	$\mathcal{N} = 400, \mathcal{B} = 50$	750.0 s
$t_{\text{max,affinity}}$	$\mathcal{N} = 480, \mathcal{B} = 60$	1000.0 s

Table 1: Parameter values of the stopping criteria employed in Algorithm 2.

3.3 Additional health-related constraints

To make sure the model places healthy items in high-attraction shelf segments, we need to create two subsets. Let $\mathcal{P} \subset \mathcal{N}$ denote the set of healthy food items, and let segments that have a high level of attraction f_k be stored in $\mathcal{Z} \subset \mathcal{K}$. Now, we can introduce a constraint, allowing healthy products, if selected, to only be placed in high-attraction shelf segments:

$$y_{kj} = 0, \forall k \notin \mathcal{Z}, j \in \mathcal{P}, \tag{3}$$

where y_{kj} denotes a binary variable with value 1 when product category j is allocated to shelf segment k, and 0 otherwise. Using constraint (3) in combination with the following constraint, healthy products could even be forced into the assortment:

$$\sum_{k \in \mathcal{Z}} y_{kj} \geq 1, \forall j \in \mathcal{P}. \tag{4}$$

Furthermore, RSPH (2019) recommend several changes to the layout of a store. Let us define $\mathcal{Q} \subset \mathcal{N}$ to denote the set of unhealthy food items. Note that there are product categories that are neither food nor drinks, and hence $\mathcal{P} \bigcup \mathcal{Q} \neq \mathcal{N}$. The RSPH recommends unhealthy food to be allocated to shelves that do not have high attractiveness. To achieve this, we can introduce the following constraint:

$$y_{kj} = 0, \forall k \in \mathcal{Z}, j \in \mathcal{Q}. \tag{5}$$

Moreover, it is advised to allocate a healthy product category to multiple shelves throughout the store. This can be achieved by replacing constraint (1b) in the original APSA model with the following two constraints:

$$\sum_{i \in B} x_{ij} \leq 1, \forall j \in \mathcal{N} \backslash \mathcal{P},$$
$$\sum_{i \in B} x_{ij} \leq d_{max}, \forall j \in \mathcal{P}, \tag{6}$$

where d_{max} denotes an upper limit on the number of shelves to which a healthy product category j can be allocated. In the current paper, d_{max} is taken to be 3.

Finally, the RSPH recommend more shelf space to be allocated to healthy foods. This would allow for an increased lower bound ℓ_j and upper bound u_j for the product categories $j \in \mathcal{P}$, and inversely a reduced lower bound and upper bound for the product categories $j \in \mathcal{Q}$. This makes up constraint (7) in our model.

3.4 Implementation

All of the above-described models and algorithms are implemented using Python (see Appendix A.9 for a detailed description of the modules and functions). Within Python, the docplex module (version 2.21.307), which uses the solver CPLEX, is available for model optimization purposes. The models and algorithms are run on an Apple Macbook Air with a 1.6GHz Dual-Core Intel Core i5 and 8 GB RAM space. This is suboptimal as compared to the computation environment employed by Flamand et al. (2018), who used a 3.40 GHz processor with associated 12 GB of RAM. Therefore, we expect to see a slightly worse performance in terms of computation time for models that are solved to optimality. For similar reasons, the obtained MIP gaps are expected to turn out slightly higher for models which reach their time limit.

4 Results

4.1 Replication

4.1.1 Choice of parameters

To replicate the results presented by Flamand et al. (2018) as accurately as possible, 10 instances of 4 pairs of $(\mathcal{N}, \mathcal{B})$ were simulated: $(\mathcal{N} = 240, \mathcal{B} = 30)$, $(\mathcal{N} = 320, \mathcal{B} = 40)$, $(\mathcal{N} = 400, \mathcal{B} = 50)$ and $(\mathcal{N} = 480, \mathcal{B} = 60)$. Each shelf $i \in \mathcal{B}$ is assumed to have 3 shelf segments k_i, stored in the set $\mathcal{K}_i$. Then, $\mathcal{K} = \bigcup_{i \in \mathcal{B}} \mathcal{K}_i$ denotes the set of segments. Each shelf segment k_i is assumed to have a capacity c_k of 6, such that the capacity $\mathcal{C}_i$ of a shelf $i \in \mathcal{B}$ equals 18. Furthermore, the minimum space requirement for a product category $j \in \mathcal{N}$, as denoted by ℓ_j, is drawn from $\mathcal{U} \sim [1, \frac{\mathcal{C}_i}{6}]$ and is rounded to the nearest integer. Consequently, the maximum space requirement u_j depends on the value of ℓ_j and is drawn from $\mathcal{U} \sim [\ell_j, \frac{\mathcal{C}_i}{3}]$ and also rounded to the nearest integer. A product category's associated profit, Φ_j, is drawn from another uniform distribution $\mathcal{U} \sim [1, 25]$ and rounded to two decimal places. The lowerbound on the amount of space allocated to a product category j is denoted by ϕ_j and takes the value of 0.1 for all $j \in \mathcal{N}$.

As for the attraction level of the shelf segments $k \in \mathcal{K}$, Flamand et al. (2018) mention that the attraction level to a certain product is based on its horizontal placement. The segments that are towards the beginning or end of the aisle are assumed to have a higher level of attraction. This can be substantiated by Larson et al. (2005), who conclude that the level of attractiveness of center-aisle product categories is significantly lower than end-of-aisle product categories. They also conclude that the placement of familiar products at the beginning or end of an aisle will lure customers into this aisle, resulting in an overall higher traffic rate for these aisles. Therefore, for the simulation, we categorize the shelves into five equal groups with associated values for t being 0.05, 0.25, 0.45, 0.65 and 0.85 respectively. Then, for middle shelf segments, we draw an associated attraction level from $\mathcal{U} \sim [t, t + 0.05]$, whereas for the other shelf segments, we draw from $\mathcal{U} \sim [t + 0.06, t + 0.10]$, all rounded to two decimal places. The products that fall within one of the affinity sets $\mathcal{L}, \mathcal{H}_1, \mathcal{H}_2$ and $\mathcal{H}_3$ have been chosen randomly without replacement.

The APSA model was solved by CPLEX and benchmarked against the performance of the heuristic in Table 2. For all instances of all sets, we can see that CPLEX fails to find an optimal solution within the time limit of 3,600 CPU seconds. This results is in line with the findings of Flamand et al. (2018). We also see that the optimality gaps increase with the number of shelves and products considered, which is in accordance with the results of the original paper. We obtain average MIP gaps of 0.677%, 0.926%, 1.301% and 1.662% for set 1, 2, 3, and 4 respectively. The original paper finds MIP gaps of 0.348%, 0.697%, 1.339% and 1.969% for these respective sets. It is interesting to compare the development of these set averages: whereas we find higher average gaps than the original paper for the smaller data instances, we find lower average gaps than the original paper for the larger data instances. This might have to do with the formulation of the set $\mathcal{R}$ by Flamand et al. (2018), who seem to iterate through all pairs of shelf segments when formulating this set. However, constraint (1b) of the original APSA model ensures that each product can only be assigned to one shelf. This means that the values of $y_{k1,j}$ and $y_{k2,j}$ can only both be one in cases where k_1 and k_2 belong to the same $\mathcal{K}_i$. Therefore, we put an extra restriction on set $\mathcal{R}$, namely that for the tuple (k_1, k_2, j) in $\mathcal{R}$, k_1 and k_2 are required to be elements in the same set $\mathcal{K}_i$. Formulating this additional constraint seems to be increasingly beneficial for larger data instances with larger amounts of shelf segments available, since having to go through a nested for-loop for all these segments can increase computation time greatly, thereby leaving less time to actually solve the model.

We can see that the heuristic with $\tau = 2$ is only able to find a feasible solution with an optimality gap smaller than 0.50% within the time limit of 1,000s twice, which is slightly lower the reported five solutions by Flamand et al. (2018). The overall average resulting MIP gap is 0.799%, which is very close to the reported 0.79% in the original paper. For $\tau = 3$, the heuristic is able to find a solution 82.5% of the time and the average MIP gap lies at 0.513%. This is suboptimal as compared to the results in the original paper, which showed a 100% success rate for $\tau = 3$ with an average MIP gap of 0.475%. However, the MIP gaps of the instances that the heuristic was not able to solve within the time limit do lie within reasonable bounds of 0.50%. The success rate for $\tau = 4$ with $\epsilon = 0.5\%$ lies at 98% with an average MIP gap of 0.49%, versus the 100% success rate and associated average gap of 0.453% presented in the original paper.

The results of the heuristic with each individual affinity set, as well as all the sets together, are benchmarked against the heuristic without any affinity sets with $\tau = 4$ and $\epsilon = 0.5\%$ in Table 3. These values for τ and ϵ were also taken for the models with the affinity sets. We see widely variant results for the heuristic for the different models. The heuristic with affinity set $\mathcal{L}$ has a success rate of 92.5% with an average MIP gap of 0.495%, whereas $\mathcal{H}_1$ performs much worse with a success rate of only 67.5% and an average gap of 0.548%. The model with $\mathcal{H}_2$ solves 82.5% of the time with an average gap of 0.510%, and the model with $\mathcal{H}_3$ solves 92.5% of the time with an average gap of 0.498%. The model with all affinity sets has a success rate of 50% with an average gap of 0.579%. These findings are in line with Flamand et al. (2018), who find success rates of 100% for the model with $\mathcal{L}$ with an average MIP gap of 0.448%, 50.0% for the model with $\mathcal{H}_1$ with an average gap of 0.550%, 87.5% for the model with $\mathcal{H}_2$ with an average gap of 0.459%, and 100% for the model with $\mathcal{H}_3$ with an average gap of 0.447%. For the model with all affinity sets included, Flamand et al. (2018) also find a poor performance with a success rate of only 27.5% with an average gap of 0.612%.

We can also see that the optimality gaps of the model with the set $\mathcal{H}_1$ individually and the model with the sets $\mathcal{L}, \mathcal{H}_1, \mathcal{H}_2$ and $\mathcal{H}_3$ simultaneously seems to decrease in an increasing number of products and shelves. The reason for this is that the amount of affinity pairs remained equal across the data set instances, making the impact of the affinity pairs relatively more significant in instances with a lower number of products $|\mathcal{N}|$. The models with $\mathcal{L}, \mathcal{H}_2$ and $\mathcal{H}_3$ individually seem to be less affected by a varying ratio of affinity pairs to the total number of products.

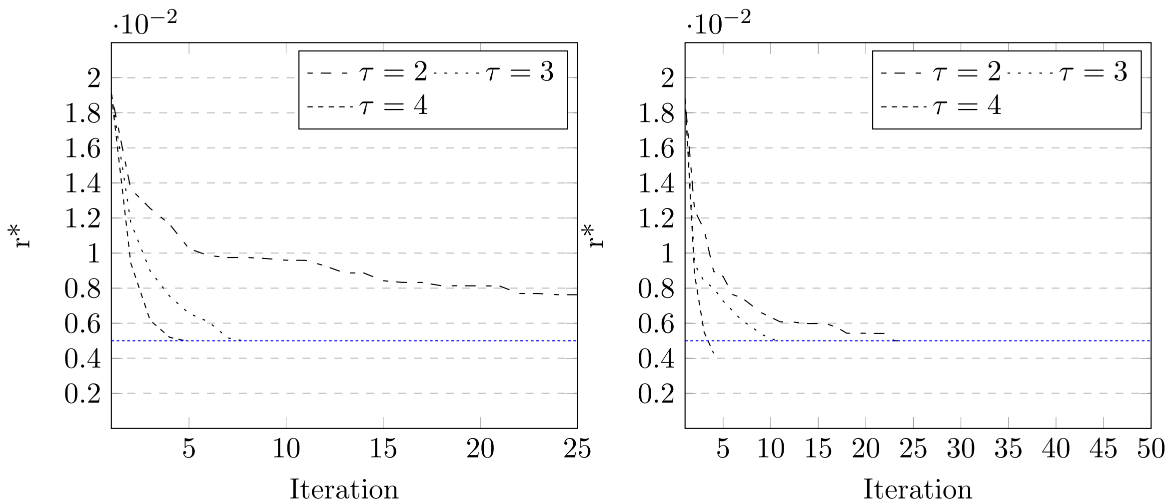
Set	$(\lvert\mathcal{B}\rvert, \lvert\mathcal{N}\rvert)$	*Inst.*	*APSA*	*Heuristic*				
				$\tau = 4$			$\tau = 3$	$\tau = 2$
				$\epsilon = 0.5\%$	$\epsilon = 1.0\%$	$\epsilon = 1.5\%$	$\epsilon = 0.5\%$	$\epsilon = 0.5\%$
			Gap(%)	*Gap(%)*	*Gap(%)*	*Gap(%)*	*Gap(%)*	*Gap(%)*
Set 1	(30, 240)	1	0.61% (3,600s)	0.49% (111.6s)	1.00% (69.3s)	1.34% (74.8s)	0.48% (442.2s)	0.66% (1,000s)
		2	0.96% (3,600s)	0.50% (114.8s)	0.96% (62.6s)	1.47% (84.0s)	0.50% (194.0s)	0.71% (1,000s)
		3	0.61% (3,600s)	0.49% (148.3s)	0.93% (71.9s)	1.49% (64.6s)	0.50% (303.8s)	0.84% (1,000s)
		4	0.86% (3,600s)	0.50% (294.8s)	0.99% (91.7s)	1.10% (73.3s)	0.48% (185.4s)	0.86% (1,000s)
		5	0.62% (3,600s)	0.46% (103.7s)	0.92% (57.9s)	1.35% (51.9s)	0.50% (225.0s)	0.69% (1,000s)
		6	0.18% (3,600s)	0.49% (233.2s)	0.99% (61.9s)	1.46% (56.6s)	0.63% (1,000s)	0.75% (1,000s)
		7	0.92% (3,600s)	0.47% (102.6s)	0.76% (55.6s)	1.48% (...)	0.50% (90.8s)	0.49% (434.8s)
		8	0.41% (3,600s)	0.42% (113.7s)	0.90% (57.1s)	1.37% (55.3s)	0.49% (193.6s)	0.50% (398.1s)
		9	0.61% (3,600s)	0.49% (97.0s)	0.79% (64.9s)	1.34% (76.4s)	0.49% (153.0s)	0.57% (1,000s)
		10	0.99% (3,600s)	0.47% (85.0s)	0.90% (63.6s)	1.38% (54.4s)	0.49% (285.8s)	0.52% (1,000s)
Set 2	(40, 320)	1	1.18% (3,600s)	0.49% (156.9s)	0.83% (117.0s)	1.40% (87.9s)	0.49% (214.3s)	0.62% (1,000s)
		2	0.54% (3,600s)	0.49% (306.8s)	0.92% (108.0s)	1.48% (94.7s)	0.49% (706.3s)	0.64% (1,000s)
		3	1.17% (3,600s)	0.49% (256.6s)	1.00% (140.6s)	1.37% (129.7s)	0.49% (606.4s)	1.03% (1,000s)
		4	0.85% (3,600s)	0.48% (165.0s)	1.00% (152.5s)	1.50% (118.5s)	0.50% (384.2s)	0.71% (1,000s)
		5	1.58% (3,600s)	0.50% (234.3s)	0.99% (131.9s)	1.05% (106.1s)	0.49% (533.1s)	0.77% (1,000s)
		6	0.65% (3,600s)	0.57% (1,000s)	0.97% (235.1s)	1.33% (107.8s)	0.67% (1,000s)	1.10% (1,000s)
		7	0.54% (3,600s)	0.49% (146.1s)	0.69% (108.0s)	1.39% (96.0s)	0.48% (401.2s)	0.60% (1,000s)
		8	0.66% (3,600s)	0.48% (192.2s)	1.00% (171.1s)	1.25% (128.1s)	0.49% (376.2s)	0.64% (1,000s)
		9	0.75% (3,600s)	0.49% (211.3s)	1.00% (131.6s)	1.36% (123.6s)	0.48% (412.0s)	0.67% (1,000s)
		10	1.34% (3,600s)	0.48% (179.3s)	0.94% (127.1s)	1.41% (105.9s)	0.50% (1,000s)	0.82% (1,000s)
Set 3	(50, 400)	1	1.80% (3,600s)	0.48% (299.5s)	0.99% (190.9s)	1.47% (154.1s)	0.50% (618.2s)	0.76% (1,000s)
		2	1.21% (3,600s)	0.50% (339.8s)	0.99% (245.8s)	1.47% (167.1s)	0.49% (677.9s)	0.85% (1,000s)
		3	1.63% (3,600s)	0.49% (475.7s)	0.94% (228.5s)	1.47% (157.2s)	0.50% (1,000s)	0.99% (1,000s)
		4	0.90% (3,600s)	0.49% (371.3s)	0.99% (211.2s)	1.50% (203.6s)	0.49% (847.2s)	0.79% (1,000s)
		5	1.01% (3,600s)	0.50% (405.9s)	0.95% (161.9s)	1.45% (150.6s)	0.50% (527.8s)	0.80% (1,000s)
		6	0.84% (3,600s)	0.50% (397.7s)	0.99% (211.2s)	1.48% (160.7s)	0.50% (713.5s)	0.99% (1,000s)
		7	1.33% (3,600s)	0.49% (405.1s)	0.98% (196.0s)	1.50% (180.0s)	0.50% (837.5s)	0.82% (1,000s)
		8	1.23% (3,600s)	0.50% (350.2s)	0.96% (181.7s)	1.50% (165.6s)	0.49% (588.7s)	0.67% (1,000s)
		9	1.48% (3,600s)	0.50% (303.3s)	0.96% (178.1s)	1.47% (160.5s)	0.49% (465.2s)	0.60% (1,000s)
		10	1.58% (3,600s)	0.50% (718.5s)	1.00% (276.2s)	1.49% (221.4s)	0.59% (1,000s)	0.89% (1,000s)
Set 4	(60, 480)	1	1.79% (3,600s)	0.50% (484.6s)	1.00% (304.9s)	1.42% (257.7s)	0.48% (800.3s)	1.01% (1,000s)
		2	1.70% (3,600s)	0.50% (664.5s)	0.89% (318.6s)	1.49% (264.1s)	0.50% (839.4s)	0.78% (1,000s)
		3	1.83% (3,600s)	0.50% (822.0s)	0.99% (371.3s)	1.44% (288.4s)	0.59% (1,000s)	0.96% (1,000s)
		4	1.58% (3,600s)	0.49% (655.7s)	0.97% (352.3s)	1.45% (287.1s)	0.63% (1,000s)	1.10% (1,000s)
		5	1.44% (3,600s)	0.50% (516.1s)	0.99% (361.0s)	1.40% (253.9s)	0.50% (736.0s)	1.01% (1,000s)
		6	1.75% (3,600s)	0.48% (528.7s)	0.88% (276.4s)	1.42% (265.8s)	0.50% (621.4s)	0.72% (1,000s)
		7	1.32% (3,600s)	0.49% (487.6s)	0.94% (340.3s)	1.48% (298.5s)	0.48% (767.8s)	0.81% (1,000s)
		8	2.35% (3,600s)	0.47% (565.3s)	0.87% (312.8s)	1.40% (277.1s)	0.50% (638.9s)	0.96% (1,000s)
		9	1.19% (3,600s)	0.49% (569.7s)	0.94% (344.4s)	1.48% (355.5s)	0.50% (833.5s)	0.90% (1,000s)
		10	1.67% (3,600s)	0.49% (465.2s)	0.92% (346.5s)	1.32% (310.6s)	0.50% (709.2s)	0.98% (1,000s)

Table 2: Model APSA as compared to the heuristic with varying permutations of the parameters $\tau = \{2, 3, 4\}$ and $\epsilon = \{0.5\%, 1.0\%, 1.5\%\}$. Note that *Gap* denotes the gap with the upperbound found by the original APSA model solved with CPLEX. The computation time of instance 7 of set 1 with $\tau = 7$ and $\epsilon = 1.5\%$ was incorrectly reported by Python due to a bug.

Set	$(\lvert\mathcal{B}\rvert, \lvert\mathcal{N}\rvert)$	*Inst.*	*Heuristic* $\tau = 4, \epsilon = 0.5\%$	*Heuristic* $\mathcal{L}$	*Heuristic* $\mathcal{H}_1$	*Heuristic* $\mathcal{H}_2$	*Heuristic* $\mathcal{H}_3$	*Heuristic* $\mathcal{L}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$
			Gap (CPU)	*Gap (CPU)*	*Gap (CPU)*	*Gap (CPU)*	*Gap (CPU)*	*Gap (CPU)*
Set 1	(30, 240)	1	0.49% (111.6s)	0.50% (129.3s)	0.49% (101.9s)	0.48% (100.3s)	0.48% (142.5s)	0.50% (127.7s)
		2	0.50% (114.8s)	0.49% (170.3s)	0.59% (300.0s)	0.49% (142.5s)	0.49% (164.6s)	0.66% (300.0s)
		3	0.49% (148.3s)	0.47% (232.2s)	0.53% (300.0s)	0.48% (139.8s)	0.49% (95.0s)	0.68% (300.0s)
		4	0.50% (294.8s)	0.50% (169.8s)	0.77% (300.0s)	0.51% (300.0s)	0.49% (198.2s)	0.79% (300.0s)
		5	0.46% (103.7s)	0.45% (116.8s)	0.50% (100.3s)	0.49% (297.1s)	0.50% (76.5s)	0.57% (300.0s)
		6	0.50% (233.2s)	0.54% (300.0s)	1.05% (300.0s)	0.66% (300.0s)	0.52% (303.9s)	1.17% (300.0s)
		7	0.47% (102.6s)	0.41% (81.9s)	0.49% (158.3s)	0.49% (86.6s)	0.50% (72.9s)	0.49% (215.3s)
		8	0.43% (113.7s)	0.49% (92.2s)	0.54% (300.0s)	0.50% (125.5s)	0.49% (91.3s)	0.59% (300.0s)
		9	0.50% (97.0s)	0.47% (96.0s)	0.48% (88.7s)	0.49% (183.2s)	0.49% (78.2s)	0.49% (132.5s)
		10	0.47% (85.0s)	0.47% (86.9s)	0.64% (300.0s)	0.50% (119.3s)	0.50% (96.5s)	0.66% (300.0s)
Set 2	(40, 320)	1	0.49% (156.9s)	0.50% (123.8s)	0.50% (322.9s)	0.49% (163.3s)	0.48% (137.4s)	0.50% (282.1s)
		2	0.49% (306.8s)	0.49% (294.2s)	0.58% (450.0s)	0.49% (249.6s)	0.50% (233.7s)	0.56% (450.0s)
		3	0.49% (256.6s)	0.48% (231.7s)	0.49% (279.5s)	0.50% (205.5s)	0.50% (263.4s)	0.50% (299.5s)
		4	0.48% (165.0s)	0.50% (150.1s)	0.49% (335.9s)	0.49% (194.8s)	0.47% (218.5s)	0.49% (345.6s)
		5	0.50% (234.3s)	0.47% (218.9s)	0.50% (307.6s)	0.49% (330.0s)	0.49% (214.4s)	0.56% (450.0s)
		6	0.57% (1,000s)	0.70% (450.0s)	0.49% (313.7s)	0.73% (450.0s)	0.62% (450.0s)	0.57% (450.0s)
		7	0.49% (146.1s)	0.49% (175.2s)	0.48% (186.6s)	0.49% (211.1s)	0.48% (173.1s)	0.49% (265.8s)
		8	0.48% (192.2s)	0.50% (229.0s)	0.50% (410.2s)	0.50% (192.5s)	0.49% (159.0s)	0.50% (329.8s)
		9	0.49% (211.3s)	0.50% (277.7s)	0.50% (251.1s)	0.48% (285.7s)	0.50% (231.9s)	0.49% (309.2s)
		10	0.48% (179.3s)	0.48% (438.6s)	0.83% (450.0s)	0.54% (450.0s)	0.53% (450.0s)	0.86% (450.0s)
Set 3	(50, 400)	1	0.48% (299.5s)	0.49% (354.2s)	0.50% (624.6s)	0.49% (346.2s)	0.50% (372.2s)	0.53% (750.0s)
		2	0.50% (339.8s)	0.47% (314.9s)	0.50% (584.5s)	0.46% (345.0s)	0.50% (372.1s)	0.49% (485.4s)
		3	0.49% (475.7s)	0.50% (501.3s)	0.74% (750.0s)	0.60% (750.0s)	0.50% (469.3s)	0.84% (750.0s)
		4	0.49% (371.3s)	0.49% (489.1s)	0.55% (750.0s)	0.49% (705.3s)	0.50% (441.4s)	0.68% (750.0s)
		5	0.50% (405.9s)	0.49% (315.6s)	0.50% (376.4s)	0.50% (445.4s)	0.49% (294.6s)	0.49% (557.3s)
		6	0.50% (397.7s)	0.49% (364.8s)	0.49% (422.1s)	0.49% (535.4s)	0.49% (345.6s)	0.52% (750.0s)
		7	0.49% (405.1s)	0.49% (320.9s)	0.50% (513.0s)	0.50% (516.2s)	0.50% (420.2s)	0.51% (750.0s)
		8	0.50% (350.2s)	0.50% (341.3s)	0.50% (743.0s)	0.50% (430.9s)	0.49% (338.3s)	0.63% (750.0s)
		9	0.50% (303.3s)	0.50% (358.8s)	0.49% (402.4s)	0.48% (324.5s)	0.49% (241.6s)	0.49% (241.6s)
		10	0.50% (718.5s)	0.52% (750.0s)	0.62% (750.0s)	0.50% (557.0s)	0.49% (337.5s)	0.54% (750.0s)
Set 4	(60, 480)	1	0.50% (484.6s)	0.49% (535.8s)	0.49% (500.0s)	0.49% (692.2s)	0.50% (517.4s)	0.48% (786.4s)
		2	0.50% (664.5s)	0.50% (549.8s)	0.48% (637.7s)	0.50% (609.9s)	0.50% (419.1s)	0.58% (1,200s)
		3	0.50% (822.0s)	0.50% (724.0s)	0.51% (1,200s)	0.63% (1,200s)	0.50% (757.8s)	0.69% (1,200s)
		4	0.49% (655.7s)	0.50% (728.3s)	0.61% (1,200s)	0.50% (1,200s)	0.49% (939.4s)	0.60% (1,200s)
		5	0.50% (516.1s)	0.50% (488.2s)	0.50% (500.4s)	0.49% (769.4s)	0.48% (550.3s)	0.50% (572.4s)
		6	0.48% (528.7s)	0.49% (414.2s)	0.50% (869.6s)	0.49% (406.1s)	0.50% (486.0s)	0.50% (984.8s)
		7	0.49% (487.6s)	0.49% (493.1s)	0.50% (726.6s)	0.50% (517.2s)	0.50% (552.2s)	0.50% (849.6s)
		8	0.47% (565.3s)	0.50% (472.9s)	0.50% (551.2s)	0.50% (590.3s)	0.50% (596.3s)	0.50% (678.0s)
		9	0.49% (569.7s)	0.50% (556.5s)	0.50% (847.5s)	0.48% (624.5s)	0.48% (605.0s)	0.49% (807.9s)
		10	0.49% (465.2s)	0.50% (555.3s)	0.50% (636.8s)	0.50% (517.7s)	0.50% (487.0s)	0.49% (465.2s)

Table 3: Model APSA as compared to the heuristic with the affinity sets $\mathcal{L}, \mathcal{H}_1, \mathcal{H}_2$, and $\mathcal{H}_3$. Note that *Gap* denotes the gap with the upperbound found by the original APSA model solved with CPLEX.

From Figure 1, we can see that the remark by Flamand et al. (2018) that the value of r* seems to make considerable jumps in the first couple of iterations, whereas the improvements seem to slow down when the number of iterations increase, indeed holds true. Note that the number of iterations does not necessarily decrease with an increasing value of τ, because the potential decrease in the value of r^* still depends significantly on the randomly selected shelves within the shelf categories in each iteration. A comparison of varying values of τ can be found in Appendix A.2.



(a) Instance 2 of Set 1 ($\mathcal{B} = 30, \mathcal{N} = 240$) (b) Instance 8 of Set 1 ($\mathcal{B} = 30, \mathcal{N} = 240$)

Figure 1: Plot of the value of the objective value r* over the iterations in Algorithm 2, starting from the first iteration

4.2 Health-related constraints in original simulation

The healthy products were selected to be $j \in \mathcal{P}$, where $\mathcal{P} = \{1, ..., 20\}$, whereas the unhealthy products were selected to be $j \in \mathcal{Q}$, where $\mathcal{Q} = \{21, ..., 40\}$. The product data associated to these sets are given in Table 6 in Appendix A.3. Furthermore, the 20 shelf segments with the highest attraction level f_k were selected into the set $\mathcal{Z}$.

Besides the additional health constraints imposed earlier, we also take into account the store layout. Most supermarkets employ a grid layout, in which several shelves are located in a straight line with intermediate passages between them. An alternative to the standard grid layout is the herringbone layout, in which a customer traverses through a single passage in the middle of the store to occasionally visit aisles that are located alongside this main passage. Whereas the grid layout is popular for large supermarkets, the herringbone layout can be a solution for smaller-sized stored to optimize their space. The store layouts, with their associated high-attraction shelf segments, can be found in Appendix A.4. The attraction levels were again based on the fact that beginning- and end-of-aisle shelf segments experience a higher level of attraction as compared to middle-aisle shelf segments. For now, the difference between the two store layouts only lies within the attraction levels of the shelf segments.

The results of the original APSA model and the health-model for each of the two store layouts, as solved by CPLEX, can be found in Table 4. For the grid layout, we see an increase from 12 healthy products in the APSA model to 20 products in the health-model, as enforced by constraint (4). There is a significant increase in the amount of shelves and segments the healthy products are allocated on. For the grid layout, we see that on 17 out of 30 shelves, there is at least one healthy product allocated, as compared to 10 in the APSA model. This is due to constraint (6), which allows healthy products $j \in \mathcal{P}$ to be allocated to at most three shelves instead of one. This occurs for product 4, 12, 17 and 18, which are all allocated to three shelves. These also happen to be the products with the four highest profit margins (see Table 6 in Appendix A.3), which lies in the line of expectation, as the objective function (1a) maximizes profit. Since these product categories take up more shelf space, less products can be selected into the assortment (only 128 were selected versus the 144 product categories selected by the original APSA model). Also, the amount of distinct segments has grown from 14 to 20. The average visibility of the healthy products increases from 0.171 to 0.889, due to higher shelf segment attraction levels f_k as enforced by constraint (3), and larger segment space s_{kj} allocated to the healthy products as enforced by constraint (7). The higher average visibility, together with the larger amount of (distinct) shelves and segments that healthy products are allocated to in the health-version of the model, implies that healthy products are going to be more visible throughout the entire store, and the conclusions by Kerr et al. (2012), Cohen et al. (2015) and Nakamura et al. (2014) would then imply that customers will actually also pick up more of these healthy product items during their shopping trip. Concerning the unhealthy products, the number of products selected into the store's product assortment decreased from 14 in the original APSA model to 12 in the health-model. The number of distinct shelves decreases from 14 to 12, and the number of distinct segments stays equal at 14. The average visibility of the unhealthy products decreases from 0.243 to 0.142, as would be expected by looking at constraint (5).

For the herringbone layout, we see very similar results as for the grid layout. While the number of healthy products, (distinct) shelves, (distinct) segments, and average visibility increases for the healthy products, the impact of the additional constraints is only marginal for the unhealthy products. However, we do see a significant drop in the objective value, which could be attributed to the

	Grid		Herringbone	
	APSA	*Health*	*APSA*	*Health*
Gap (%)	1.34	0.35	0.74	0.39
CPUs	3,600	3,600	3,600	3,600
Objective value upon termination	907.10	897.42	647.26	639.31
Best bound upon termination	919.26	900.54	652.07	641.81
B&B/C	3,163	5,669	2,088	3,817
Number of products selected	144	128	142	131
Healthy products	12	20	11	20
Shelves	12	28	11	27
Distinct shelves	10	17	10	17
Segments	14	28	12	27
Distinct segments	14	20	11	19
Average visibility ($\frac{f_k s_{kj}}{c_k}$)	0.171	0.889	0.135	0.614
Unhealthy products	14	12	14	14
Shelves	14	12	14	14
Distinct shelves	11	11	10	11
Segments	14	14	16	16
Distinct segments	14	14	15	15
Average visibility ($\frac{f_k s_{kj}}{c_k}$)	0.243	0.142	0.141	0.097

Table 4: Comparison of the original APSA model with the APSA model with all the health-related constraints

fact that the number of high-attraction shelf segments is lower in the herringbone layout than in the grid layout, since the former does not allow for as many beginning- or end-of-aisle shelf segments as the latter does. Namely, the grid layout has a total of 44 high-attraction shelf segments with associated $0.5 \leq f_k \leq 1.0$, whereas the herringbone layout only has 16. This is also the reason why the average visibility of the healthy products only lies at 0.614, since there are 4 shelf segments with an attraction level $f_k < 0.5$ in $\mathcal{Z}$, thereby forcing some healthy products into lower-attraction shelf segments. This is not necessarily a bad thing, since the products are still displayed in the 19 top-attraction shelf segments of the store, even though 3 of these segments have a low absolute attraction level.

The objective values upon termination for the health models seem to lie within reasonable distance from the objective value of the APSA model. The slight decrease between the two models is due to the consideration of healthy and unhealthy products. For instance, unhealthy products with a high profit margin would be allocated to high-attraction shelf segments in the APSA model, whereas they are restricted from doing so by the health model. The space that was allocated to the unhealthy product in the APSA model, now needs to be allocated to a product with a lower profit margin, thereby decreasing the overall objective value. Furthermore, the fact that healthy products are forced into the assortment means that low-profit healthy products that are not selected by the APSA model, take up the space of an alternative higher-profit product category in the health model. However, the objective value can possibly increase due to the fact that healthy products can be allocated to multiple shelves throughout the store. This means that the health model can leverage from high-profit healthy products by including them into more than one shelf.

4.3 Store simulation

The above-presented models maximize profit as defined by the objective function. This becomes clear when, for instance, looking at the APSA model with the grid layout. This model picks 144 products into the assortment, of which 138 are amongst the 144 products with the highest profit margin. For the health model with the grid layout, out of the 128 products selected, 118 belong to the 128 products with the highest profit margin. We see similar results for the models with the herringbone layout: out of the 142 product selected for the APSA model, 138 belong to the top 142 highest profit margin products, whereas for the health model, out of the 131 products selected, 122 belong to the 131 products with the highest profit margins. Although it is obvious that some suboptimal healthy products are now selected into the product assortment for the health models due to constraint (4), it seems like overall the high profit product categories are still selected. Therefore, by shifting around the shelves in the store, the store still contains the high-profit products, but the consumer can be nudged into buying healthy products by strategic positioning of the shelves. Hence, this simulation serves the purpose of finding a health-encouraging shelf positioning throughout the store given the profit-maximizing the product assortment and shelf allocation found by the APSA or the health model for a given store layout.

The layout of the store was parsed as an input to the program. This layout was designed in a 12-cell by 16-cell csv-file, where each cell represents a part of the store. Each cell contains one of the following values: -3, -2, -1, 0, or a positive integer: -3 denotes the exit node, -2 denotes the entry node, -1 indicates that the customers cannot walk in this part of the store, 0 means that a customer can walk in this part of the store, and a positive integer indicates a shelf segment (e.g. 1 indicates shelf segment 1 is located in this part of the store). The shelf segments come in tuples of three, comprising a shelf. These shelves can be positioned either horizontally or vertically in the store layout.

For the simulation, the following set-up was used:

- Customers can arrive in a 12 hour span (say from 9:00 am until 9:00 pm). There are a total of 43,200 seconds in these 12 hours. When customers the store, they are able to complete their shopping trip.

- 2,000 customers were simulated, each with their own arrival time in seconds. This arrival time, $t_{arrival}$, is measured relative to the opening time of the store, meaning $t_{arrival} = 3,600$ implies that the customer arrives at the store exactly at 10:00 am. The arrival time is calculated by means of the rounded-up value of a randomly generated $\mathcal{U} \sim [0; 43,200]$.

- A shopping list was created for each customer. The length of the shopping list is based on Larsen et al. (2020), where it was reported that the average number of products purchased in a grocery store was equal to 6.06, with a standard deviation of 6.15. This implies that the purchasing data cannot be approximated by a Poisson distribution (where the mean is assumed to be equal to the variance), but should rather be approximated by a Negative Binomial Distribution:

$$Pr[X = k] = \frac{\Gamma(\alpha^{-1} + k)}{k!\Gamma(\alpha^{-1})} \left(\frac{\alpha^{-1}}{\alpha^{-1} + \mu}\right)^{\alpha^{-1}} \left(\frac{\mu}{\alpha^{-1} + \mu}\right)^{k},$$

  where $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$ is the Gamma-distribution, $E(X) = \mu = 6.06$ and $Var(X) = \mu + \alpha\mu^2 = \mu(1 + \alpha\mu) = 6.15^2$, implying that $\alpha \approx 0.8649$.

- After obtaining the number of products to be bought by the customer, the products on the shopping list are generated by drawing a random product without replacement from the list of products that have both been selected by the APSA model and the model with additional health constraints for both layouts. Although here we lose the ability to select from all offered products in each of the resulting assortments, it allows us to "keep" the same simulated customers for the two models and the two store layouts, and thus a comparison of the additional sales can be made. 118 products remained, of which 11 are healthy and 11 are unhealthy.

- The length between each location in the store (which is equivalent to the length of a cell in the store lay-out csv) is assumed to be 5.0. The average walking speed of customers, as investigated by Larsen et al. (2020), is equal to 0.49 meters per second. Therefore, in the simulation, we assume a walking speed of 0.50 meters per second for each customer.

- The store layout will be investigated by means of a network graph. Edges, with their associated nodes, will be added to this graph when a customer is able to move from the entrance of the store (as indicated by a -2), or a walking space (as indicated by a 0), to one cell directly vertically or horizontally adjacent to their current cell. Customers can only move into cells with value 0 or value -3, the latter being exit of the store.

- Each customer starts at the entrance of the store and finishes at the exit of the store, and visits the nodes that gives the customer access to a product on their shopping list. For each of the shelf-permutations, Dijkstra's algorithm will be applied for each customer to determine their path through the store. Here, it is assumed that the customer will traverse from their current node to the closest unvisited node that contains one of the product categories on their shopping list. In case a product is located on multiple segments, the segment to visit is randomly chosen amongst all segments that contain the product at hand. Overall, this algorithm seems reasonable in terms of mimicking an actual customer journey throughout a supermarket, since customers might be inclined to move to the next closest item each time they cross an item off their shopping list. An illustration of the determination of a customer's path is given in Appendix A.5.

- Each time a customer moves from one node in their path to the next, the time will increase by the walking speed, as measured by the time it takes for the customer to get from one node to the next based on the customer's walking speed in meters per second.

- If the customer comes across nodes with direct access (either vertically or horizontally) to a segment, a uniformly-distributed random number between 0.0 and 1.0 will be drawn. If this number lies below the attraction level of the shelf segment, the customer "notices" the shelf segment and starts considering each of the products on this segment. For each product, a random number between 0.0 and 1.0 will be drawn, and if this number lies below the impulse purchase potential γ_j of the product, the customer buys this product as well. For healthy products, the impulse purchase potential is set to be 0.10, whereas for unhealthy products, it is set to 0.5. For products that fall in neither of these two categories, we assume the impulse purchase potential is 0.4. These rates are chosen based on Flamand et al. (2018), who found that the impulse purchase potential rates could be divided into three categories: (1) low [0, 0.1), medium [0.1, 0.4) and high [0.4, 0.5]. It is assumed that unhealthy products have a higher level of product impulse purchase potential, since customers are more inclined to pick up unhealthy items to reward themselves since these products have a higher hedonic value to the customer than their healthy alternatives. For each of the segments that the customer notices, a decision-time from $\mathcal{U} \sim [0, 26]$ will be drawn. This decision-time is based on Nielsen (2015), which states that on average, a customer makes a purchasing-decision in approximately 13 seconds.

- For a given set of 2,000 simulated customers, 1,000 random permutations of the shelf allocation throughout the store for each possible combination of the two models (APSA, health-constraints) and the two store layouts (grid, herringbone) will be considered. With "shelf permutation", it is meant that a product will stay on its allocated shelf, but the shelf moves throughout the store, giving the shelf segments a different attraction level and a different location. The permutation that leads to the highest amount of healthy product impulse purchases can be considered the most health-encouraging set-up for a store for that set of customers. Hence, this entire model and simulation will provide supermarket managers with an optimal health-inducing product assortment with associated shelf allocation, and a health-inducing store layout.

4.4 Comparison original APSA versus health-version of APSA

The results of the simulation are given in Table 5. Additional healthy sales and additional unhealthy sales are defined by the number of healthy and unhealthy products respectively that were not initially on the customer's shopping list, but have been picked up as an impulse purchase. The "score" is defined as the difference between the number of additional healthy sales and additional unhealthy sales, and serves as a measure of how well our model achieves its goal of stimulating a healthier purchase pattern. First of all, both the grid and the herringbone store layout with the health-model result in a much higher number of average additional healthy products picked up than the APSA model. These differences in means is significant, with $t(1998) = 150.90, p < 0.01$ for the grid layout, and $t(1998) = 58.09, p < 0.01$ for the herringbone layout. We also see that the grid layout with the health model results in a lower number of average additional unhealthy items sold during a shopping trip than the APSA model, $t(1998) = 42.36, p < 0.01$. For the herringbone layout, we see a higher number of average additional unhealthy sales, which is also significant, $t(1998) = 6.02, p < 0.01$. This might be due to the misspecification of high-attraction shelf segments, since it shows that low-attraction shelf segments as defined by the model receive high customer traffic. The overall score for both layouts is significantly lower, with $t(1998) = 84.77, p < 0.01$ for the grid layout and $t(1998) = 10.13, p < 0.01$ for the herringbone layout.

The fact that the sample standard deviation of additional unhealthy sales lies at a much higher level than the standard deviation of additional healthy sales might be due to the fact that the impulse purchase potential of unhealthy products lies at 0.5, whereas it lies at 0.1 for healthy products. If

	Grid		Herringbone	
Full sample	*APSA*	*Health*	*APSA*	*Health*
Average additional healthy sales	767.37	2511.69	927.84	2168.96
Sample standard deviation additional healthy sales	140.73	337.38	350.29	577.80
T-test statistic (one-tailed)		150.90***		58.09***
Average additional unhealthy sales	4894.65	3494.64	5071.25	5510.57
Sample standard deviation additional healthy sales	841.50	620.04	1492.62	1759.37
T-test statistic (one-tailed)		42.36***		6.02***
Average score	-4127.28	-982.95	-4143.41	-3341.61
Sample standard deviation score	878.12	777.74	1562.32	1955.41
T-test statistic (one-tailed)		84.77***		10.13***
Best permutation				
Additional healthy sales	966	3311	1140	3434
Additional unhealthy sales	2478	2025	1444	1584
Score	-1512	1286	-304	1850

Table 5: Comparison of the original APSA model with the APSA model with all the health-related constraints.

a healthy and an unhealthy product category only differ in their impulse purchase potential, we expect to see additional sales of around 5 times as high for unhealthy products as compared to their healthy counterparts. Moreover, for the herringbone layout, we see higher standard deviations for both types of additional sales, which could be due to the fact that in certain shelf permutations, the products that can be on a customer's shopping list (which is a subset of all products available in the store, as expanded upon in section 4.3) are randomly allocated to either many beginning-of-aisle shelf segments simultaneously, or many end-of-aisle shelf segments. In the first scenario, customers receive less product exposure since they do not have to traverse the entire aisle to retrieve a product, thereby decreasing overall additional sales. In the second scenario, the customer does have to traverse the entire aisle and is therefore exposed to more products, increasing overall additional sales.

In Figure 2, a heatmap of the total number of times each node was visited for the randomly generated set of 2,000 customers is presented. Due to the randomly generated shopping list of the customer, it seems logical that there is not much variation in the number of visits for aisle-nodes. However, it becomes immediately clear that segments at the beginning of the store receive an increased amount of exposure due to its geographical position within this layout. The fact that only healthy products comprise the complete shelf segments 9, 10 and 12 located at nodes close to the exit of the store implies that these products receive significant exposure due to the high level of customer traffic in these "final" nodes before reaching the exit. Furthermore, many healthy products are allocated to the high-attraction shelf segments at the beginning and end of each aisle, which allows them to be spotted from a node horizontally adjacent to the segment, as well as vertically. The heatmaps of the other simulations can be found in Appendix A.6.

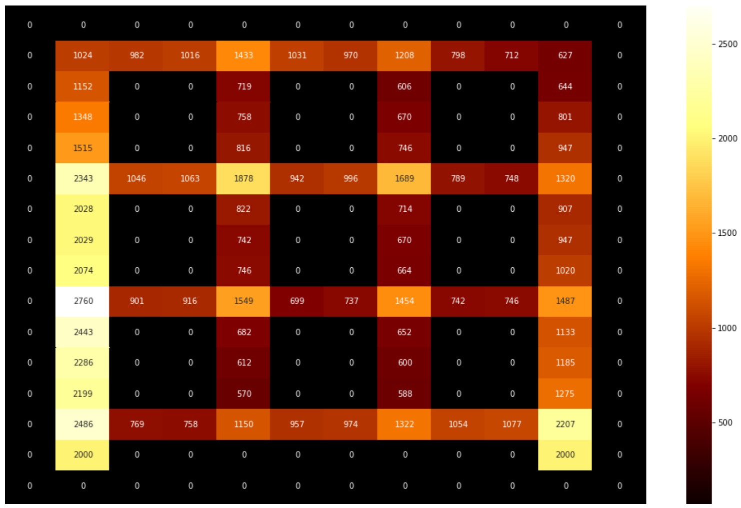


Figure 2: Heatmap best-attained store layout CS Grid

5 Discussion

By introducing health-related constraints to the APSA model, we see a great increase of the number of healthy products selected, the number of shelves and segments to which they are allocated to, and average segment attraction level. All together, this stimulates impulse purchases within the healthy food categories. Furthermore, unhealthy products find themselves in lower-attraction shelf segments, which decreases their overall attraction level and probability of being purchased. By employing this adjusted APSA model, store managers are able to introduce an overall healthier product assortment. The fact that the level of health-consciousness of customers keeps increasing due to better availability of information offers supermarkets and producers the opportunity to reformulate their current product assortment and provide the supply to meet the rising demand for healthy alternatives. Implementing the health-modified version of the APSA model therefore results in beneficial opportunities for all the parties involved: producers and supermarkets obtain a higher profit and a better brand image, whereas customers are encouraged to enjoy a healthier lifestyle.

The simulation offers stores the possibility to play into this health-conscious movement even further by implementing the most health-inducing shelf positioning. Stores could use this as a marketing opportunity, because customers might be more inclined to shop at a store that is advertised to have been proven to make customers pick up healthier products. Namely, customers often have the intention to pick up healthy products upon entering the store, but lose the required level of self-regulation to pursue this goal throughout their entire shopping trip. However, customers could also start rewarding themselves by purchasing additional unhealthy products since they convince themselves that by shopping in a store like this have made them unconsciously pick up more healthy products, while this might not necessarily be the case. The exact effect of advertising with this mix of marketing variables should therefore be further investigated.

By easier accessibility of healthier food products by storing them in highly-visible shelf segments and on multiple shelves throughout the store, customers experience less effort obtaining them, which increases the probability of purchase. Furthermore, by including more healthy products into the assortment, the customer's need for variety is satisfied, and the healthy products are perceived as the default option as compared to the unhealthy products. This introduces the unwritten norm to be healthy products, and customers have to go out of their ways to pick up the unhealthy items.

The current paper comes with several limitations. First of all, many assumptions on the customers, products and shelf segments had to be made. Although these assumptions were based on the current body of literature within this field, the assumptions should be cross-checked with real-life data (e.g. on the actual segment attraction level, customer movement thoughout the store, etc.). Fortunately, the model has been formulated in such a way to allow for a considerably level of flexibility when it comes to these assumptions. This level of flexibility also holds true for the store layouts.

The algorithm could use some additional rules when it comes to customer movement throughout the store. When a customer is faced with the decision between two products which lie at equal (minimum) distance, the first product of the shopping list is selected to traverse to. Although rarely, this can cause unrealistic movements in the herringbone layout specifically. For instance, if a customer enters the store, and the first product on the shopping list lies in segment 82, whereas the second product on the list lies in segment 84. The algorithm chooses to first let the customer travel to the node adjacent to segment 82, and only then to the node adjacent to segment 84. Realistically speaking, the customer would probably first visit segment 84 and only then segment 82, as to prevent having to walk back, and thereby increasing the overall distance walked on their shopping trip. If the customer also has a product on their shopping list that is stored in segment 81, they will travel from segment 82 to segment 81 first (since segment 81 is now closer to the current location of the customer than segment 84) before possibly traversing back to segment 84. Fortunately, the impact on the layout score is minimal, since this only occurs for approximately 1% of the 2,000 customers.

Additionally, the currently portrayed traffic density is influenced by the selection of the shortest path amongst multiple shortest paths of equal lengths. For instance, from the top right node to the store exit has $\binom{6}{3}$ possible shortest paths. Namely, each path consists of 6 "moves": at each intersection, the customer would have to choose whether to traverse the aisle to the south or to the west. Out of the 6 moves, 3 have to be west, and 3 have to be south to get to the store's exit in a shortest path. However, the Python network graph module is deterministic in returning a shortest path. Namely, for this example, it always returns the path first going completely westward, and then completely southwards. While it seems more logical for customers to first traverse in one direction in its entirety before traversing in a different direction (e.g. either first south and then west, or first west and the south, instead of choosing alternating directions), the results of the simulation might still be biased to some extent since the realistic path of first south and then west is not considered. Future research could adjust the method to randomly select amongst all shortest paths to possibly remove this bias.

Another qualification of the current methodology concerns the one-dimensionality of the attraction levels of shelf segments considered in the models. The attraction levels of the shelf segments in the current paper were chosen based on the consensus in the literature that beginning- and end-of-aisle shelf segments receive more attention from the customer. Although Flamand et al. (2018) noted that shelf segments at the beginning and end of the store also have a significantly larger attraction level than segments located elsewhere, this was not taken into account in the current research due to the ambiguity concerning which shelf segments exactly would experience this larger attraction level. When looking at the heatmaps, we can see that certain shelf segments at the beginning and end of the store indeed experience a larger traffic density. In future research, one can take these heatmaps as a basis of determining exactly which shelf segments experience higher customer traffic in a certain store layout.

Finally, the queue is an important place in the store in the context of impulse purchases. Queues were initially considered in the simulation as well, but were left out due to the scope of the current paper. This is the reason why for instance the arrival time of a customer and their walking speed is taken into account. The initial framework developed to accurately take into account the queues in this health-promoting store layout can be found in Appendix A.7 and A.8.

References

Adjoian, T., Dannefer, R., Willingham, C., Brathwaite, C., and Franklin, S. (2017). Healthy checkout lines: a study in urban supermarkets. *Journal of nutrition education and behavior*, 49(8):615–622.

Alliance, O. H. (2018). Out of place – the extent of unhealthy foods in prime locations in supermarkets. Technical report.

Brightwell, G., Fairthorne, M., and Luczak, M. J. (2018). The supermarket model with bounded queue lengths in equilibrium. *Journal of Statistical Physics*, 173(3):1149–1194.

Brightwell, G. and Luczak, M. (2012). The supermarket model with arrival rate tending to one. *arXiv preprint arXiv:1201.5523*.

Broniarczyk, S. M. and Hoyer, W. D. (2006). Retail assortment: more≠ better. In *Retailing in the 21st Century*, pages 225–238. Springer.

Brug, J. (2008). Determinants of healthy eating: motivation, abilities and environmental opportunities. *Family practice*, 25(suppl_1):i50–i55.

Caspi, C. E., Lenk, K., Pelletier, J. E., Barnes, T. L., Harnack, L., Erickson, D. J., and Laska, M. N. (2017). Association between store food environment and customer purchases in small grocery stores, gas-marts, pharmacies and dollar stores. *International Journal of Behavioral Nutrition and Physical Activity*, 14(1):1–11.

Chandon, P., Hutchinson, J. W., Bradlow, E. T., and Young, S. H. (2009). Does in-store marketing work? effects of the number and position of shelf facings on brand attention and evaluation at the point of purchase. *Journal of marketing*, 73(6):1–17.

Chapman, K. and Ogden, J. (2012). Nudging customers towards healthier choices. an intervention in the university canteen. *Journal of Food research*, 1(2):13–21.

Cohen, D. A., Collins, R., Hunter, G., Ghosh-Dastidar, B., and Dubowitz, T. (2015). Store impulse marketing strategies and body mass index. *American journal of public health*, 105(7):1446–1452.

Crawford, D., Ball, K., Mishra, G., Salmon, J., and Timperio, A. (2007). Which food-related behaviours are associated with healthier intakes of fruits and vegetables among women? *Public health nutrition*, 10(3):256–265.

Dhar, S. K., Hoch, S. J., and Kumar, N. (2001). Effective category management depends on the role of the category. *Journal of Retailing*, 77(2):165–184.

Finnegan Jr, J. R., Viswanath, K., Rooney, B., McGovern, P., Baxter, J., Elmer, P., Graves, K., Hertog, J., Mullis, R., Pirie, P., et al. (1990). Predictors of knowledge about healthy eating in a rural midwestern us city. *Health Education Research*, 5(4):421–431.

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.

Handbury, J., Rahkovsky, I., and Schnell, M. (2015). *What Drives Nutritional Disparities?: Retail Access and Food Purchases Across the Socioeconomic Spectrum*. Number w21126. National Bureau of Economic Research Cambridge, MA.

Horsley, J. A., Absalom, K. A., Akiens, E. M., Dunk, R. J., and Ferguson, A. M. (2014). The proportion of unhealthy foodstuffs children are exposed to at the checkout of convenience supermarkets. *Public Health Nutrition*, 17(11):2453–2458.

Huitink, M., Poelman, M. P., Seidell, J. C., Pleus, M., Hofkamp, T., Kuin, C., and Dijkstra, S. C. (2020). Can unhealthy food purchases at checkout counters be discouraged by introducing healthier snacks? a real-life experiment in supermarkets in deprived urban areas in the netherlands. *BMC public health*, 20:1–9.

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.

Kerr, J., Sallis, J. F., Bromby, E., and Glanz, K. (2012). Assessing reliability and validity of the gropromo audit tool for evaluation of grocery store marketing and promotional environments. *Journal of nutrition education and behavior*, 44(6):597–603.

Larsen, N. M., Sigurdsson, V., Breivik, J., and Orquin, J. L. (2020). The heterogeneity of shoppers' supermarket behaviors based on the use of carrying equipment. *Journal of Business Research*, 108:390–400.

Larson, J. S., Bradlow, E. T., and Fader, P. S. (2005). An exploratory look at supermarket shopping paths. *International Journal of research in Marketing*, 22(4):395–414.

Levy, L. (2013). Dietary strategies, policy and cardiovascular disease risk reduction in england. *Proceedings of the Nutrition Society*, 72(4):386–389.

Levy, M. and Weitz, B. A. (1995). Retailing management. richard d. irwin. *Inc., Chicago, IL, USA*.

Miwa, K. and Takakuwa, S. (2008). Simulation modeling and analysis for in-store merchandizing of retail stores with enhanced information technology. In *2008 Winter Simulation Conference*, pages 1702–1710. IEEE.

Morabito, R. and de Lima, F. C. R. (2004). A markovian queueing model for the analysis of user waiting times in supermarket checkouts. *International Journal of Operations and Quantitative Management*, 10:165–178.

Nakamura, R., Pechey, R., Suhrcke, M., Jebb, S. A., and Marteau, T. M. (2014). Sales impact of displaying alcoholic and non-alcoholic beverages in end-of-aisle locations: An observational study. *Social Science & Medicine*, 108:68–73.

Nederkoorn, C., Guerrieri, R., Havermans, R., Roefs, A., and Jansen, A. (2009). The interactive effect of hunger and impulsivity on food intake and purchase in a virtual supermarket. *International journal of obesity*, 33(8):905–912.

Nielsen (2015). Make the most of your brand's 20-second window.

OECD (2017). Obesity update 2017. Technical report.

of Agriculture, U. S. D. (2013). Supplemental nutrition assistance program education and evaluation study (wave ii). Technical report.

Pan, Y. and Zinkhan, G. M. (2006). Determinants of retail patronage: a meta-analytical perspective. *Journal of retailing*, 82(3):229–243.

Questionmark (2020). Superlijst gezondheid 2020. Technical report.

RIVM (2021). Rivm collaborates at european level on the implemention of the nutri-score label.

RSPH (2019). Health on the shelf. Technical report.

Sorensen, H. (2003). The science of shopping. *Marketing Research*, 15(3):30–30.

Van Kleef, E., Otten, K., and van Trijp, H. C. (2012). Healthy snacks at the checkout counter: A lab and field study on the impact of shelf arrangement and assortment structure on consumer choices. *BMC public health*, 12(1):1–10.

Verplanken, B., Herabadi, A. G., Perry, J. A., and Silvera, D. H. (2005). Consumer style and health: The role of impulsive buying in unhealthy eating. *Psychology & Health*, 20(4):429–441.

Vohs, K. D. and Faber, R. J. (2007). Spent resources: Self-regulatory resource availability affects impulse buying. *Journal of consumer research*, 33(4):537–547.

Wilkinson, J. B., Mason, J. B., and Paksoy, C. H. (1982). Assessing the impact of short-term supermarket strategy variables. *Journal of Marketing Research*, 19(1):72–86.

Wilson, A. L., Buckley, E., Buckley, J. D., and Bogomolova, S. (2016). Nudging healthier food and beverage choices through salience and priming. evidence from a systematic review. *Food Quality and Preference*, 51:47–64.

Winkler, L. L., Christensen, U., Glümer, C., Bloch, P., Mikkelsen, B. E., Wansink, B., and Toft, U. (2016). Substituting sugar confectionery with fruit and healthy snacks at checkout–a win-win strategy for consumers and food stores? a study on consumer attitudes and sales effects of a healthy supermarket intervention. *BMC Public Health*, 16(1):1–12.

A Appendix

A.1 SSP(i*) model

Maximize

$$\sum_{k\in\mathcal{K}_{i^*}} \sum_{j\in\mathcal{N}\setminus\mathcal{S}} \Phi \frac{f_k s_{kj}}{c_k} \tag{2a}$$

s.t.

$$\textstyle\sum_{j\in\mathcal{N}\setminus\mathcal{S}} s_{kj} \leq c_k \qquad \forall k \in \mathcal{K}_{i^*} \tag{2c}$$

$$\ell_j \textstyle\sum_{i\in\mathcal{B}} w_j \leq \sum_{k\in\mathcal{K}_{i^*}} s_{kj} \leq u_j \sum_{i\in\mathcal{B}} x_{ij} \qquad j \in \mathcal{N}\setminus\mathcal{S} \tag{2d}$$

$$\phi_j y_{kj} \leq s_{kj} \leq \min\{c_k, u_j\} y_{kj} \qquad \forall j \in \mathcal{N}\setminus\mathcal{S}, k \in \mathcal{K}_{i^*} \tag{2e}$$

$$s_{k_2,j} \geq c_{k_2}(y_{k_1,j} + y_{k_3,j} - 1) \qquad \forall j \in \mathcal{N}\setminus\mathcal{S}, i \in \mathcal{B}, k_1, k_2, k_3 \in \mathcal{K}_{i^*} | k_1 < k_2 < k_3 \tag{2f}$$

$$y_{kj} \leq w_j \qquad \forall i \in \mathcal{B}, j \in \mathcal{N}\setminus\mathcal{S}, k \in \mathcal{K}_{i^*} \tag{2g}$$

$$w_j \leq \textstyle\sum_{k\in\mathcal{K}_{i^*}} y_{kj} \qquad \forall i \in \mathcal{B}, j \in \mathcal{N}\setminus\mathcal{S} \tag{2h}$$

$$q_{kj} \geq y_{kj} + y_{k+1,j} - 1 \qquad \forall i \in \mathcal{B}, j \in \mathcal{N}\setminus\mathcal{S}, k \in \mathcal{K}_{i^*}\setminus\{\beta_i^*\} \tag{2i}$$

$$\textstyle\sum_{j\in\mathcal{N}\setminus\mathcal{S}} q_{kj} \leq 1 \qquad \forall i \in \mathcal{B}, k \in \mathcal{K}_{i^*}\setminus\{\beta_i^*\} \tag{2j}$$

$$w_j + w_{j'} \leq 1 \qquad \forall (j, j') \in \mathcal{L}, i \in \mathcal{B} \tag{2k}$$

$$w_j - w_{j'} = 0 \qquad \forall (j, j') \in \mathcal{H}_1, i \in \mathcal{B} \tag{2l}$$

$$w_j \leq w_{j'} \qquad \forall (j, j') \in \mathcal{H}_2, i \in \mathcal{B} \tag{2m}$$

$$y_{k_1,j} + y_{k_2,j} \leq 1 \qquad \forall (k_1, k_2, j) \in \mathcal{R} | k_1, k_2 \in \mathcal{K}_{i^*} \tag{2n}$$

A.2 Results heuristic for varying values of τ

Figure 3 shows the development of the objective value r^* over the iterations for increasing values of τ. For $\tau = 2$, the algorithm is not able to solve within the time limit of 1,000 seconds. For $\tau \in \{3, ..., 6\}$, the algorithm is able to solve within the time limit, and the number of iterations needed decreases with an increasing value of τ. Then, for $\tau > 6$, the algorithm only needs one iteration to solve for $\epsilon < 0.5\%$, but takes a larger time. Namely, for $\tau = 7, 8, 9, 10$, the solving time is respectively $1032.72, 2710.06, 6164.57$ and 4880.98 seconds. The fact that the solving time for $\tau = 9$ is larger than for $\tau = 10$ is due to the randomness of choosing shelves within the respective shelf categories. Since only one iteration with 3 re-optimization steps were needed, the attraction levels of the randomly selected shelves and their associated segments can greatly impact the performance of the algorithm.

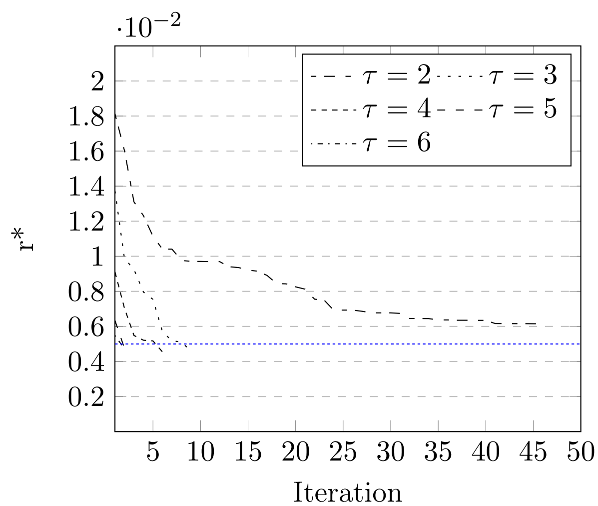


Figure 3: Plot of the value of the objective value r* over the iterations in Algorithm 2, starting from the first iteration, for varying values of τ

A.3 Product information

j	ℓ_j	u_j	Φ_j	ϕ_j
1	2	6	18.57	0.1
2	2	3	4.74	0.1
3	1	5	15.43	0.1
4	2	2	24.28	0.1
5	3	4	5.36	0.1
6	1	3	13.59	0.1
7	2	3	15.68	0.1
8	1	2	9.79	0.1
9	2	5	5.79	0.1
10	2	4	2.11	0.1
11	2	3	2.56	0.1
12	3	6	20.40	0.1
13	2	2	17.42	0.1
14	2	2	12.88	0.1
15	1	6	7.21	0.1
16	2	3	13.48	0.1
17	2	3	24.27	0.1
18	3	6	22.48	0.1
19	2	6	3.12	0.1
20	1	1	8.81	0.1

j	ℓ_j	u_j	Φ_j	ϕ_j
21	2	3	20.89	0.1
22	2	3	14.02	0.1
23	1	5	2.79	0.1
24	3	5	5.77	0.1
25	1	5	17.96	0.1
26	2	5	2.78	0.1
27	2	2	21.71	0.1
28	2	3	2.53	0.1
29	2	3	18.51	0.1
30	2	6	12.33	0.1
31	1	5	19.26	0.1
32	2	5	12.85	0.1
33	2	4	1.61	0.1
34	1	1	16.27	0.1
35	2	4	22.78	0.1
36	1	3	19.13	0.1
37	1	1	7.95	0.1
38	1	6	20.39	0.1
39	2	5	20.29	0.1
40	1	5	13.94	0.1

Table 6: Data on the healthy products (left) and the unhealthy products (right). ℓ_j denotes the minimum space requirement of product category j, u_j denotes the maximum space requirement of product category j, Φ_j denotes the largest possible profit of product category j, and ϕ_j denotes the minimum allocated space for product category j.

A.4 Store Layouts

-1	79	80	81	82	83	84	85	86	87	-1	-1
1	0	0	0	0	0	0	0	0	0	0	67
2	0	13	22	0	31	40	0	49	58	0	68
3	0	14	23	0	32	41	0	50	59	0	69
4	0	15	24	0	33	42	0	51	60	0	70
5	0	0	0	0	0	0	0	0	0	0	71
6	0	16	25	0	34	43	0	52	61	0	72
7	0	17	26	0	35	44	0	53	62	0	73
8	0	18	27	0	36	45	0	54	63	0	74
9	0	0	0	0	0	0	0	0	0	0	75
10	0	19	28	0	37	46	0	55	64	0	76
11	0	20	29	0	38	47	0	56	65	0	77
12	0	21	30	0	39	48	0	57	66	0	78
-1	0	0	0	0	0	0	0	0	0	0	-1
-1	-3	-1	-1	88	89	90	-1	-1	-1	-2	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 4: The grid layout, with the following meaning: Store entrance (-2), store exit (-3), walkable area (0), segment (>0), and non-walkable area(-1). Segments cannot be walked either, but should be accessed from a walkable area (0) horizontally or vertically adjacent to the segment.

The following segments were considered as high-attraction shelf segments: 1, 12, 13, 15, 16, 18, 19, 21, 22, 24, 25, 27, 28, 30, 31, 33, 34, 36, 37, 39, 40, 42, 43, 45, 46, 48, 49, 51, 52, 54, 55, 57, 58, 60, 61, 63, 64, 66, 67, 78, 79, 87, 88, and 90. These segments have an attraction level $f_k \sim \mathcal{U}[0.5, 1.0]$, whereas the other segments have an attraction level $f_k \sim \mathcal{U}[0, 0.5]$.

1	0	13	25	0	37	49	0	61	73	0	-1
2	0	14	26	0	38	50	0	62	74	0	-3
3	0	15	27	0	39	51	0	63	75	0	-1
4	0	16	28	0	40	52	0	64	76	0	-1
5	0	17	29	0	41	53	0	65	77	0	-1
6	0	18	30	0	42	54	0	66	78	0	85
0	0	0	0	0	0	0	0	0	0	0	86
0	0	0	0	0	0	0	0	0	0	0	87
0	0	0	0	0	0	0	0	0	0	0	88
0	0	0	0	0	0	0	0	0	0	0	89
7	0	19	31	0	43	55	0	67	79	0	90
8	0	20	32	0	44	56	0	68	80	0	-1
9	0	21	33	0	45	57	0	69	81	0	-1
10	0	22	34	0	46	58	0	70	82	0	-1
11	0	23	35	0	47	59	0	71	83	0	-2
12	0	24	36	0	48	60	0	72	84	0	-1

Figure 5: The herringbone layout, with the following meaning: Store entrance (-2), store exit (-3), walkable area (0), segment (>0), and non-walkable area(-1). Segments cannot be walked either, but should be accessed from a walkable area (0) horizontally or vertically adjacent to the segment.

The following segments were considered as high-attraction shelf segments: 6, 7, 18, 19, 30, 31, 42, 43, 54, 55, 66, 67, 78, 79, 85, 90. These segments have an attraction level $f_k \sim \mathcal{U}[0.5, 1.0]$, whereas the other segments have an attraction level $f_k \sim \mathcal{U}[0, 0.5]$.

A.5 Illustration of path-determination

To illustrate the simulation, let us consider customer 14. This customer has three products on their shopping list: product 34, 27 and 199. In the original APSA model with herringbone segment attraction levels, these products are stored in segments 22, 37 and 6 respectively. These segments are located on node 158, 5, and 73 in the herringbone layout, where the node number is a counter starting from the top-left at value 1 to the bottom-right at value 192 of the layout presented. The distance matrix between these nodes, the starting node S (the store entrance) and exit node E (store exit) is given in Table 7. These distances are the shortest distances between the two nodes and are calculated by means of Dijkstra's algorithm. The associated network graph is given in Figure 6. The shortest distance between the starting node and a node adjacent to the starting node is chosen as the first node to visit, which is node 158. Note that the edge weight between node S and node 158 and 73 respectively is both 19. Yet, because the product at node 158 is listed before the product at node 73 on the customer's shopping list, node 158 is chosen as the first node to travel to. As soon as the customer has arrived at node 158 and has picked up their product, they only need to visit nodes 73 and 5 before visiting the exit node. The distance to node 73 is the shortest (8), and hence the customer will travel to this node next. Then, from node 73, only node 5 still needs to be visited, which requires the customer to travel a distance of 10. Now, all nodes that needed to be visited have been visited, and the customer will travel to the exit node. The exit node cannot be visited before all required nodes have been visited.

	S	158	5	73	E
S	0	19	21	19	15
158	∞	0	16	8	22
5	∞	16	0	10	18
73	∞	8	10	0	16
E	∞	∞	∞	∞	0

Table 7: Distance matrix between the nodes that customer 14 needs to visit on their shopping trip.

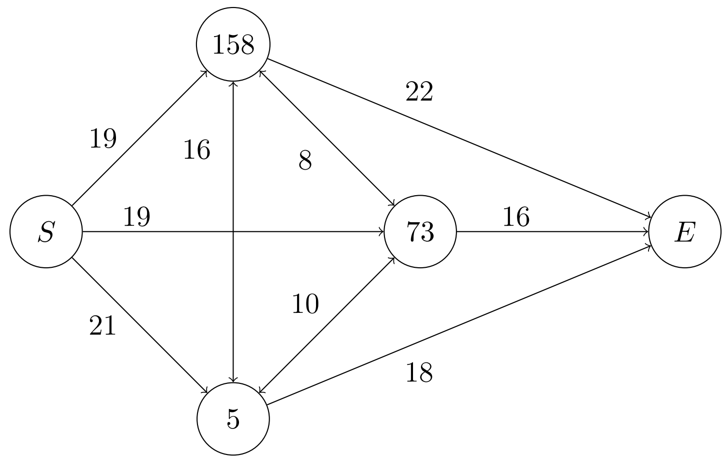


Figure 6: Simplified network graph associated to customer 14.

A.6 Heatmaps

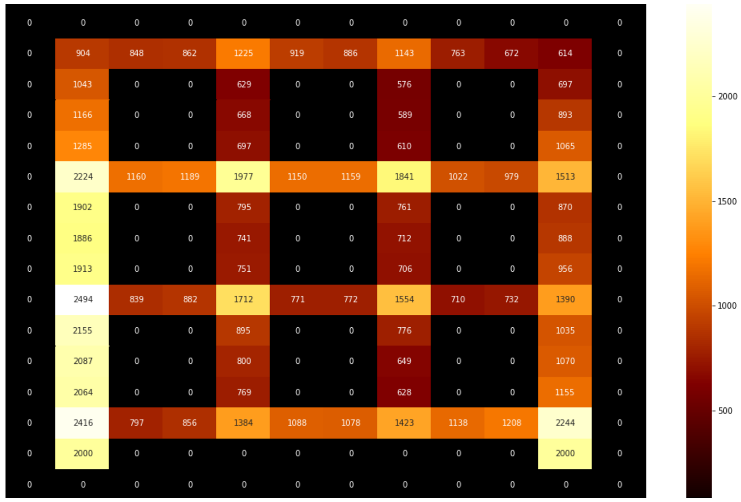


Figure 7: Heatmap best-attained store layout APSA Grid

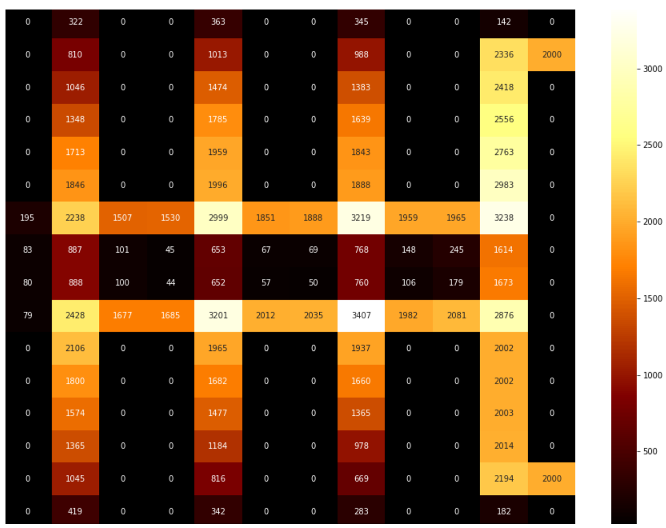


Figure 8: Heatmap best-attained store layout APSA Herringbone

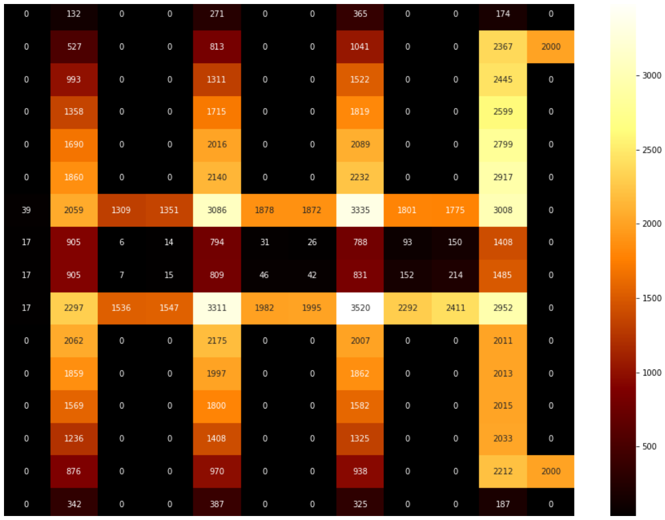


Figure 9: Heatmap best-attained store layout CS Herringbone

A.7 The role of queues

The queue at the checkout serves as one of the most profitable shelf segments in a store. This has several reasons. First of all, every customer has to pass the queue, and thus the products on the shelf segment, when they want to leave the store (Horsley et al., 2014). Furthermore, when the customer has to stand in line, they are continuously exposed to the products for a relatively long period of time, which increases the probability of impulse purchases within the categories on offer (Chapman and Ogden, 2012). Here, it is important to note that an individual's self-regulation heavily depends on a finite level of so-called regulatory resources that tend to exhaust relatively quickly. Therefore, after resisting the urge to pick up unhealthy snacks during your entire shopping trip, which made the individual's energy resources decrease significantly, it is much more difficult for the individual to resist the upcoming urge to buy the unhealthy snack impulsively (Vohs and Faber, 2007).

Customers might try to limit the effort put in to obtain food , and increasing the number of healthy food items offered throughout the store increases the product's visibility, and in turn, desirability. The results of the study by Van Kleef et al. (2012) confirm this, since it is shown that the number of shelf facings of healthy versus unhealthy products significantly influences the consumer's perception of the products and brands on display. The same result was independently found by Wilkinson et al. (1982) and Chandon et al. (2009).

Huitink et al. (2020) recommend to limit a supermarket's unhealthy product assortment at checkout and use that shelf space to instead place healthy items at all the checkouts. This recommendation can be confirmed by the findings of several studies. For instance, Adjoian et al. (2017) found that by increasing the number of healthier products at checkout, customers are more likely to pick these up. Moreover, Winkler et al. (2016) found that a positive effect when replacing unhealthy snacks by carrot snack packs at the checkout shelves. Furthermore, they found that supermarkets can enhance their brand image as customers expressed in exit interviews that they perceive this to be a positive intervention.

A.8 Optimizing queues

The importance of the queueing system in this simulation is the fact that the RSPH (2019) has recommended placing healthy products at the checkout to induce more healthy purchases. Most of the checkout-sales in current supermarkets are unhealthy candy bars and snacks, and therefore, in the APSA model, unhealthy products will be placed in these segments. In the health-version of the model, healthy products will be placed in these segments instead. The selection of these products occurs in a similar fashion as any other product in the "regular" portion of the store.

Now, let us assume that there are three shelf segments along the queue of a supermarket's checkout. Customers who are not in front of the queue are exposed to segments with healthy products. Each of the first three positions in the queue is exposed to a different segment. This means that customers who enter a queue in the fourth position are exposed to all three segments while traversing through the queue over time. The customers who are immediately serviced are assumed not to be exposed to these segments. Customers second and third in line are respectively exposed to the first and the first two segments along their queueing journey. In this case, we would ideally want to optimize the number of customers in the queue in such a way that a random customer finds themselves entering a queue in position 4. This is the situation when one customer (in the front of the queue) is currently being serviced, and the other three customers are waiting and thus exposed to the shelf segments directly adjacent to their position in the queue. To avoid unnecessary long lines in the store, which possibly causes in-store congestion, as well as dissatisfaction for the customer, we cannot simply opt for a very small number of queues to achieve this.

The service time of a customer at the front of the queue is assumed to be $2.43 * p + 14.79$, where p denotes the number of products in the customer's basket. This service time is based on Miwa and Takakuwa (2008), who discovered that scanning an item takes 2.43 seconds, and a small 0.03 seconds is added for the constant portion in terms of scanning time. Paying takes on average 14.76 seconds. Note that Miwa and Takakuwa (2008) also estimated the average customer needs 6.76 seconds to bag an item. However, this is irrelevant for the queue, since a customer can bag their items as soon as they leave the queue upon payment.

Jockeying (switching queues) is allowed. As soon as the customer visits the exit node, the customer will join (one of) the shortest queue(s), as measured by the number of customers in the queue. If, at some point, the customer can switch to a queue that has two or more customers less in the queue than a customer's current queue, it will pay off for the customer to switch to the other queue. Note that if a queue is only one customer shorter than a customer's current queue, a customer will not benefit from switching, as they will remain in the same position in the other queue as in their current queue. Besides this, note that a customer is assumed to not know how long each other customer's service time is going to be. Realistically, this is not entirely true: a customer in a store can investigate the number of items put on the counter by the customer who is currently being serviced in front of them. However, the customers behind the services customer but in front of the considered customer may not have put their products on the counter yet, making it more difficult for a customer to analyze the possible waiting times for each of the queues in their sight.

Concerning the latter point raised, the methodology used in the current paper is based on the so-called supermarket model (Brightwell and Luczak, 2012; Brightwell et al., 2018). Upon a Poisson distributed arrival, a customer will consider d queues at random (with replacement), and will join the shortest queue in terms of the number of customers amongst those d queues.

This assumption seems to be far-fetched. Take the example of a store with 10 checkout stations, and thus 10 queues. A customer arriving near queue 1 will probably only consider a set of adjacent queues near queue 1, and not a randomly drawn permutation of d queues (say, for $d = 3$, queue 4, 8 and 10, but rather queue 1, 2 and 3). Therefore, in the current paper, a randomly drawn number will determine the first out of the d adjacent queues. Then, the shortest queue amongst the d adjacent queues is being selected as the queue to join.

Not only is the current paper novel in its approach of picking d adjacent queues that the customer will consider, it also test for the two situations in which jockeying between these d adjacent rows is and is not allowed. The algorithm optimizes the number of queues for the store. The algorithm starts with two queues, and will calculate the average number of customers in a queue for all queues considered and their associated standard deviation. d will vary between 1 and 5, and we will also consider the difference between jockeying being allowed and being prohibited.

Let us consider a $G/G/c$ queue. This means that there are c servers, or checkout stations, with a random service time μ and an average arrival rate λ. We can then define some mathematical definitions:

$$L_s = \lambda E(S) = \frac{\lambda}{\mu},$$

where L_s is the average number of busy checkout stations. A busy checkout station is defined by a station that is currently serving a customer. Based on the above equation, for stable systems where $\lambda < c\mu$, we can define the average checkout utilization rate to be

$$\rho = \frac{L_s}{c} = \frac{\lambda}{c\mu}$$

Let us now define the case in which jockeying between queues is allowed. Let us define the queue at time t to be as follows, following the notation of Morabito and de Lima (2004):

$$(n_1, n_2, ..., n_c), 0 \leq n_i \leq N. \tag{7}$$

Mathematically speaking, if two queues have lengths that differ with at least k customers, the customer at the end of the longest queue will switch to the shorter queue. We can redefine the state at time t to be as follows:

$$(n_1, n_2, ..., n_c), 0 \leq n_i \leq N, |n_i - n_j| \leq k, i \neq j, i \in \{1, ..., m\}, j \in \{1, ..., m\}.$$

In the current paper, we let k vary between 2 and 4. Obviously, the current state of the queueing system changes when a new customer enters the system. If the customer joins queue $n_q, 1 \leq q \leq c$, then the new state space is defined as follows:

$$(n_1, ..., n_q + 1..., n_c), 0 \leq n_i \leq N, |n_i - n_j| \leq k, i \neq j, i \in \{1, ..., m\}, j \in \{1, ..., m\}.$$

The number of possible state space transitions m', with our number d in mind, can then be defined as follows:

$$m' = \sum_{i=s}^{s+d-1} \delta(|n_i + 1 - n_j| \leq k),$$

where

$$\delta(|n_i + 1 - n_j| \leq k) = \begin{cases} 1, & \text{if } |n_i + 1 - n_j| \leq k, \forall j \in \{s, ..., s+d-1\}, j \neq i \\ 0, & \text{otherwise} \end{cases}$$

When a customer leaves a queue, the state changes from Equation 7 to:

$$(n_1, ..., n_q - 1..., n_c), 0 \leq n_i \leq N, |n_i - n_j| \leq k, i \neq j, i \in \{1, ..., m\}, j \in \{1, ..., m\}.$$

Then, the number of possible state space transitions m' can be defined as follows:

$$m' = \sum_{i=s}^{s+d-1} \delta(|n_i - 1 - n_j| \leq k),$$

where,

$$\delta(|n_i - 1 - n_j| \leq k) = \begin{cases} 1, & \text{if } |n_i - 1 - n_j| \leq k, \forall j \in \{s, ..., s+d-1\}, j \neq i \\ 0, & \text{otherwise} \end{cases}$$

It would be interesting to see how the average waiting time and average queue length changes between non-jockeying and jockeying situations, as well as various values for d.

A.9 Python code

In this appendix, the Python code used to produce the results in this paper will be explained. The programs can be split up in the Mixed Integer Programming part of the results, and the simulation part of the results.

A.9.1 Mixed Integer Programming programs

main.py: This is the main program to run. You will be asked quite some questions about the set-up of the program and the parameters of your model. First of all, you will be asked whether you want to use the original settings of the paper by Flamand et al. (2018). This means that the program will calculate the original APSA model and the heuristic for $\tau \in \{2, 3\}$ and $\epsilon = 0.5\%$, as well as $\tau = 4$ and $\epsilon \in \{0.5\%, 1.0\%, 1.5\%\}$. The program will also calculate the affinity results with $\tau = 4$ and $\epsilon = 0.5\%$. In case you choose not to employ these settings, the program will ask you to put values for τ and ϵ. Then, the program will ask the number of products and the number of shelves you would like to consider. It will also ask you the number of data instances you want to create. Then, it will move on to ask you whether you would like to employ the health-constraints. If you select to do so, you will be asked whether you want to employ only the model with all the health constraints, or whether you want to calculate the model with each possible combination of constraints. Finally, it will ask you which store layout you want to consider. Here, you have the option to choose the store layout as used by Flamand et al. (2018) (option 1), the grid layout (option 2) or the herringbone layout (option 3). Now, the program will start executing its calculations. A text file with some summary statistics on the results will be created and updated everytime a model finishes.

heuristic.py: This module will be called from the main in case the heuristic needs to be applied. It contains a function with the same name, from which the functions algorithm_1 and algorithm_2 will be called. This function will also write to the result files upon termination.

model_mip.py: This model contains three functions.

- *init*: The init function will simulate the data with the parameters parsed as an input by the user.
- *build_problem*: The build_problem function will build the MIP model by defining the objective

value and constraints.

- *solve*: The solve function will solve the model parsed as an input to this function.

Algorithm_1: This module contains two functions.

- *algorithm_1*: The algorithm_1 function will execute algorithm 1 as described in the paper.
- *diff*: This function will calculate the set difference between two lists parsed as an input to the function.

Algorithm_2: This module contains a function with the same name, which will execute algorithm 1 as described in the paper.

healthconstraints.py: This module contains a function with the same name, which is called whenever the model with some variation of the health constraints need to be solved. This function will also write to the result files upon termination.

sol_to_excel.py: This module contains a function with the same name, which is called whenever the current solution needs to be saved to a csv-file. It will create a csv-file for each of the five decision variables with appropriate names.

A.9.2 Simulation programs

Simulation.py: This module can be considered the main module of the simulation. It contains a Customer class, which contains the following functions:

- *__init__*: This function initializes an instance of a Customer.
- *choose_route*: This function determines the route the customer will take through the store.

Besides this, the module uses a lot of input csv-files. The required csv-files can be found in the simulation-folder. However, if you want to run the simulation on your own created MIP solutions, make sure to replace the current files with your own obtained results (that is: 'Results_APSA_TwoInequalities_y_grid.csv', 'Results_APSA_TwoInequalities_s_grid.csv', 'Results_APSA_TwoInequalities_y_herringbone.csv', 'Results_APSA_TwoInequalities_s_herringbone.csv', 'Results_CS1 + CS2 + CS3 + CS4 + CS5_y_grid.csv', 'Results_CS1 + CS2 + CS3 + CS4 + CS5_s_grid.csv', 'Results_CS1 + CS2 + CS3 + CS4 + CS5_y_herringbone.csv' and 'Results_CS1 + CS2 + CS3 + CS4 + CS5_s_herringbone.csv').

The module also outputs a lot of csv-files. For each of the four models, it will create a csv-file with the permutations of the shelves considered in each iteration, a file with all nodes to visit, a file with all paths, and a file with some summary statistics on the best attained result.

Graph_init.py: This module contains a function with the same name, and initializes our network graph.

prods_shopping_list.py: This module contains a function with the same name, and determines the intersection of all product sets of all models considered. This function also takes the result files for the variable y to determine which products were selected in each model.

neg_bin.py: This module contains a function with the same name, and can be used when determining the probability of a negative binomially distributed variable.

get_cust_stats.py: This module contains a function with the same name, and is called by the main Simulation.py to retrieve descriptive statistics on the customer.